

LUISS GUIDO CARLI

Bus Travel Time Analysis using Real-Time Data

by

Davide Viviano

Advisor: Giuseppe Ragusa

Chair of
Applied Statistics and Econometrics

A thesis submitted in partial fulfillment for the
BACHELOR DEGREE

in
Economics and Business
Department of Economics and Finance

June 2016

Declaration of Authorship

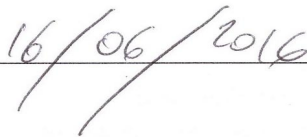
I, Davide Viviano, declare that this thesis titled and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a bachelor degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:



Date:



“L’ attesa era snervante. Erano tutti sulla stessa barca, o per meglio dire zattera, aspettando un segno, un cambiamento. Guardandosi attorno, era facile scorgere sotto ombrelli e cappucci sguardi disperati, pieni di odio e rancore. Ma una luce si avvicinava, fiacco, da una strada vicina. Ormai la luce era a pochi metri, e anche i più rancorosi si dovettero rassegnare. Era arrivato. Il tram 19. ”

(Ginevra Candidi, Racconti)

LUISS GUIDO CARLI

Abstract

Economics and Business
Department of Economics and Finance

by Davide Viviano

During this study I analyzed Real-time data from the API service of the transportation agency in Rome, ATAC SPA, integrated with static data in order to explore the effect of several variables on bus travel time. The introduction of GPS tracers and the improvement of Automatic Vehicle Location Systems has hugely increased the amount of real time information available. Unfortunately this information is not fully exploited to improve the quality of the service. In this sense, the purpose of this study is to develop useful real-time predictors of bus travel time capable to significantly outperform the one actually in use. The results show a high performance of many different linear and non-linear models trained on 20 000 observations collected within two weeks between April and May 2016. Cross validation has been used as an unbiased way to test the models. Furthermore, slack time at the end stops, number of stops to go through and distance of the bus from arrival, turned to be the most important variables affecting bus travel time.

Acknowledgements

I would like to express my deepest appreciation to professor Giuseppe Ragusa for the crucial and patient help provided in developing this study. Most of this work was developed thanks to challenging objectives that he set.

I would like to thank also Siria Angino, for the important guidance throughout the entire period of this thesis.

I thank Chiara Perricone for the material she kindly recommended.

I am deeply indebted to Kay System Italia S.r.l for the kind support in providing the processor during the entire period of this study. In particular, I am extremely grateful to Valerio Talarico for his efforts in setting the machine and giving me basic tools for programming in a Linux environment.

I thank professor Marco Scarsini, for the recommendations he pointed out about some critical points.

I thank professor Yoseph Rinott, for the important comments on model selection and for the notes that he kindly sent to me.

Finally, I thank my brother Riccardo, first year in computer science, for his guidance for programming in Python at the beginning of this study.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	ix
Abbreviations	x
1 Introduction	1
1.1 Objective of the study	2
1.2 Transportation and Economics?	2
1.3 Data Collection	3
1.4 Methodology	3
1.5 Structure	4
1.6 Background	4
1.6.1 Useful Concepts	4
1.6.2 Transit Service Reliability	5
1.6.3 A review of previous research	6
1.6.4 Predictive models in past literature	7
2 Theoretical Framework	11
2.1 A general overview	12
2.2 Predictive Power of a model	12
2.2.1 The Bias-Variance trade off	13
2.2.2 The complexity of the model	14
2.2.3 Selection criteria	14
2.2.4 Test and Training Set	15
2.2.5 Cross Validation	16
2.2.6 The right and the wrong way to do Cross-Validation	16
2.2.7 The Bootstrap	17
2.2.8 The Code	17

2.3	Linear Regression	18
2.3.1	Test Statistic on the Coefficients	19
2.4	Shrinkage Methods for Linear Regression	20
2.4.1	How do you choose lambda?	21
2.5	Stepwise	22
2.6	Generalized Methods of Moments	22
2.7	Basis Expansion and Local Regression	23
2.7.1	Splines	24
2.7.2	Kernel Smoothers	25
2.7.3	Course of Dimensionality	26
2.8	Regression Tree	26
2.8.1	Bagging	28
2.8.2	Random Forest	28
3	Description of the Problem	30
3.1	Transportation system in Rome	30
3.2	Potential variables affecting on-time-performance in Rome	31
3.3	Scheduled Arrivals	32
3.4	Real-time prediction: Muoversi a Roma	33
3.5	Atac Open Data	34
3.5.1	Static Data	35
4	Description of the Data	37
4.1	The queries	38
4.2	Construction of the Dataset	39
4.2.1	The process	40
4.2.2	Variables	40
4.2.3	Why picking only a sub-set of observations?	42
4.2.4	Critical points	42
5	Forecast Optimality under Flexible Loss	45
5.1	Generic Definition	45
5.2	Estimation of the loss	46
5.3	J-test and robustness of results	48
5.4	Quantile Test on Optimal Forecast	49
5.5	Generalized Mincer-Zarnowitz regression	50
5.6	Comments and Critical Points	50
6	The Empirical Analysis	51
6.1	Descriptive Statistics	51
6.1.1	Variable description	51
6.2	Univariate linear regression	52
6.3	A smoother function	53
6.4	Multivariate regression with Lasso and Ridge Regression	54
6.5	Stepwise	55
6.6	Pruned tree and Random Forest	56
6.6.1	Variables Importance	56
6.7	Stop 70988: A case study for future research	57

6.8	Test out of sample using Random Forest and MSE	60
6.9	Prediction with a different loss	60
7	Conclusions	62
7.1	Achievements	62
7.2	Future Research	64
7.3	Critical Points	64
7.4	Final Comments	66
A	Miscellaneous	67
A.1	Proof EDF	67
A.2	Proof Moment Condition	68
A.2.1	Moment Condition 1	68
A.2.2	Moment Condition 2	68
A.2.3	Moment Condition 3	68
A.3	Stepwise: results	69
A.4	Code	69
A.4.1	Code for elaboration of data	69
A.4.2	Traffic Function	72
A.5	Final notes on the models	73
	Bibliography	74

List of Figures

2.1	Lasso: Cv with standardized and Non Standardized Covariates	21
2.2	Smoothing Splines: examples with different lambdas	25
2.3	Pruned tree on sub-sample of Data	27
3.1	Metro Lines	31
3.2	Mean Number of Arrivals per stop	32
3.3	Distribution of Travel time on 5,10,20,30 minutes prediction	33
3.4	Prediction of Atac	34
4.1	The four routes analyzed	37
4.2	CPU Usage	38
4.3	Distribution Travel Time in DataSet	41
4.4	Travel time Estimation	43
4.5	Overlapping	43
5.1	Loss Function Estimated with Moment Conditions	47
6.1	Spline, EDF VS MSPE	54
6.2	MSPE and Number of Covariate - Lasso 193 variables	55
6.3	MSPE of Random Forests and pruned tree	57
6.4	Variables importance in random forest	58
6.5	Stop70988	59
6.6	Out of sample MSE on 2nd,3rd,4th,5th of May	60
6.7	Loss function used for our prediction	61
7.1	Summary of Model Performance	64

List of Tables

5.1	Generalized Methods of Moments: Results	48
5.2	Tests for optimality: indicator test and Mincer-Zarnowitz test	49
6.1	Descriptive Statistics	52
6.2	Univariate linear model	53
6.3	Performance of Atac and Univariate linear model	53
6.4	Linear Regressions - feature selection with RF	58
6.5	Test-set error of Atac and GMM under asymmetric loss	59
6.6	Performance out of sample of Atac and Random Forest under MSE	60
A.1	Features selection with Stepwise	69

Abbreviations

AIC	Aikake Information Criterion
APC	Automatic Passenger Count
API	Application Programming Interface
AVL	Authomatic Vehicle Location
CV	Cross Validation
DGP	Data generating Process
DF	Degrees of Freedom
EDF	Effective Degrees of Freedom
ELNET	Elastic Net
GMM	Generalized Method of Moments
LOOCV	Leave One Out Cross Validation
MSE	Mean Squared Error
MSPE	Mean Squared Prediction Error
NN	Neural Network
OOB Err	Out Of Bag Error
OLS	Ordinary Least Squares
RF	Random Forest
RR	Ridge Regression
SVM	Suppor Vector Machine

*Dedicato ai miei genitori, a nonna Anna, mio fratello Riccardo, i
miei nonni paterni
e al mio carissimo amico Federico*

Chapter 1

Introduction

The large use of innovative technologies, such as GPS tracers on each bus, has increased the amount of data available to transit companies. Whereas transport agencies have made huge efforts in collective significant amount of data, this information has not been particularly helpful in improving the quality of the service. Automatic Vehicle Location (AVL) or Automatic Passenger Count (APC) are some examples of new technologies that are still not particularly useful for operating strategies of transit companies.

In this sense, Atac SPA , the agency that controls the public transportation system in Rome, represents an interesting case. The AVL system in Rome covers almost all the buses, metro and trains , reporting their geographic position in real-time. Furthermore, the system provides data also on traffic conditions, on number of stops of each bus, on the route, on the identity of the vehicle and others. Unfortunately, the large amount of data still does not find any useful application. Real-time prediction of arrival time are based only on the length of a route and on the average speed of the vehicle. In 2014, the agency gave free access to the API service that provides all real-time information collected by the AVL system. In addition, they published on-line the “General Transit Feed Specification Level” containing all static transit information of the agency, including scheduled time for each route at each stop and features of routes and trips. The large amount of data at disposal makes this a case-study whose analysis can lead to interesting results that can be implemented by other transit agencies.

The collection of data was possible thanks to the kind help of Kay System Italia s.r.l. that provided a Server with a Intel Xeon Processor E5645 with 4-physical cores and 16-virtual CPUs for the whole period of the study. The analysis was done using the

computer language R, implemented with several packages that I will describe in the next chapters.

1.1 Objective of the study

First, I will construct a predictor for arrival-time using different approaches using different loss functions. In fact, the low performance of the predictions provided by Atac makes this a crucial topic. To test the predictor I will use both cross validation and out of sample observations collected in different days.

Furthermore, given the large amount of variables we can control for, we will try to find the variables most correlated with bus travel time. As I will show, many transit agencies have studied real time data in order to find causal relationships with travel time. This analysis will be useful in order to assess the impact of different operating strategies on the quality of the service.

1.2 Transportation and Economics?

You may wonder - as my parents do - why a future economist should deal with transportation problems. I will try to provide a short personal answer in order to tell why I believe that this problem is particularly important for an economist.

Nowadays, massive amount of data flows on-line. Unfortunately it is usually difficult to collect, to store or even to interpret these data. For this reason, computer science is playing a growing role in economics, together with statistics. The use of data can be a potential way to improve the understanding of complex phenomena. In fact, many problems in finance, macroeconomics and microeconomics deal with data collection and data analysis. Furthermore, it is my strong belief that data are particularly useful if exploited to improve the macro or microcosm where we live. In this sense, in this thesis I try to use the large flow of real-data from the AVL system in order to understand how these data can be useful for the people in Rome. Similar techniques used in this thesis can be applied in different fields of economics in order to find potential ways of interpreting economic phenomena.

1.3 Data Collection

In order to carry out this study I have collected data during two weeks between April and May. The data were collected sending every 10 seconds a query to 26 different stops uniformly distributed on Rome and receiving information about the route, the identification number of the vehicle, the predicted waiting time, the distance in terms of missing stops and other variables of all the buses directed to that stop. Every 10 seconds we received and saved between 80 and 250 observations, with a total of more than 2 millions of observations. These data were then merged with other data sets to obtain further features at disposal.

Whereas AVL data are available for the traffic service agency in Rome, APC data are not available. On the other hand, the data collected from ATAC contain real-time information controlling for bus stops, identification of the vehicle and route. The availability of the whole dataset with scheduled time for each day, features of a route and many other variables make this case interesting to understand how to evaluate the reliability of a public traffic company. In the construction of the dataframe I extracted arrival times and other variables. In the next chapters I will provide further details about the data mining process.

1.4 Methodology

I will provide to the reader a brief enumeration of the steps followed during this work. Any step will be described in details in the next sections.

1. Problem understanding. Identify potential strength and weaknesses of the public transit system in Rome.
2. Data collection. Explore the most efficient way to collect data and implement the process.
3. Data analysis. Analysis and exploration of the data collected, with particular attention to missing values; merge of dataframes to obtain the final dataset.
4. Evaluation of forecast optimality under a general class of loss functions.
5. Modelling for prediction and inference. Construction of models and evaluation of performance. Interpretation of the models.
6. Final considerations. Evaluation on the assumptions of the models and their weaknesses. Description of challenges for future research.

1.5 Structure

In the next section I will provide a review of literature. In chapter 2 I will go through the theoretical framework necessary to understand the empirical analysis. Chapter 3 will describe the problem, focusing on the reliability of the transit service in Rome. Chapter 4 will be about the process of data collection and data mining. In chapter 5 I will discuss the test for forecast optimality assuming a general class of losses. Chapter 6 will be about the empirical analysis, in chapter 7 there are final conclusions, achievements and critical points.

1.6 Background

1.6.1 Useful Concepts

Before entering in the details of the past literature, I will provide to the reader key concepts for the understanding of this thesis:

1. Traffic signal priority is a system that coordinates along a specific route the traffic lights in order to reduce stop time.
2. Automatic vehicle location is a system that provides to the user real-time data regarding the location of buses and the expected arrival at specific stops.
3. Automatic passenger count is a system that counts the number of passengers on the bus.
4. On-time-performance is a measure of schedule deviation.
5. A route is a constant path between two end-points and it has a specific identification name.
6. Direction refers to the direction of the bus on a specific route and it can be either inbound or outbound.
7. A trip is always on the same route with a specific direction.
8. Slack time is the time that a bus waits at the stop before leaving again.

1.6.2 Transit Service Reliability

Transit service reliability is a key issue for transit agencies. Unreliable transit services may increase passengers cost, have negative impact on the reputation of the company and decrease the number of customers. Transit service reliability (TSR) typically relates to on-time-performance and travel time variation and it has been defined in several ways by the researchers. According to most of the literature TRS is defined as the ability of the transit system to adhere to scheduled time. Abkowitz et Al. [1] defined TSR as “the invariability of transit service attributes that affects the decisions of users and operators”. For Strathman et al.[47] reliability depends mainly on delays and schedule adherence. El Geneidy et al.[14] instead underlined four different aspects of reliability: “high accessibility from the origin to the destination of the travel, high predictability of waiting time, short in-vehicle time and low variance in run time.”

According to the Transit Cooperative Research Program[41] a key element in assessing transit service reliability is travel time. Running time is defined as the time that a bus takes to travel between two different points[14]. More broadly, Carrion et al. [6] defined travel time as “the time elapsed when a traveler displaces between two spacial positions”. Travel time is made of two different components, waiting time and in-vehicle time. Waiting time may depend on both the behaviour of the passenger and the travel system. Passengers who control the scheduled arrival time should have a lower expected waiting time than passengers who do not check scheduled arrival time at the stop. This is true if, on average, the probability of bus arrival increases as the time is closer to scheduled time, as it should be. Distribution of time arrivals has been studied by many researchers as a key issue in service reliability. The typical distribution that is used in the literature is a skewed distribution, either a gamma or a lognormal distribution [6]. In fact, buses normally do not leave before scheduled time, while they may be late at some stops. Another important topic affecting running time is slack time. Slack time tend to be high if there are few vehicles on a particular route. By increasing the in-vehicle time of passengers it may reduces the waiting time for other passengers. Some studies have been carried out to find optimal solutions for slack time under different operating strategies. Zhao et al.[52] found a slack time ratio of 25 per cent analyzing Los Angeles County Data. According to the schedule of the transport agency in Rome Atac, the only slack time is between end stops of a route. Finally, important attention has been given to the double nature of travel time delay[6]. Delays may be predictable and unpredictable. Predictable delays, if known by potential passengers, have not negative impact on the expected travel time. Some examples of predictable delays are traffic during peak hours. Unpredictable waiting times have instead a huge impact on transit

service reliability. They may be due to driver's inexperience, mechanical problems of the vehicle, unexpected traffic congestion, etc.

1.6.3 A review of previous research

Most of the research that I will describe in this subsection was aimed to improve service quality of several travel agencies. Research in bus service reliability was initially developed in many American cities, such as New York, Minneapolis and Portland. The analysis on the metro of London represents an European example. Nowadays many Chinese areas are showing increasing interest in this research field given the increasing importance of public transportation in many Chinese regions. Perhaps, the large use of Automatic Vehicle Location (AVL) and Automatic Passenger Count (APC) represents challenging and useful instruments to analyze the service reliability of public transportation.

According to Diab et al.[11], the study of on-time-performance was developed following different approaches. One was to study the frequency distribution of schedule deviations, focusing on the absolute number of delays of buses. A second one, introduced by Abkowitz et al.[1], consisted in obtaining indicators in service reliability, focusing on travel time and schedule deviation. The availability of data to carry out empirical analysis was a major problem in the last two decades of the 20th century. In this sense, the effectiveness of operating strategy has been tested using different simulation techniques. With a Monte Carlo simulation, Senevirante et al.[45] studied the point to point travel time in order to evaluate the sensitivity of operational changes on the quality of service in terms of headway variation. The variables they considered were the number of time stops, the passenger demand and the length of a route.

Henderson et al.[22] studied the service reliability of New York Subway looking at the data collected between 1988 and 1990. They constructed a multinomial logistic model with dependent variable the probability of deviation from schedule time and with independent variables the number of routes merged, whether public schools were in session, a crowding index and others. In 1993 Strathman et al.[47] studied the fixed route system in Portland, Oregon using a multinomial logit model to understand the effects of elements such as the number of passengers, the scheduled headway, the distance and the experience of the driver on on-time-performance.

The introduction of AVL made it easier to collect real-time data. In 1999 Kalaputapu[29] analyzed the AVL data from Tidewater regional Transit in Virginia on a single route,

to identify a powerful predictive model for travel time. In doing so he proposed several alternatives, comparing a neural network model to ARMA and ARIMA. In 2001 Kimpel[30] carried out a further analysis in Portland, looking at data from the public service company Tri-Met. He constructed a model where the delay variation depends the accumulated delay variation, the route type, the number of stops, the time of the day - peak and off-peak hours - and others. Analysing the AVL data from Metro-Transit in Minnesota, El Geneidy et al.[14] built four different multivariate regression models looking at the effects of passenger activities, peak-hours, driver experience and other variables on running time deviation. Diab et al.[11] used linear regression models to identify the effect of different operating strategies implemented by the Societ  de Transport de Montreal between 2007 and 2011. Figliozzi et al.[2] studied the stop-to-stop travel time to determine the recovery of the bus at each bus stop in case of delay. Controlling for transit signal priority at the intersections, they showed an high variance of recovery time among different stops. Finally, in 2015, Feng et al.[16] showed that stop location, signal delay and traffic condition have a significant impact on travel time.

1.6.4 Predictive models in past literature

In this section I will provide a brief overview of the most common models used in prediction of bus travel time. For seek of brevity, I provided mathematical intuitions of few models. In the next chapter I will discuss in a comprehensive way the models that I will use, the main limitations and the selection criteria.

Prediction for bus arrival was first developed in 1999 by Kalaputapu[29], comparing neural network (NN) to ARIMA and ARMA.

In general, neural network have been commonly used in the past literature. The large use of Artificial Neural Network is due to their adaptive features and the facility to operate with real-time data. The structure of NN is characterized by different nodes that elaborates the input variables and communicate to subsequent nodes new input variables. These inputs are weighted by minimizing a certain loss function. Lin et al.[26] showed an high performance of NN analyzing data from Jinan, China and using 20 per cent of the data as test set. They constructed sub-ANN controlling for am/pm peak hours and weekend due to the lack of traffic information.

Johar Amita et al.[3] used neural network to analyze real time data of the public transit agency in Delhi, India. They used as input variables accrued delays, dwell time and distance in order to compute travel time between two points. The model outperformed linear regression. Gurmu et al.[20] constructed ANN to analyze bus travel time with GPS data from the public transportation system of Macae, Brazil, and they showed an

higher performance of ANN compared to historical averages. Many other researchers have have developed similar methods.

Lin et al.[25] described a Markov Chain model to predict delays of buses. Their model considers recovery time as a key issue in travel time predictability and it assumed uniformly spaced bus stops. The model suggests a positive relationship between degree of travel recovery and distance. In the next few lines I will provide a brief description of the model:

P is defined as the one-step transition matrix of dimension N by N. It reports the probabilities $p_{i,j}$ of a delay d_j at stop $k+1$ conditional on a delay d_i at stop k . $p(k+1) = p(k)P$ where $p(k)$ is a vector 1 by N with the current state of nature. ¹ To construct the predictor we substitute k times $p(k)$ with $p(k-1)P$ until we get:

$$p(k+1) = p(0)P^k. \quad (1.1)$$

This result come from the fact that P is constant for any k .

Given d^T as a sequence of possible delays, $E_{t,k} = p(0)P^k d^T$, where $E_{t,k}$ is the expected delay at stop k at time t , $p(0)$ is the known state of nature at the initial stop.

Similarly, many researchers have used a Kalman predictive algorithm. Kalman filter is very similar to Markov chain predictors, with the difference that it recursively estimates the error covariance matrix and the prediction. Kalman filter predicts the vector y on a discrete time where:

$$y_k = Ay_{k-1} + Bu_k + w_{k-1} \quad (1.2)$$

and with a measurement error:

$$\eta_k = Hy_k + o_k \quad (1.3)$$

where w, o are gaussian white noise ².

A, H span R^n and , whereas they may depend on the state k , we assume to be constant for seek of simplicity. The filter computes an a priori prediction error and a posterior error based on η_k . The predictions are:

¹An example can clearly illustrate this point: assume a current delay of two minutes at stop 0. Given a vector $d = [-100 -98 \dots -94 \dots -2 \dots 100]$, $p(0) = [0 0 \dots 0 \dots 1 \dots 0]$, where the i th element of $p(0)$ equal to one corresponds to the i th element of d equal to the current delay(in this case -2).

²Gaussian white noise are independent normally distributed random variables with mean zero and finite variance.

a priori: $\hat{y}_{pr,k} = Ay_{k-1} + Bu_k$

a posteriori: $\hat{y}_{po,k} = \hat{y}_{pr,k} + K(\eta_k - H\hat{y}_{pr,k})$

where K weights the distance between the error measurement and its expectation. The error is $\epsilon = \hat{y} - y$ and it can be both on the a priori and a posteriori prediction.

To find K , the Kalman filter minimized the sum of square residuals. In particular, P is the a posterior error covariance matrix, where $P_k = E[\epsilon_k \epsilon_k^T]$, with ϵ the a priori error term.

By substituting K in P the algorithm solves for $\arg \min \text{trace}(P_k)$.³

In 2004 Shalaby et al.[46], used two distinct Kalman Filter Algorithms to predict the travel time between two points and dwell time, based on expected number of passengers, showing a better performance than historical average, regression and neural networks in terms of error on a test set. In their analysis they used real-time data of the Toronto public transport system. Similar algorithm has been used by Wang et al.[50] using data from the city of Yinchun, China.⁴

Chen et al.[9] used an autoregressive models in bus trajectory prediction, showing linear correlation between future and past travel times and checking the convergency of the coefficients. Their results reported unstable estimation for the first stops of a route. An autoregressive model of order p , $AR(p)$, is a model that describes the dependent variables depending linearly on the sum of its previous p values and a stochastic term. The general notation is

$$y_t = \alpha + \sum_{1}^p \rho_p y_{t-p} + u_t \quad (1.4)$$

\hat{y}_t is computed as the linear projection of y_t onto R^{p+1} spanned by the regression matrix whose columns are the vector of ones, y_{t-1} , y_{t-2} , ..., y_{t-p} . Ordinary least squares are one possible way to estimate the regression coefficients. The problem of auto-regressive models relies on the autocorrelation between observations at different time. Asymptotic theory applies if certain conditions are satisfied. These conditions are stationarity, ergodicity and mixingale. A time-series is stationary if the joint distribution of observations depends only on the lag time between these observations. The same applies to marginal distributions: for a stationary time serie Z_t , $E[Z_t|T]=E[Z_t]$ and $E[Z_t' z_{t-j}] = \gamma(j)$, depending only on the lag j and not on t . Ergodicity means that as the lag between two time windows tend to infinity, the joint distributions of the observations in each window are independent. Finally, mixingale is a necessary condition that states that the sum of the covariances between z_t and $z_{t-j} \forall j \in [1, \infty)$ converges. This property is necessary to assume a finite variance of the serie. ARMA model adds an additional stochastic

³The trace of a matrix is the sum of the elements on the main diagonal. In this case it corresponds to the sum of squared residuals.

⁴For a more comprehensive understanding of Kalman filter look at Bishop, G., Welch, G. (2001). An introduction to the kalman filter.

component where MA(q) identifies:

$$y_t = u_t + \theta_1 u_{t-1} + \dots + \theta_q u_{t-q} \quad (1.5)$$

where u is a white noise. An ARMA(p,q) is:

$$y_t - \rho_1 y_{t-1} - \dots - \rho_p y_{t-p} = u_t + \theta_1 u_{t-1} + \dots + \theta_q u_{t-q} \quad (1.6)$$

Usually an ARMA model is used when the stochastic error is thought to be persistent.

Jing-nan Wang et al.[49] used instead support vector machine analyzing real time data of public transportation in Beijing, China. Support vector machine is a powerful techniques developed during the 90s. The idea behind support vector machine is intuitive: it maps the explanatory matrix onto an higher dimensional space to find linear boundaries in this space. I will provide an example from a classification problem to better illustrate the problem: suppose that we want to find circular boundaries to classify a certain dependent variable using only two covariates. Any linear regression would not be able to give this result because it can construct only a linear split between the observations. If instead the X matrix is mapped onto \mathbb{R}^3 , linear boundaries on a sphere would correspond to circular boundaries on the 2-dimensional plane. In general, support vector machine uses the Kernel-trick, to avoid the problem of choosing a specific transformation from a set of infinite transformations. SVM minimizes a given loss function without imposing a predetermined shape to the boundaries. Finally, in order to avoid overfitting , penalty parameters are added. One of the main difficulty of SVM is to find the best tuning parameters to obtain a strong predictor.

Chapter 2

Theoretical Framework

One of the most powerful concept in Econometrics and Applied Statistics is the concept of regression. Regression consists in finding a general function $\hat{f}(x)$ that describes a continuous output variable y conditional on an input matrix X . Regression is commonly used for different objectives. The first one is to assess causal relationship between variables and , in this sense, the notion of ceteris paribus is crucial. A ceteris paribus analysis consists in understanding what is the expected change in y given a marginal increase of x_p , keeping constant all other variables. It generally requires a simple and understandable model to estimate partial effects of certain variables. The most commonly used is the linear model. Hypothesis testing on the coefficients are considered powerful methods to test causal relationship conditional on certain assumptions.¹ A different “industry” of studies is prediction. The objective of a predictor is to minimize the error on out-of-sample values of y , sometimes with no means of interpretation.

The most commonly used method to find the best regression function is to pick the function that minimize a certain loss function from a given class of regressor.²

This chapter is organized as follows: the first part is focused on general concepts regarding model complexity and model selection. The second part explains the models used in this thesis. The plots showed are built using specific sub-samples of the data set used during the empirical analysis.

In the chapter I will use matrix notation where $X \in \mathfrak{R}^p$, is a real valued random input vectors and $Y \in \mathfrak{R}$ is a real valued random output variables.

¹Note: Whereas variables may be correlated, this does not imply causal relationship.

²A loss function is a function that attributes a cost to the prediction error.

2.1 A general overview

Any output variable can be described as:

$$Y = f(x) + \epsilon \quad (2.1)$$

Where $f(x)$ is the true function and ϵ is a random noise. In estimating the output variables you can use an infinite set of models. The broad description of a linear model is:

$$\hat{Y} = SY \quad (2.2)$$

Where (S_{ij}) does not depend on y_i . This is a very broad definition that considers a large class of predictors. S is the projection matrix of Y . $S = X(X'X)^{-1}X'$ for linear regression, as I will show later.

The best regressor is the one that is able to best approximate the true function. Of course, this is an hard task and a solution to this problem is to find the predictor that most minimize a certain loss function from a given class of regressors. A common loss function λ is a squared loss function.

$$\Lambda(Y - \hat{f}(x)) = (Y - \hat{f}(x))^2 \quad (2.3)$$

The main feature of the squared loss function is its symmetry where $\Lambda(\theta) = \Lambda(-\theta)$. Consequently, $\hat{f} : X \rightarrow \Re$ such that $\hat{f} = \underset{\hat{f}}{\operatorname{argmin}} E[\Lambda(\hat{f}(X_i) - y_i)]$.

The generic solution to this minimization problem with squared loss is[35]:

$$\hat{f}(x) = E[Y|X] \quad (2.4)$$

2.2 Predictive Power of a model

The mean squared error is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

Whereas the MSE is an easy concept and it is easy to compute, this does not correspond to the true error. Before entering in further details I introduce the concept of overfitting and underfitting. Underfitting means that the model that we derive do not capture enough variance of in-sample observations. Underfitting may depend on the rigidity of the models, on the number of covariates used in the regression or on many other factors and it would lead to high MSE.

With overfitting the $\hat{f}(x)$ captures also the in-sample random noise. Whereas the MSE on the in-sample observation is almost zero, the MSE on new observations can be high, leading to poor predictive power.

According to Efron[13], the true error can be decomposed into the sum of the MSE and an optimism component that depends on the complexity of the model and on the observation set.

2.2.1 The Bias-Variance trade off

Two important concepts in statistics are bias and variance. I will provide definitions referring to the bias and variance of an estimator. Bias is the distance between the expected value of the estimated function $\hat{f}(x)$ and the true function $f(x)$. In general bias is defined as: $E[\hat{f}(x) - f(x)]$. Variance is defined as the average squared distance between each prediction and the expected value of the prediction. $\text{Var}(\hat{f}(x)) = \frac{1}{n} \sum_{i=1}^n (\hat{f}(x_i) - E[\hat{f}(x)])^2$.

The bias-variance trade off is the following ³:

$$\begin{aligned}
 E[(Y - \hat{f}(x))^2] &= E[(Y^2 + \hat{f}(x)^2 - 2Y\hat{f}(x))] \\
 &= E[Y^2] + E[\hat{f}(x)^2] - 2E[Y\hat{f}(x)] \\
 &= E[Y^2 - E[Y]^2] + E[Y]^2 + E[\hat{f}(x)^2 - E[\hat{f}(x)^2]] + E[\hat{f}(x)^2] - 2YE[\hat{f}(x)] \\
 &= \sigma_\epsilon^2 + \text{Var}(\hat{f}(x)) + E[Y - E[\hat{f}(x)]]^2
 \end{aligned}
 \tag{2.6}$$

The first term of the equation is random noise. The second term is the variance of the estimator and the last term is the bias squared. Whereas the first term is an irreducible error, the sum of the last two terms is the reducible error. We would like to reduce as much as possible both the bias and the variance. Unfortunately, there is a trade off between the two. In fact the higher the complexity of the model the lower the bias, but the higher the variance. Intuitively, a more complex and, consequently, flexible model, is able to capture more in-sample information, but it generally requires to estimate more parameters, increasing the error.

³ The result comes from: $E[X^2] - E[X]^2 = \text{Var}(X)$.

2.2.2 The complexity of the model

Complexity is a very broad and difficult concept in regression. A model is considered more complex if it includes additional information from the sample observations. In linear regression complexity usually relates to the number of covariates X_p included in the model. The higher the number, the more the in-sample variance captured by the predictor and the higher the complexity. The complexity of the model is related to the effective degrees of freedom of the regression matrix. For linear regression, the number of effective degrees of freedom corresponds to the number of variables to be estimated p ⁴. For a linear function this is the number of coefficients plus the intercept.⁵

Degrees of freedom measure the complexity of the models by identifying the dimension of the sub-space spanned by the regression matrix. As we will see in the next few sections, many selection criteria relies on this concept.

Unfortunately for many models the number of parameters to be estimated, is different from the effective degrees of freedom. Ridge regression, smoothing splines, lasso and many other models can have EDF different from DF.⁶

A more general definition of effective degrees of freedom is the trace of the matrix S . This is a trivial result for a linear projection, while it is less intuitive for other models. According to Hastie et al.[23] “when S is not a projection, $trace(S)$ accumulates fractional degrees of freedom for directions of y that are shrunk, but not entirely eliminated, in computing μ ”, where μ is the expected value of the output variable. It turns out that $trace(S)$ ⁷ = $\sum_{i=1}^n cov(\hat{y}_i, y_i) / \sigma^2$.

2.2.3 Selection criteria

An intuitive measure that tells how much the model fit the observations is the R^2 . The R^2 is equal to the ratio between the estimated sum of squares over total sum of squares. Intuitively it represents the percentage of in-sample variance captured by the model. Although it is commonly used in the literature it has not theoretical justification relying on asymptotic theory. Furthermore, any additional variable always increases the R^2 . For

⁴note: model degrees of freedom, equal to p must not be confused with residual degrees of freedom equal to $n - p$, with p equals to the number of parameters estimated. See Janson, L., Fithian, W., Hastie, T. J. (2015). Effective degrees of freedom: a flawed metaphor. *Biometrika*, asv019 for a better understanding.

⁵If we recall (2.2) , effective degrees of freedom for a linear regression corresponds to rank of matrix.

⁶E.g.: RR: $\hat{\beta} = \arg \min_{\beta} (Y - X\beta)'(Y - X\beta) + \lambda \beta^2$, where λ introduce a penalty on the betas. In this case the effective degrees of freedom are less then the covariates in the model because each variable is shrunk towards zero.

⁷See the Appendix for a proof of this statement.

this reason it is usually used the adjusted R^2 , adjusted for the number of parameters in the model:

$$\text{Adjusted } R^2 = 1 - \frac{(n-1)}{(n-k-1)} \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y}_i)^2} \quad (2.7)$$

The second term introduces a penalty for each additional parameter to be estimated. An alternative measure of the deviation is the Mallows's C_p :

$$C_p = MSE + \frac{d}{n} \hat{\sigma}^2 \quad (2.8)$$

Where d is the effective degrees of freedom, n is the number of observations and σ is noise variance. C_p is an unbiased estimator of the true error for OLS estimators, assuming a normal distribution of the output variable, where the optimism parameter is $\omega(\mu, \sigma^2) = 2(p/n) \sigma^2$.⁸

The Aikake Criterion Information is very similar but more widely applicable. Defining the loglikelihood function as $\text{loglik} = \sum_{i=1}^n \log P_{\theta}(y_i)$ [40]:

$$AIC = -\frac{2}{N} \text{loglik} + \frac{d}{N} \quad (2.9)$$

The AIC is not valid if the model is chosen adaptively and the effective degrees of freedom d is less than the number of parameters estimated. In case of gaussian covariates, the AIC coincides with the C_p .⁹

2.2.4 Test and Training Set

A common method in statistics to validate a predictor is to divide data into a training and test set. A training set is a random sub-sample of the observations, used for constructing the estimator. The test set is the set of the remaining observations used to test the model and to assess the out-of-sample error. Note that there is a trade-off: the more the observations in the training set the lower the bias on the estimated of the MSPE (mean squared prediction error). This is because the model is trained with more observations. On the other hand, the higher the number of observations in the test set the lower the variance of the out-of-sample prediction error. As economists say, there is no free-lunch.

⁸For a more comprehensive proof of this statement see Efron, B. (1986), How biased is the apparent error rate of a prediction rule?. Journal of the American Statistical Association, 81(394), 461-470.

⁹There are many other selection criteria such as the BIC, but a more accurate description goes beyond the scope of this thesis.

2.2.5 Cross Validation

Cross validation is a well-known technique to validate a predictor. It does not require any assumption. K-fold Cross Validation works in the following way: the data are divided into k random folds of the same length. Each times it is constructed the model using K-1 folds and the remaining fold is used as a test set. This process is repeated K times, considering the out-of-sample error as the mean of the k different mean squared prediction errors(MSPE) obtained on the k different folds.

Common features for K are either 5 or 10 folds. Generally, this number depends on the number of observations in the dataset. In fact, a bigger K would reduce the bias of the MSPE, but it would increase its variance and vice-versa. leave one out cross validation consists in taking $K = n$, where n is the number of observations. In this case the MSPE has the lowest possible bias, because almost all the observations are used to construct the model, but it has an high variance, because all the models are strongly correlated each other. LOOCV is very common in practice because it can be easily computed using the following formula[18]:

$$GCV(\hat{f}) = \frac{1}{N} \frac{\sum(y_i - \hat{f}(x_i))}{(1 - \text{trace}(S)/N)} \quad (2.10)$$

2.2.6 The right and the wrong way to do Cross-Validation

A common pitfall in doing cross validation is to choose the model and test it in two different moments. I will provide an example from Hastie, Tibshirani and Friedman that makes the point clear [18]. Suppose we have classification problem with 30 observations and 1000 covariates independent on the class labels. Consider we use a stump, a tree with a single split, to classify our observations. The argument against Cross Validation is: “Fitting to the entire training set, we will find a predictor that splits the data very well. If we do 5-fold cross-validation, this same predictor should split any 4/5ths and 1/5th of the data well too, and hence its cross-validation error will be small (much less than 50 per cent) Thus CV does not give an accurate estimate of error.” Where does this argument fails?

The pitfall is the following: choosing first the best stump and then test it , is incorrect because the choice of the model is correlated with all the data. In this case the right way of doing cross validation is to find the best stump on the 4 folds, test on the fifth, store the MSE and repeat the process k times, every time finding a stump on different folds and test it on a new fold. Hastie et al.[18] provided empirical evidence of it with many simulations.

2.2.7 The Bootstrap

Bootstrap consists in creating new samples by randomly picking with replacement observations from the initial sample, assuming perfect randomization. This methodology permits to artificially increase the number of samples and to decrease the variance of estimated statistics. It can be applied in several ways. Unfortunately the variance of the statistics cannot be go beyond a certain threshold with bootstrapping. Another common application is to use the standard deviation of the n statistics as an unbiased estimator of the standard error of the estimator.

2.2.8 The Code

In my analysis I used cross validation to evaluate the performance of the model and I bootstrap N times to get an estimate of the standard error of the MSPE. In this section I provide the structure of the code that I used to cross validate with bootstrapping. The MSPE is the mean of MSPE and the SE is the standard deviation of MSPE. The code is written for parallel execution. For seek of brevity, I used some abbreviations from the syntax of R; y refers to travel time in `boot.dat`. K is the number of folds

```

MSPE is:append
├─ foreach i in 1:N
├─ export Data, export k, call the libraries in the cores
├─ ii = sample(rep(1:k, length = nrow(data)))
├─ pred = vector of NA with length= nrow(data)
├─ b = sample(nrow(data), repl=TRUE)
├─ boot.dat = data[b,]
│   └─ for j in 1:k
│       ├── hold = (ii==j)
│       ├── train = (ii!=j)
│       ├── data.train = boot.dat[train,]
│       ├── data.test = boot.dat[hold,]
│       ├── find the best.model on data.train
│       ├── model = best.model(y~ ., data= data.train)
│       └─ pred[hold] = predict(model, newdata=data.test)
└─ mspe = mean((y - pred)2)
└─ print(mspe)

```

2.3 Linear Regression

Linear regression assumes that $E[Y|X]$ is linear in the input covariates:

$$y_i = \beta_0 + \sum_{p=1}^k x_{p,i} \beta_p + u_i \quad (2.11)$$

Where y is the dependent variable, β_p is the coefficient of the p th covariate, X_p is the value of the covariate p for the i th observation and u_i is the error term for the i th observations with zero expected value.

Linear models were one of the first models developed in statistics and they are still very powerful. In fact, they can outperform fancy flexible models in many circumstances. A general definition of a linear model is:

$$g(x) = \sum_{p=0}^k X_p \beta_p \quad (2.12)$$

This is a broad definition and it can contains any basis expansion leading to polynomials transformations, linear transformation on the covariates, dummies and interactions. In fact, a linear model is linear in the parameters β . Note that , according to this definition, β_0 is the intercept and X_0 is the ones vector. There are two main ways to estimate the coefficients: Maximum Likelihood Estimation and Ordinary Least Squares. Whereas MLE requires some assumptions on the distributions of the error term, OLS exploit the concept of linear projection . OLS find the parameters by minimizing the sum of squared residuals.

$$\beta^{OLS} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)'(Y - X\beta) \quad (2.13)$$

The solution to this problem is $\beta = [E(X'X)]^{-1} E[X'Y]$ for the population. By the analogy principle ¹⁰ $\hat{\beta} = (X'X)^{-1}(X'Y)$.

The geometric interpretation of this solution is the following: suppose we want to find the closest point \hat{Y} in the column space of X , with $\operatorname{rank}(X) = p$. That is, we want to find $\|Y - X \hat{\beta}\| \leq \|Y - X \beta\| \forall \beta \in \mathbb{R}^p$. Because \hat{Y} is in the column space of X , it exists $\hat{Y} = X \hat{\beta}$ that is the linear projection of Y in the column space of X . By the orthogonal

¹⁰“The analogy principle for choosing an estimator says to turn the population problem into its sample counterpart”, Wooldridge, *Econometric Analysis of Cross Section and Panel Data*.

decomposition theorem, $Y - \hat{Y}$ is orthogonal to each column of X . Consequently, if we dot each component of this distance with $X_p \forall p$, we get a zero vector. $X'(Y - \hat{Y}) = 0 \rightarrow X'(Y - X\hat{\beta}) = 0 \rightarrow \hat{\beta} = (X'X)^{-1}(X'Y)$.

Note that the main assumption is that $y_i - \hat{y}_i = u_i$ is orthogonal to $x_{p,i}$ for any p and $X'X$ is invertible. In general, the crucial assumption for linear regression is $E[X'U] = 0$. We can prove that $E[\beta^{OLS}] = \beta + E[X'E[u|X]]$. If the previous assumption is satisfied, the estimator is an unbiased estimator for the true parameter $\forall n$ observations in the sample size, that is $E[\beta^{OLS}] = \beta$

Another important property for OLS, is consistency. Under the assumption of invertibility of $E[(X'X)]$ and finite second moment of X , the estimated beta converge in probability to the true one as n goes to infinity.

Finally, according to the Markov-Gauss theorem β^{OLS} is BLUE, best linear unbiased estimator, under the assumption of homoskedasticity. ¹¹

2.3.1 Test Statistic on the Coefficients

An important concept is testing the significance of the coefficients. Practically, we want to test whether there is a certain correlation between X_p and Y , assuming as null hypothesis $\beta_p = 0$ and alternative hypothesis $\beta_p \neq 0$. The t-statistic on the parameters is $\frac{\hat{\beta}}{SE(\hat{\beta})}$, where the heteroskedastic robust standard error is:

$$SE(\hat{\beta}) = (X'X)^{-1} \left(\sum_{i=1}^n u_i^2 x_i' x_i \right) (X'X)^{-1} \quad (2.14)$$

Testing the significance of the betas is a powerful tool to assess whether there might be a certain correlation between the variables. However, the values of the betas and their standard error depend also on the selection of the model. Furthermore, whereas the p-value on the test might be very small, this does not prove causality. In fact, correlation does not imply causality and the study of causal relationship require a more complex understanding of the problem.

Finally, in case of homoskedastic standard error, the variance of the error is independent of the output variables and the formula has a simpler formalization. The homoskedastic error is usually smaller than the heteroskedastic one. R by default computes the homoskedastic errors. To adjust for it I used the `ase` package written by prof. Giuseppe Ragusa that contains the `sandwich` package.

¹¹Unbiased estimator with lowest variance. Note that with homoskedasticity we assume that the variance of the error term u_i is independent on X_i . With heteroskedasticity we relax this assumption.

2.4 Shrinkage Methods for Linear Regression

Shrinkage methods for linear regression add a penalty parameter on the estimated coefficients. What they do is to maximize a new objective function that depends not only on β but also on the tuning parameter λ . Recalling (2.2) the tuning parameter reduces the column space of the regression matrix S and it decreases the effective degrees of freedom. Consequently shrinkage models have a lower complexity than common linear models. The effect of this is a lower variance and an higher bias. The higher bias is because the new β do not exploit all the in-sample information and it is different from the OLS estimators. The lower variance is because the model is less flexible. When the gain in terms of variance is higher than the loss in terms of bias, the shrinkage method outperforms linear regression in terms of predictive power.

Ridge regression finds the beta by minimizing the following objective function:

$$\beta^{Ridge} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)'(Y - X\beta) + \lambda\beta^2 \rightarrow \beta^{Ridge} = (X'X + \lambda I)^{-1} X'Y \quad (2.15)$$

This result was first developed to control for problems of singularity of the matrix $X'X$. In fact, in case of many variables it may happen that two or more columns of the matrix are linearly dependent leading to no OLS estimation. By using the singular value decomposition of the matrix $X = UDV'$, where U and V are orthonormal matrices spanning one the column and the other the row space of X and D is a diagonal matrix with entries d_j , using the formula $EDF = \operatorname{trace}(S)$, with $S = X(X'X + \lambda I)^{-1} X'$, it is possible to show that $EDF = \sum_1^j \frac{d_j}{d_j + \lambda}$. That is, EDF depends negatively on the size of λ . With $\lambda=0$, $\beta^{Ridge} = \beta^{OLS}$, with $\lambda \rightarrow \infty$, $\beta \rightarrow 0$

The main feature of ridge regression is that it shrinks towards zero all the parameters but none of them is set equal to zero. This property comes from the nature of the penalty component $\lambda \beta^2$.¹² Consequently, to be able to select just some of the variables in the covariate matrix X , the constrain is set to be linear, such that the new objective function becomes:

$$\beta^{Lasso} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)'(Y - X\beta) + \lambda|\beta| \quad (2.16)$$

Lasso regression is a valid alternative to many model selection. On the other hand it may perform better or worse than ridge regression depending on the problem. Consequently an alternative model is to weight by a certain coefficient α the two penalty parameters

¹²See Friedman, Tibshirani, Hastie, The elements of Statistical Learning, pag. 71 for a comprehensive explanation of this point.

such that:

$$\beta^{ELNet} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)'(Y - X\beta) + \lambda(\alpha|\beta| + (1 - \alpha)\beta^2) \quad (2.17)$$

Finally, note that the magnitude of the beta depends on the unit of measure. Consequently the covariate matrix must be standardized to give to each variable the same weight. The process of standardization consists in subtracting the mean and dividing for the standard deviation. Dummies can be standardized in a more complex procedure.

2.4.1 How do you choose lambda?

A very difficult tool is to pick the best lambda to maximize the prediction power of the model. This seems a very complex optimization problem that depends on how we estimate the prediction power of the model. A common way to choose lambda is to use cross validation, either K folds or leave one out (which is much less computing intensive, given the formula 2.10). The procedure is the following: set lambda equal to a certain sequence, large enough to find the optimal. You train your model on $k - 1$ folds for each lambda in the sequence, you make all the predictions and repeat the process k times. Finally you pick the lambda in the sequence with the smallest MSPE. There are many packages in R that do this for you. The one that I used is glmnet, setting sequences of different size, between the order of 10^{-4} to 10^4 and with $k = 5$. An example of the behaviour of the MSPE as a function of lambda is shown in the next two graphs:

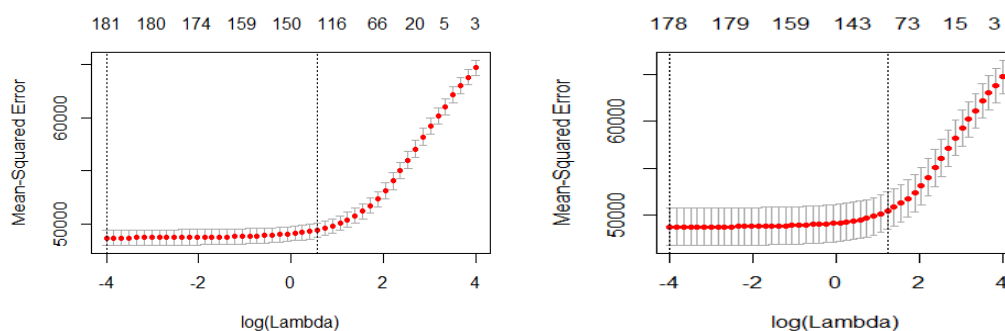


FIGURE 2.1: Lasso: Cv with standardized and Non Standardized Covariates

2.5 Stepwise

Before entering in the details of the algorithm I start from this useful definition of an additive model to predict y_i :

$$g(x_i, \theta) = \sum_{j=1}^k \beta_j f(x_i, \gamma_j) \quad (2.18)$$

With $\theta_j = (\beta_j, \gamma_j)$. For a linear regression function $f(x_i, \gamma_j) = x_{j,i}$, while in other cases such as basis expansion models the function can be much more complex. Given $\Lambda(y_i, f(x_i, \theta_j))$ a certain loss function we would like to solve:

$$\theta = \underset{\beta, \gamma}{\operatorname{argmin}} \sum_{i=1}^n \Lambda(y_i, \sum_{j=1}^k \beta_j f(x_i, \gamma_j)) \quad (2.19)$$

Unfortunately this is a very complex problem. There are many short-cuts to avoid this impossible computation. An approximation can be done by fixing a certain function f and use a stepwise algorithm. In this case the problem becomes:

$$\theta = \underset{\beta, \gamma}{\operatorname{argmin}} \sum_{i=1}^n \Lambda(y_i, f_{j-1}(x_i) + \beta_j f(x_i, \gamma_j)) \quad (2.20)$$

where $f_j = f_{j-1}(x_i) + \beta_j f(x_i, \gamma_j)$. This method can be applied to many different functions. Stepwise can be both forward and backward. Backward finds a full model exploiting all the in-sample information, and then it drops information until it reach the minimum of the loss function. Forward does the opposite, starts from an unconditional prediction, and it adds parts of in-sample information until it reaches the minimum error. The results may be different because the process depend on the order. Each time the model finds the covariate that most minimizes the loss and it adds this covariate to the model to recompute all the betas. ¹³ The algorithm that I used for stepwise minimizes the AIC. The package that I used in R is named MASS.

2.6 Generalized Methods of Moments

GMM is a common practice to estimate parameters using moment conditions. For a certain parameter vector $\theta_0 \in \Theta \subset R^p$, and a set of iid random vectors $w_i \subset R^L$ a certain function $g(w_i, \theta)$ satisfies $E[g(w_i, \theta_0)] = 0$. The classic example is for linear

¹³It may be used a slower version named forward stagewise, that does not update the previous betas to introduce a slower learning mechanism in the process.

regression, for which each column of the covariates matrix is orthogonal to the error such that the moment condition is $E[X_p'(Y - X\theta)] = 0$ for each covariate p in the column space of X . With the analogy principle we substitute expectations with sample averages. The GMM estimator minimizes a quadratic form of this function [51]

$$\theta^{GMM} = \underset{\theta}{\operatorname{argmin}} \left[\frac{1}{N} \sum_{i=1}^n g(w_i, \theta) \right]' W \left[\frac{1}{N} \sum_{i=1}^n g(w_i, \theta) \right] \quad (2.21)$$

With W $L \times L$ symmetric and positive defined weighting matrix. The main feature of a GMM estimator is the efficiency. In fact the weighting matrix is chosen by minimizing the variance of the estimator. It can be shown [51] that under the assumption of weakly independence the best weighting matrix is

$$\hat{W} = \operatorname{var}(g(\theta))^{-1} = E[g(\theta)g(\theta)']^{-1} \quad (2.22)$$

If moment condition are greater then the number of parameters, under the assumption of iid, $\frac{1}{N} \sum_1^n g(w_i, \theta)$ converge to a normal distribution by the central limit theorem. Consequently the objective function, product of two normals, converge to a chi-squared distribution with $k - p$ degrees of freedom. In this sense, data can be tested if they fit well comparing the minimized value of the objective function under the null hypothesis that it is equal to zero. A very high value (J statistic) would show that moments conditions are wrongly specified and the model does not fit well the data. In chapter 5 J-test and GMM will be extremelly useful to test the optimality of the prediction of Atac under unknown loss function.

2.7 Basis Expansion and Local Regression

Basis expansion is a common method to go beyond linearity. Basis expansion for additive models is based on the definition:

$$g(X) = \sum_{j=0}^k \beta_j f_j(X) \quad (2.23)$$

In this chapter we will break the range of X_p in non-overlapping pieces and we will treat $f_j(X) = I(o_j \leq X_p \leq t_j)$, a function that depends on a given range of X . For seek of brevity I will provide basic notions just about splines and smoothers.

2.7.1 Splines

We define k_j a knot on a point j on the range of X_k . Knots can be chosen arbitrarily but they are commonly set at the percentiles of X_k . The more the knots the more the flexibility. In a single covariate case, one of the easiest basis expansion is: ¹⁴

$$E[y_i|x] = \beta_0 + \beta_1 x_i + \sum_1^k \beta_{j+1} f_j(x_i) \text{ where } f_j(x_i) = (x_i - k_j)_+ \quad (2.24)$$

This model consists in finding a linear regression between each piece of x within two consequent knots. With a linear piecewise regression function the model is not differentiable on the knots and it shows low flexibility. A more flexible model allows to fit local polynomials such that $f_j(x_i) = (x_i - k_j)_+^n$, with $E[y_i|x] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_n x_i^n + \sum_1^j \beta_{j+n} f_j(x_i)$. The higher n the higher the flexibility. A cubic spline fits cubic piecewise polynomials. Finally, natural cubic splines are cubic splines with an additional constrain: linear function beyond the two extreme knots. Natural cubic splines are function that solve the following objective function:

$$\min_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt \quad (2.25)$$

where lambda is the penalty parameter on the smoothness of the function, measured as the integral of the squared second order derivative. If lambda is equal to zero the function interpolates all the observations, while with lambda equal to infinity the argument is a straight line because no second order derivative is tolerated. A smoothing spline is the argument that solves this objective function- a natural cubic spline - and usually with a knot on each observation. Note that smoothing splines belong to the class of ridge regression and the way to compute the effective degrees is similar to the one showed in the previous section.

Finally, the choice of the EDF - or in an equivalent way of the best λ - for the best spline can be done using the same approach used for shrinkage methods in section 2.4.1. Usually for smoothing splines LOOCV (2.10) is used due to the easiness in computing the trace of the smoother regression matrix S_λ for any finite lambda. The package in R that I used is splines.

¹⁴ $(x_i - k_j)_+ = x_i - k_j$ if it is positive, zero otherwise.

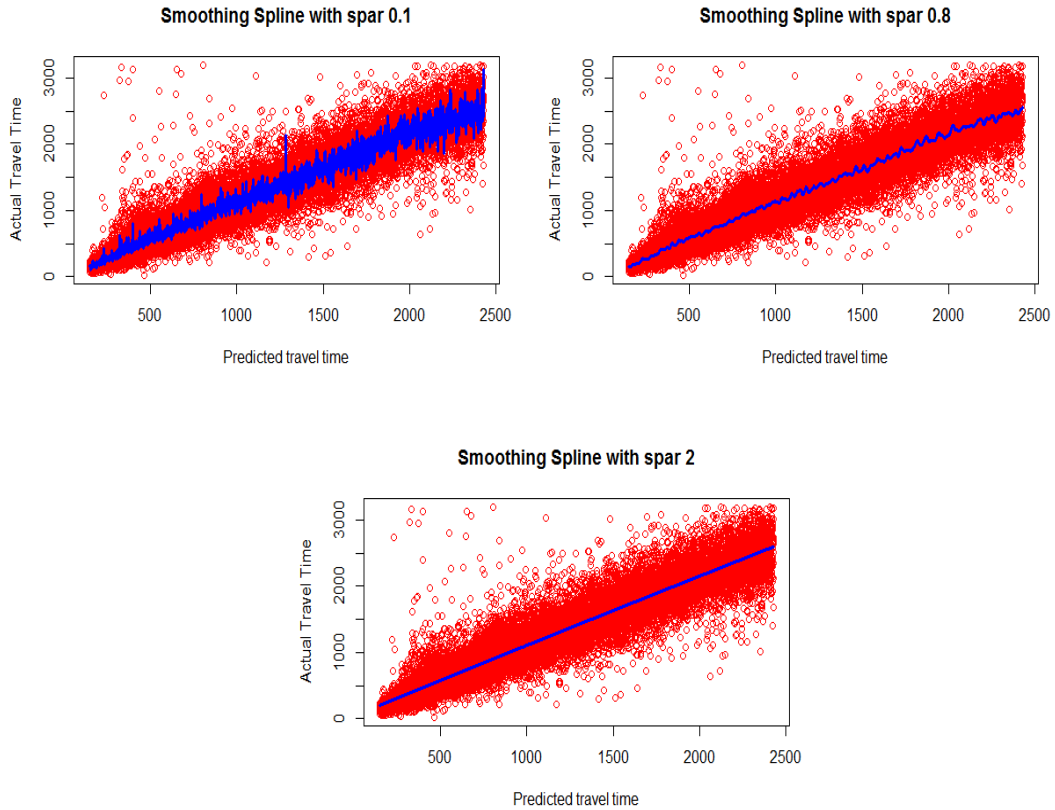


FIGURE 2.2: Smoothing Splines: examples with different lambdas

2.7.2 Kernel Smoothers

Suppose we use the following predictor:

$$f(x) = \frac{1}{N_k(x)} \sum_{j \in N_k(x)} y_j \text{ where } N_k(x) = \{i : \|x - x_i\| < d_{k,x}\} \quad (2.26)$$

with $d_{k,x}$ equal to a certain distance k conditional on x , $N_k(x)$ the number k of nearest points to x within the distance, y and x are the dependent and independent variables. This predictor is a K -nearest-neighbors: the prediction given a certain input variables x_j is equal to the average output variables of the observations i with x_i within a certain distance from x_j . Whereas this prediction often works very well for classification problems, it may lead to a discontinuous and ugly $\hat{f}(x)$ for continuous output variables when d is equal to a constant, especially when x has a high variance and the observations are not enough to cover all the range. Consequently, instead of picking the observations only within a certain bandwidth and weight all observations in this interval in the same way, different weighting functions may lead to continuous and differentiable predictors. We define $K_h(x_i, x) = W\left(\frac{x_j - x}{h}\right)$ a local kernel. In (2.34) $W(t) = 1$ if $|t| < 1$, 0 otherwise,

with t equal to the inside function of W . A different example is $W(t) = 1 - |t|^2$ if $|t| < 1$, 0 otherwise and so on. A locally weighted average solves:

$$\hat{f}(x) = \operatorname{argmin}_{\mu} \sum_{i=1}^n K_h(x_i, x)(y_i - \mu)^2 \quad (2.27)$$

While linear local fits solve:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \sum_{i=1}^n K_h(x_i, x)(y_i - X\beta)^2 \quad (2.28)$$

Polynomial fits or other functions may be used for Kernel Smoothers.

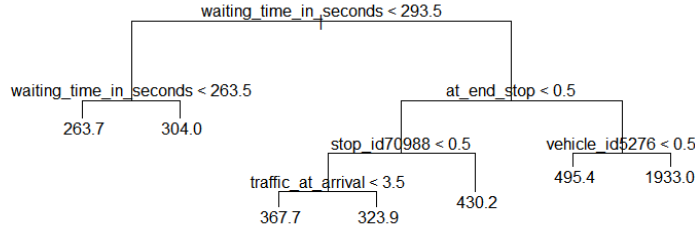
2.7.3 Course of Dimensionality

Course of dimensionality is a crucial problem for local regression (and classification). Local regression fits a certain regression function estimated locally in a given bandwidth of x . Suppose now that we want to move from one covariate case to a two covariate case. Whereas in the first case the bandwidth is computed in one single dimension, in the second case the bandwidth is in two dimensions and the distance is now the area within a certain radius from the i th observation. With three covariates we move to a three dimensional problem where the distance is within a certain 3 dimensional space and so on. As the dimension of the covariate matrix X increases, the likelihood of finding a certain number of observations in a p -dimensional bandwidth decreases exponentially. The effect is that, by keeping $N_{k,x}$ constant, we need to have much more observations to obtain the same performance of the predictor.

Finally, a key point is to standardize the distance of each covariate to make each x having the same weight. How to compute the distance for factors is an open question that I will not discuss in this thesis.

2.8 Regression Tree

Trees are a way to represent observations by grouping them into rectangular spaces. The idea is to find for each subset of observations the cluster that is able to minimize the deviance within each cluster, weighted on the number of observation in the subset, or in an equivalent way, to minimize the MSE. The MSE can be weighted for the number of observations in each single rectangle in order to avoid trivial splits that will easily lead to overfitting. Clusters are chosen by picking one single split on a certain covariate x_p . Going to my example of bus travel time, the first split showed in figure 2.3 is for



(2.29)

FIGURE 2.3: Pruned tree on sub-sample of Data

the observations with a predicted waiting time from Atac ¹⁵ less then 293.5 seconds and more then 293.5 for the other cluster. The algorithm to find the best tree is a stepwise algorithm, that add splits until when the MSE cannot go beyond a certain treshold or until when the size of a rectangle is small enough.

The general formula for a tree that minimizes the MSE with M partitions R_M is the following:

$$\hat{f}(x) = \sum_{m=1}^M d_m I(x \in R_M) \text{ with } d_m = \text{ave}(y_i | x_i \in R_m) \quad (2.30)$$

$I(x \in R_M)$ is equal to one if the observation belongs to the partition R_M and zero otherwise. Given the output covariate X_p for the jth observation and the split at s for this variable, the objective function is:

$$\min_{s,p} [\min_{d_1} \sum_{x_i \in R_1(j,s)} (y_i - d_1)^2 + \min_{d_2} \sum_{x_i \in R_2(j,s)} (y_i - d_2)^2] \quad (2.31)$$

This formula provides a formalization of the problem, where for each split, the algorithm finds the minimum for a certain variable p above and below the threshold s of the sum of the squared residual in each cluster, where the prediction d for each cluster is the argument that minimizes the sum of squared residuals within the cluster. It is trivial to show that d is the average in each subgroup of observations.

To avoid overfitting a common practice is to use prune backing. The idea is to grow a large tree to reach the maximum complexity. Then to reduce the number of nodes pruning back. A loss function with a new penalty parameter that depends on the complexity of the tree is computed. We define N_m the number of observations in each rectangle, $|T|$ the number of terminal nodes in a certain tree, that is the number of

¹⁵The data showed are built by fixing C= 5 minutes. Go to chapter 4 for a clear explanation.

rectangles at the bottom of the tree and $MSE_m = \frac{1}{N_m} \sum_1^m (y_i - \bar{y})^2 \forall y_i \in R_m$. The objective function to minimize (cost-complexity criterion) is:

$$C_\gamma(T) = \sum_{m=1}^{|T|} N_m MSE_m(T) + \gamma|T| \quad (2.32)$$

Gamma introduces a penalty parameter for the complexity of the model and the best gamma can be choose adaptively using cross validation (also in this case the problem becomes a generalized shrinkage regression). T_γ is each subtree within the overfitted tree T_0 . The package in R that I used for tree is tree(no surprise!).

2.8.1 Bagging

Bagging is a common practice used not only for regression trees but also in many other cases. Bagging exploits the concept of bootstrapping to reduce the variance of an estimator. In fact, it picks with replacement n observation from the initial sample of size n and builds a new tree on this sample. It does so several times and the final prediction is the averaged prediction of each tree built on each new sample. By increasing the number of trees it artificially increases the number of observation, with the assumption of perfect randomization of the initial sample. Consequently, the new predictor has a lower variance. Unfortunately, it can be shown that the reduction in variance cannot be beyond a certain threshold. For this reason bagging should be done by compromising the reduction in variance and the time cost in running the process.

2.8.2 Random Forest

It is likely that trees built on bootstrapped samples are very similar each other. Say we have 1000 observations with 30 covariates and one of this is particularly important for a predictive model. We would expect that each tree will use this covariate in one of the first splits and each tree will be similar each other. This similarity has a negative effect on the reduction on the variance of the predictor. In fact, we would like to have unbiased but different trees on each sample to have a great variability of predictions and a low variance on the outcome. For this reason Random Forest adds an additional constrain. For each split of each tree in each sample it randomly picks a subset of covariates and it constrains the tree to use only that subset of covariates for the split. The random choice is repeated for each split. The final result is a decrease in the correlation between trees of different samples. The number of variables to pick from the p covariates may vary but empirically it is usually used \sqrt{p} . Also the number of trees may vary. The package randomForest in R uses 500 trees as default.

The out of bag MSE is a valid alternative to cross validation for any process that uses bagging. Every time that you pick n observations with replacement from your initial sample you leave on average $1/3 n$ observations in the initial sample. These observations can be treated as a test set for the predictors built on the new sample. The out of bag error rate consists in testing the trees that have not been trained on the k observations and picking the mean squared error of the average of these k predictions. The process is repeated for all the observations in the sample and the out of bag error rate is the average of the mean squared errors. Note that the OOB error overestimates the real mean squared prediction error because it uses only a subset of predictors. Furthermore, the size of the positive bias of the OOB is usually greater than the one for the k -folds CV MSPE.

Chapter 3

Description of the Problem

3.1 Transportation system in Rome

The agency ATAC is a company owned 100 per cent by the municipality of Rome. The company manages, together with Roma TPL, all public transportation in Rome.

Atac owns more than 2700 bus, 165 tram, 30 filobus, 83 metropolitan trains and 88 trains for normal railways. The transportation system has in total 377 lines. Of these 377, 326 are bus routes, 6 are trams, 11 are trains, 4 are metro(A,B,C) or metro deviations(B1), 30 are bus by night.

Figure 1 reports the metro system in Rome. There are in total three metro lines, named A, B and C that cross the whole city, with a total length of 60 km. Metro B has a deviated line, named B1 from stop Bologna to stop Jonio. The metro works all the day from 5:30 am to 11:30 pm. On Friday and Saturday it works until 01:30 am. The intra-city trains work from 5:30 am to 22:30 or 23:30 depending on the route.

The low accessibility of the transportation system is a key issue in Rome. Pinelli et al.[33] carried out an interesting study about the accessibility of the transit system in Rome using bus GPS traces. The researchers developed an agent based algorithm to simulate human mobility in order to study the accessibility from different locations during different hours of the day. According to their results, accessibility increases in the vicinity of metro stations and it significantly changes during the time of the day, with a peak at 8:00 am.

Given the poor extension of the metro lines, the bus system plays an important role for the mobility within the city. In this sense, the distribution of the vehicles and the number of routes are important for the accessibility to the transit system.

The main trade-off in the allocation of vehicles and routes is between passengers demand, that may be not uniformly distributed on the territory, and accessibility from

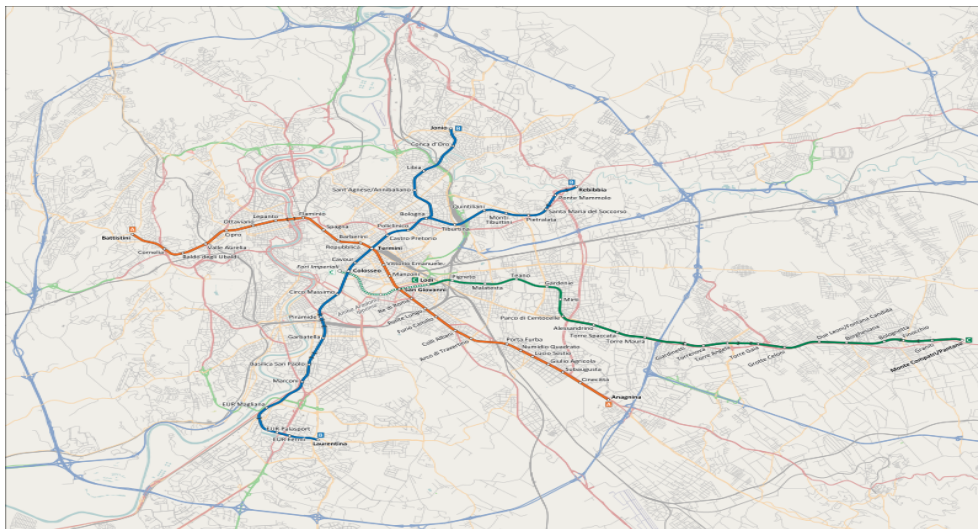


FIGURE 3.1: Metro Lines

all the points of the city. In fact, although the stops are uniformly distributed on the territory, the number of vehicles in use may be different for different parts of the city.

3.2 Potential variables affecting on-time-performance in Rome

As already showed in chapter one, on time performance may be affected by several variables. In the literature particular attention has been given to the deviation of travel time from scheduled travel time as a measure of on-time-performance. In this sense, traffic conditions, number of stops and number of intersection may increase travel time. The number of passengers may affect boarding time with an indirect effect on travel time.

Weather condition is a further variable related to travel time. Rain in Rome would incentive the usage of car and it may increase the traffic. Moreover, it could cause dangerous conditions in the streets, increasing the likelihood of accidents. Finally, rain would decrease the speed of buses increasing the in-vehicle travel time.

Weather prediction can equally have an impact on traffic condition, increasing the number drivers with car.

Driver behaviour may be a further variable correlated with schedule deviation. Drivers who do not respect scheduled departure from end stops of a route can cause unpredictable delays.

Segment distance and accrued delay are further variables that can determine future delays. Many authors noticed that delays at the upcoming stops depend on accrued

delays at past stops. Bertini et al.[16] showed that as the segment distance between two stops increases, the correlation between the delay at the two end-points of the segment decreases due to schedule recovery.

3.3 Scheduled Arrivals

In the next section I will show some statistics about arrivals of buses. ¹ In Rome, there

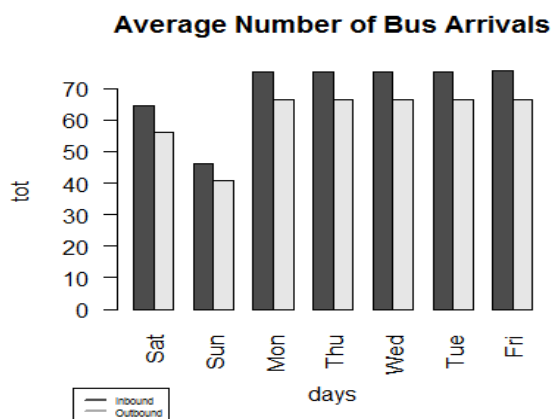


FIGURE 3.2: Mean Number of Arrivals per stop

are in total 8718 stops and 377 routes. The number of scheduled bus arrivals per day varies depending on the day. In fact, Atac has not a dataset with a fixed number of arrivals per each day of the week, but it has a different number of scheduled buses for each day of the year. On the other hand, according to my analysis, this number mainly depends on the day of the week and on holidays. In the previous graph I reported the average number of bus arrivals on 400 random stops between April and May 2016. ² The two different bars represent the number of bus arrivals per direction (inbound and outbound). The inbound number of buses is more than the number of outbound. One possible reason is because some routes have a unique direction and they are given the value one. On the other hand, I cannot prove this statement. A second reason, which seems less reasonable to me could be related to the fact that most of the buses may perform the first and/or the last trip towards the center of the city, but there is neither a proof nor available data for this statement.

¹For a more accurate statistical description of the bus service go to the next chapter. The following statistics are computed using the open data of Atac available at <http://www.agenziamobilita.roma.it/it/progetti/open-data/dataset.html>.

²In this report 25th of April and the first of May, both holidays were not considered.

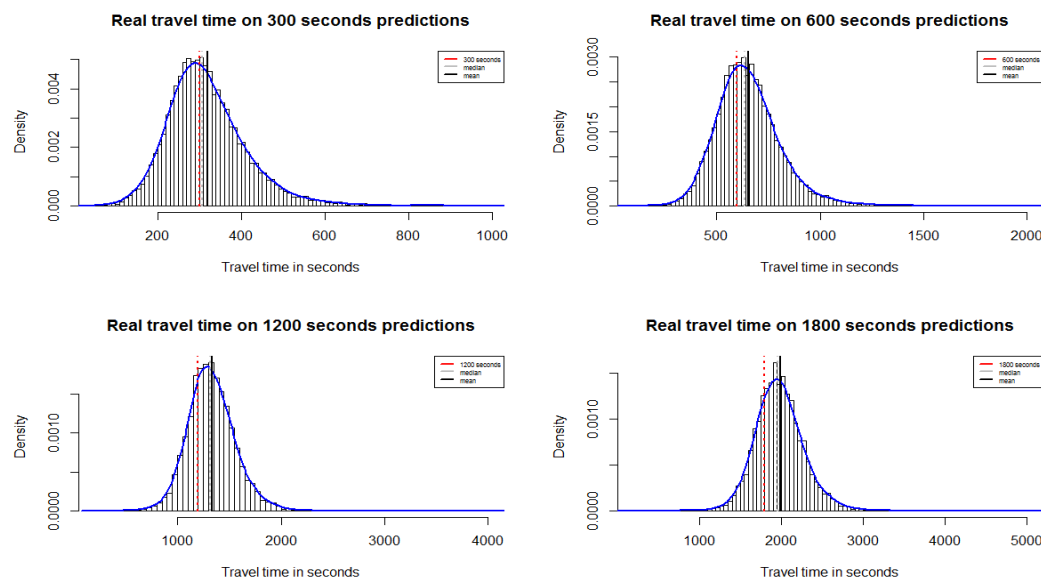


FIGURE 3.3: Distribution of Travel time on 5,10,20,30 minutes prediction

3.4 Real-time prediction: Muoversi a Roma

Muoversi a Roma is the application of Atac that makes real-time predictions of bus arrivals and compute the route from two given points of the city. To the extent of my knowledge, prediction of bus arrivals use either an historical average of bus speed at previous stops or actual speed of the vehicle and space distance. Figure 3.4 was printed from a presentation done by Atac and it is the unique source that displays this information.³

In figure 3.3 I show the distribution of arrivals on 5, 10, 20, 30 minutes prediction at 26 different stops uniformly distributed in Rome. The data have been collected between the 16th of April 2016 and the 5th of May 2016. A comprehensive description of this data is in the next chapter. The prediction accuracy does vary according to the predicted time distance. For each data set the prediction of Atac shows a performance close to the in-sample performance of the ex-post mean of arrival time. The higher the expected travel time distance, the higher both the variance of travel time and the MSE of the prediction. One reason can be that more time may increase the likelihood of external factors that affect the travel time.

³See: <https://bitbucket.org/agenziamobilita/muoversi-a-roma/downloads/PresentazioneCercaPercorso.pdf>

- Cost for **bus waiting** edges:
 - Waiting time for catching first arriving bus, if real-time data available
 - Average waiting time from historic data or schedule, otherwise
- Cost for **bus ride** edges:
 - Use traffic speed, if real-time data
 - Use historic speed, otherwise

FIGURE 3.4: Prediction of Atac

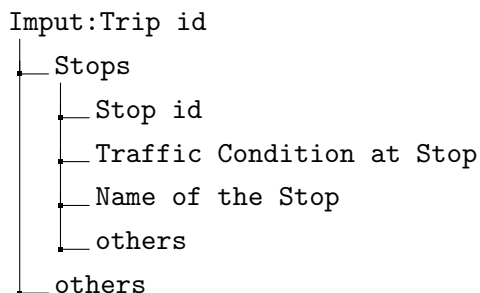
3.5 Atac Open Data

The real time data are available through an API system. It is possible to use many different inputs to get several information. By specifying the route the system provides information about the id of each stop in that route, the traffic condition at each stop and other variables. With the id of the stop as input, the API report information regarding the condition of the stop - if it is activated or not, if there is a footboard at the stop - and the many values for each bus arrival predicted by the AVL system. The data are in form of a dictionary. For each bus , it reports the identity of the vehicle, the route of the vehicle, whether on the vehicle there is a cold air system and a system to buy ticket on board, whether the vehicle is at the end stop in that moment, the departure time from the end stop, a boolean value if the autobus is next to the stop , the number of stops between the actual position of the bus and the stop, the predicted waiting time both in minutes and seconds and a variable, named banda , that has no description but should be related to the velocity of the trasmission of the signal.

```

Input: Stop id
├── Boolean footboard
├── Bus Arrivals
│   ├── Route id
│   ├── Vehicle id
│   ├── Boolean next to the stop
│   ├── Number of stops missing
│   ├── Boolean Air
│   ├── Waiting time
│   ├── Waiting time in second
│   ├── Banda
│   ├── Boolean MEB
│   ├── Boolean Moby
│   ├── Boolean at end stop
│   └── others
└── others

```



Whereas these are the core variables, there are many other possible trees and combinations of inputs and outputs.

3.5.1 Static Data

I will describe the data reporting information about static information of the bus service system. The data frames available are the following:

- Calendar Dates:
 - Service id: the id of the service (each service is unique for a given trip and day).
 - Date : the date in year, month, day.
 - Boolean Exception: whether there is an exception of the service in that day.
- Stop Times:
 - Trip Id: the id of the trip, that is unique for a given route and direction.
 - Arrival Time: scheduled arrival time at a given stop for a given service.
 - Departure Time: scheduled departure time at a given stop for a given service.
 - Stop id: the specific id of the stop.
 - Stop sequence: the sequence of the stop in the trip.
- Trips.
 - Trip id.
 - Service id: unique for a given date and trip.
 - Direction id: direction of the trip.
 - Shape id: shape that reports the geographic shape of the trip.
- Stops.
 - Stop id.

- Stop Name.
- Stop Latitude.
- Stop Longitude.
- Location Type: side of the location.
- Parent Station: whether the stop is in a bigger station.
- Shape id.
 - Shape Latitude: the latitude of one of the points in the shape of a given trip.
 - Shape Longitude: the latitude of one of the points in the shape of a given trip.
 - Shape Sequence: the position of the point in the shape.
- Routes
 - Route Id.
 - Route Type: type of the route (metro, bus, tram, etc.)
 - others

In the next chapter I will describe how I merged these datasets to get further variables and which kind of real time data I collected. Note that there are further static available.

Chapter 4

Description of the Data

The core part of this thesis relies on the collection of data. During the first weeks I manually collected data with Python to understand the potential problems to deal with. Then, I wrote a script in R to query 26 stops every ten seconds and to convert the Json file in a nice matrix reporting all the variables of interest. The matrices were saved as text files. I chose 26 stops because it was the optimal number of stops to make the process run within the ten seconds. The process was in parallel on 15 cores(out of 16 available cores). The stops were uniformly distributed on the territory of Rome and they were picked from 4 different trips of the following buses: 92 , direction inbound, 280 , direction inbound, 98, direction outbound, 671, direction outbound. The routes through these 26 stops were in total 54. The period of data collection was between the 16th of April and the 5th of May 2016.

It is worth to mention two problems: bad connection to the server of Atac, treated using an if statement in case of error; four different interruptions of the execution of the code due to fatal errors, with reactivation of the machine few hours later. The second phase

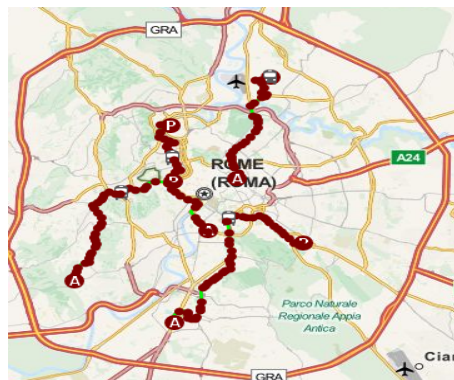


FIGURE 4.1: The four routes analyzed

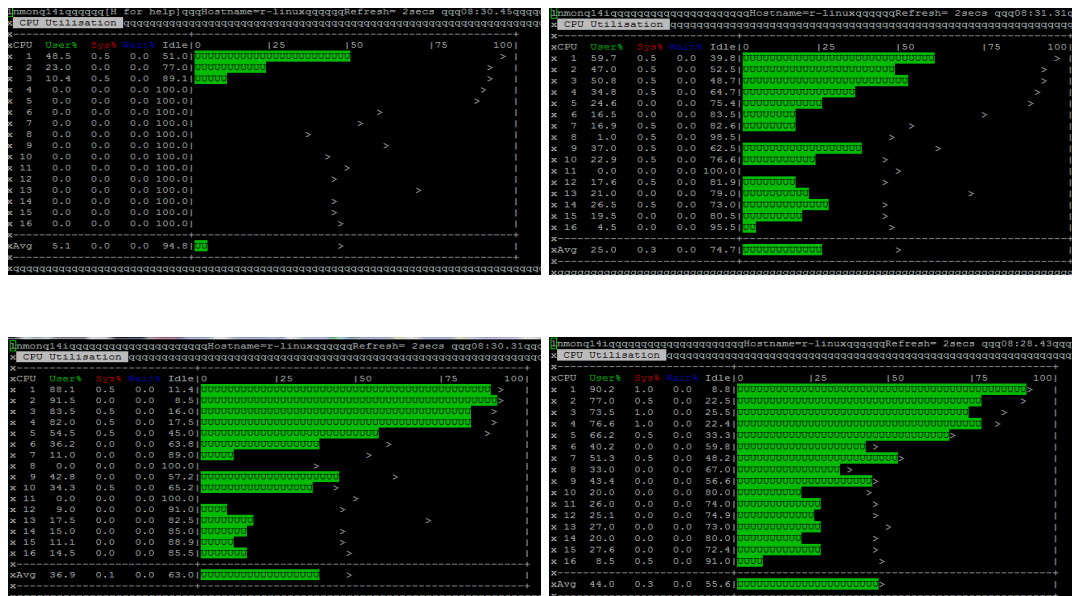


FIGURE 4.2: CPU Usage

of this process was the construction of the final dataset from all the files. Each step is carefully described in the next sections. The packages that I used were XMLRPC to work with API and foreach, DoSNOW, parallel to implement a parallel execution of the code in R.

4.1 The queries

The output was in form of a dictionary containing single elements, lists or other dictionaries. Furthermore, the structure of the Json file could change depending on peculiar situations. The number of bus arrivals had a random length for each single query. Finally, traffic condition was the only variable that required a separate process. The loop repeats the process every ten seconds for more than two weeks. For each k , it records the time of the query and it shares this information with all the virtual cores. In each core it is called in the library the XLMPRC package, it is saved the traffic function in RAM and it is created an empty list s_2 and this list is filled with the query on a certain stop S_i . The variables of interest for each query are inside the dictionary bus arrivals. Each observation is saved as a vector of variables reporting information about each bus arrival at each stop. The time of the query, the traffic condition and the id of S_i is appended to this vector. Finally all these vectors are appended by row to a matrix and this matrix is appended to a final matrix.

```

For k in 1:140000 do
  | Wait max(10 seconds - time of loop, 0)
  | try the token query
  | if token is error
  |   | wait five second and go to 1
  | time = actual month,day,hour,minute,second
  | finalobject is: do in parallel
  |   | Create and empty object: MATRIX = NULL
  |   | for each STOP
  |   |   | s2 = query on STOP
  |   |   | initialize the accumulator: set x equal to 1
  |   |   | while x is less then 1000
  |   |   |   | if is null the xth element of s2
  |   |   |   |   | exit the loop : x = 1001
  |   |   |   | take the values of the xth bus arrival of s2
  |   |   |   | Traffic Function(xth trip of s2 values)
  |   |   |   | xx = append STOP id, traffic, time, xth values
  |   |   |   | append by row xx to MATRIX
  |   |   |   | set x = x+1
  |   |   | bind all matrices by rows
  |   | write finalobject

```

Traffic Function is described in the appendix.

4.2 Construction of the Dataset

After the collection of Data, I was interested in predicting travel time between two given points adjusting for problems of correlation between observations. The first dataset that I constructed was a dataset with all the vehicles arrived at the 26 stops that I observed. I extrapolated from each text file, all the observations having the dummy variable “next to the stop” equal to one and I picked the same observation for same vehicle id, same day, time window and stop id with smallest ex-ante prediction of Atac (usually between ten and 0 seconds). In this sense I obtained all the actual arrival time at the 26 of each vehicle with no doubled observations. For a comprehensive understanding of the process see the code in the appendix.

4.2.1 The process

The main concern during the process was to avoid overlapping for same time and same vehicle. I will make this point clear with an example: if bus with id vehicle 44398, was observed both when his distance from stop 123 was 3 km on Monday 16 of April at 12:00 am and in the same day at 12:10 with 2km distance from the same stop, then these two observations perfectly overlap (one is “inside” the other). On the other hand, fixing the distance to a constant - like picking all observations with expected time distance equal to 300 seconds- hugely reduces the in-sample variance of the observations and decreases the amount of information that we might be able to analyze. For this reason , to avoid overlapping I did the following: after storing all the actual time of arrival for each bus at each stop, I clustered all the arrivals of the same day, same hour, same vehicle, same route and same stop of arrival. For example, in each cluster you can have all observations of vehicles 1243 going to stop 111 on the 16th of April between 2 and 3 pm. For each cluster I picked only one random observation. Then I merged the actual arrival data set and this new data set controlling, for day, vehicle id, stop id and time window.¹ Two variables of this dataset were time of arrival, named time.x, and time at the moment of the query called on the observation in the cluster , named time.y. From this new dataset were dropped all the observations with time.x lower then time.y - observations wrongly matched. For all duplicated observations with same identity of the vehicle, same stop id, same trip id and same time.x, I picked the one with the highest time.y in order to avoid duplicates in different time windows at the moment of the first query. Finally, for all duplicated observations with same identity of the vehicle, same stop id, same trip id and same time.y, I picked the one with the lowest time.x in order to avoid duplicates in different time windows at the moment of the query on the arrival ². With this process I tried to drop mismatched observations. See figure 4.3 for the statistics of travel time.

4.2.2 Variables

Once the dataset was built, additional variables were created. In this process I used the static datasets of Atac. The variables in the dataset were:

- id vehicle;
- id trip;

¹See the appendix for a comprehensive understanding of the procedure.

²Higher and lower time.x and time.y is intended in terms of the position in time: higher is after and lower is before in time.

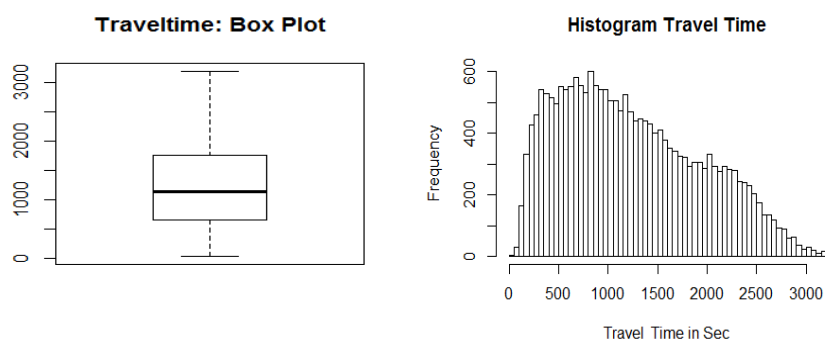


FIGURE 4.3: Distribution Travel Time in DataSet

- id stop;
- boolean cold air on the bus;
- boolean stop activated;
- boolean moby and meb: buyable ticket on board of the vehicle;
- Cartels at the stop;
- factor banda: velocity of the transmission of the signal;
- boolean footboard at arrival stop;
- route id;
- traffic state at the stop of arrival and at the starting point;
- predicted waiting time in seconds from initial position;
- number of stops to pass through from initial position named “missing stops”;
- Boolean at end stop: whether the bus is at the end stop at the moment of the query on the initial position;
- traveltime: difference in seconds between time.x and time.y plus predicted waiting time in seconds from the final position of the bus to the stop;³
- hour of the day: In which hour is the observation;
- day, specific day of the year;
- day of the week , 7 levels;

³Predicted waiting time was around 10 seconds.

- Boolean direction, weather the bus goes inbound or outbound. ⁴
- Weather: weather during the day;
- Weather predicted: predicted weather for the next day. ⁵

4.2.3 Why picking only a sub-set of observations?

There could be many ways to analyze the data that I collected. What I did is not the unique way and further research should be developed in this sense.

I define “time point” a unique vector on a given space and time and “time segment” a finite continuous interval of time points. The endpoint of a time segment is always a given stop. Travel time is the time difference between two endpoints of a given time segment. My claim is that for same vehicles, the higher is the portion of time segment shared between the two observations, the stronger the correlation on the travel time of the two. I will not prove this result, but this is a trivial result if the two time segments overlap for the same vehicle. In this case the two observations are the same.

More generally, the higher is the area of the intersection set of the two time segments for same vehicles, the higher is the set of information in common between the two observations. The problem is that this set of information is over-counted because it is present not one time but n times, for each vehicle with a same portion of travel time. This is a problem of overlapping and to avoid it, I fixed a length C of a time segment, different for each cluster, and I picked only one vehicle with each possible endpoint of a certain time segment with the time segment length closer to C . By doing so each vehicle is detected only once for a same time window. On the other hand, different time segment length may present different features. For example, driver speed may have a low effect on short trip and an higher effect on longer trip. To have enough variance in observed travel time C was a random variable uniformly distributed between 3 and 39 minutes for different clusters.

4.2.4 Critical points

The first critical point of this procedure is that buses in two different but subsequent hours, say 12 am and 1 pm are not in the same cluster. This means that buses at the

⁴To compute this variable I used the latitude and longitude information for each stop and I matched stop id, trip id and shape id by finding the shape containing the point with the smallest euclidean distance from the stop.

⁵Both these variables were manually added. Further work should exploit API services for weather to have better accuracy.

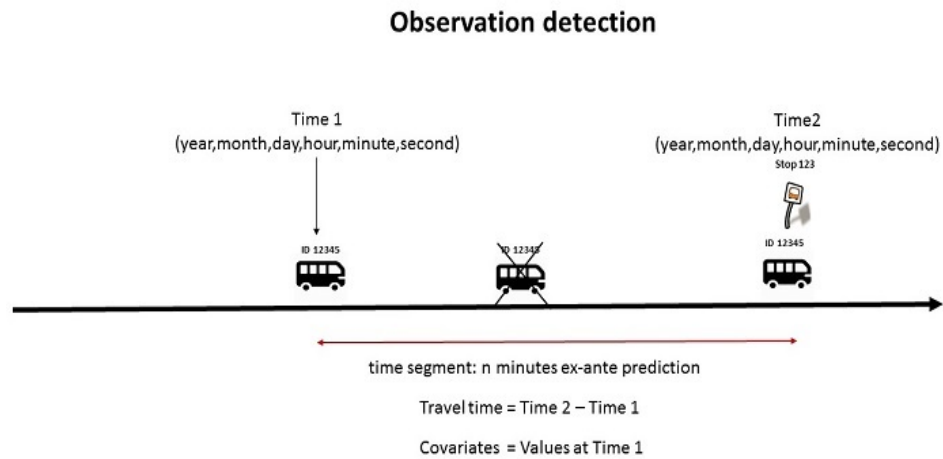


FIGURE 4.4: Travel time Estimation

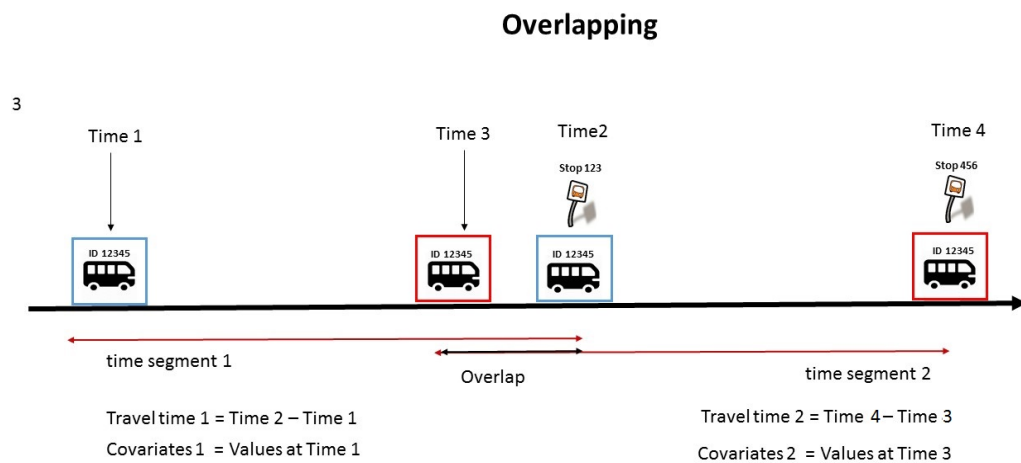


FIGURE 4.5: Overlapping

boundaries may be present in the data frame more than once. Moreover, the data set has only a subset of observations (around 20000) and strongly depends on the information provided by Atac. Buses that do not work anymore but are still around the city with the GPS system on, are detected as normal buses from the AVL system and this can create a bias.

One further critical point is that for longer time travel, the set of shared information between variables increases. In fact, same vehicles with different stops of arrivals may share a certain portion of their time segment if the distance between the two stops

is less than the travel time. This may reduce the out-of-sample prediction power of the model and may create a certain bias for the Cross Validated MSPE.⁶ A possible solution to this problem is to manually drop the observations with common time segment length higher than a certain threshold. Alternatively to adjust for the bias on the MSPE I can compute the MSPE on a test set obtained by running the process few days later. Anyway, the stops have been chosen far away each other and only four routes have many stops queried. Consequently few observations show a problem of overlapping.

Finally the last critical point regards the strong assumption of reliability of the real time information provided by Atac. This assumption may be weak in some circumstances and further investigation seems necessary for future research.

⁶Because the train and test sets may share common information.

Chapter 5

Forecast Optimality under Flexible Loss

In this chapter I will test the optimality of the forecast of Atac without imposing strict restrictions on the shape of the loss function. Most of the literature only considers the mean squared error as a loss function. The reasons are many: it is a symmetric loss (negative and positive errors have the same weight), it is differentiable and easy to minimize. But for this problem we want to relax the assumption of a symmetric loss. In fact, whereas a predicted travel time shorter than the actual travel time of the bus increases waiting time at the stop, it decreases the likelihood of losing the bus when it comes. On the other hand, a predicted travel time greater than actual travel time can induce the consumer to go to the stop after the bus has already gone. In this sense, the objective of this section is to estimate the loss function given the predictions and to test the rationality of the forecasts, under a general class of losses.

The estimation of the loss unknown to us is done using generalized method on moments on the empirical data. The main references for this chapter are Elliot et al.[15] and Patton et al.[31,32]. The description of GMM is in chapter 2. The package used in R is gmm.

5.1 Generic Definition

A generic definition of loss function can be summarized by the following formula[15]:

$$\Lambda_i(\rho, \alpha, \theta) = [\alpha + (1 - 2\alpha)I(y_i - \hat{f}(x_i) < 0)]|y_i - \hat{f}(x_i)|^\rho \quad (5.1)$$

where $I(\text{boolean}) = 1$ if true, 0 otherwise. The main assumption in this case is that the loss depends on two parameters, ρ and α and $\hat{f}(X) = \theta'X$, conditional on a certain information set. Whereas the restriction to a linear model may be strict in particular circumstances, X is any subset in the information set. Although we may extend the class of loss functions by estimating a loss on an higher number of parameters ¹ this description already contains many symmetric and asymmetric losses commonly used. A further assumption is that the loss depends on the error term, that is $\Lambda = \Lambda(e)$ where $e = Y - \hat{f}(X)$. Whereas this is reasonable assumption for this problem, this is not always the case. For example many researchers showed that losses for GDP prediction may depend also on other parameters[32]. Finally, we will assume that our data are weakly independent. ²

5.2 Estimation of the loss

As we might expect the mean difference between travel time and the prediction from Atac is positive and equal to 110 seconds. In fact, as I will show in table 6.2 of the next chapter, for each estimated additional minute of travel time corresponds , on average, slightly more then one minute, and the final prediction has a positive constant term.

Assuming that $\hat{f} = \theta'X$, for some X in the information set, the forecaster is assumed to find theta by solving:

$$\underset{\theta}{\operatorname{argmin}} E[\Lambda(\rho_0, \alpha_0, \theta)] \quad (5.2)$$

For given values of α, ρ .

The optimal forecast error[15] is the distance between the best forecast and the actual value. If \hat{f} is the linear projection of Y onto X , the optimal error is orthogonal to each column of X , by the orthogonality decomposition theorem . This means that also \hat{f} - a linear combination of the column of X - and the error ϵ are orthogonal.

Given a vector of parameters to estimate $\gamma = (\alpha, \rho)$, the conditional nature of moments conditions implies that $E[wg(\hat{x}, \gamma)] = 0$, $\forall w \in W$, where W is a vector of instruments in the information set.

¹Patton et al. showed that it was possible to do this by estimating the derivative of the loss for theta using a smoothing spline with 3 knots, and then use GMM to find the values of each piecewise polynomial.

²The way the dataset has been built allow for iid assumption. On the other hand the weakly independence is a less strong assumption which is necessary and sufficient for this problem.

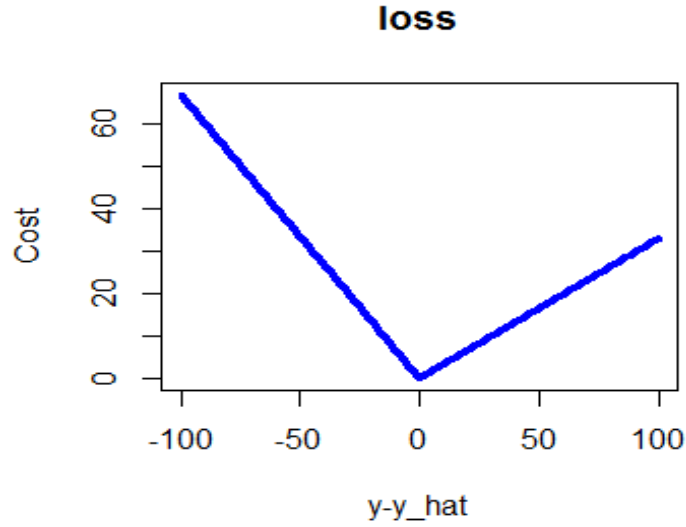


FIGURE 5.1: Loss Function Estimated with Moment Conditions

To find the two parameters α and ρ , according to Elliot et al., we do not need the whole information set, but we only need that the number of instruments k must be at least as equal as the number of parameters p to estimate. Consequently we define :

$$\lambda(\hat{f}, y, \gamma) = \frac{\partial \Lambda(\theta)}{\partial \theta}, W = (1, \hat{f}, \Lambda) \quad (5.3)$$

Where the moment conditions are $\forall w \in W$:

$$E[W\lambda(\hat{f}, y, \gamma)] = 0 \quad (5.4)$$

The orthogonality condition to λ holds for each instrument for the following reasons: the derivative of the loss is always equal to zero, also when multiplied by a constant; it is orthogonal to any $w \in W$; any function is orthogonal to its derivative³.

To conclude by assuming $\alpha \in [0, 1]$, $\rho \geq 1$, the system of functions to set to zero becomes[15] :

$$E[h(\gamma, W)] = E[W(I(y - \hat{f}(x) < 0) - \alpha)]|y_i - \hat{f}(x)|^{\rho-1}] = 0 \quad (5.5)$$

The problem is solved by replacing expectations with sample averages.

³See Appendix for a proof of these three moment conditions.

	GMM 2 parameters	GMM 1 parameter
alpha	0.350*** (0.003)	0.336*** (3.3e-03)
rho	1.000*** (0.000)	1.1
Observations	20,685	20,685
J-Test: degrees of freedom is	1	2
Test E(g)=0		
J-test	1.2966e+02	1.2964e+02
P-value	4.8700e-30	7.06e-29
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01	

TABLE 5.1: Generalized Methods of Moments: Results

5.3 J-test and robustness of results

There are many ways to test the optimality of the loss. A first test we could perform [15,32] is the J-test. The main requirement is to have an over-identified model - number of moments greater than number of parameters to estimate.

By defining $m(\gamma)$ the objective function to minimize, $m(\gamma_0) = 0$ under the null hypothesis, where γ_0 is the argument that minimizes the function. We can check whether $\hat{m}(\hat{\gamma})$ is close enough to zero. ⁴

$$Jstat = N \left(\frac{1}{N} \sum_{i=1}^n h(\hat{\gamma}, W) \right)' \hat{V} \left(\frac{1}{N} \sum_{i=1}^n h(\hat{\gamma}, W) \right) \rightarrow \chi_{k-p}^2 \quad (5.6)$$

Where V is the best weighting matrix. Under the null hypothesis the J-statistic should converge to zero, under the alternative it goes to infinity. A very high J-statistic may signal underfitting of the model. The J-stat should be trivially equal to zero for in-sample observations, while this result may be different for out-of-sample observations. Given the results showed in table 5.1 the J-test rejects the hypothesis of optimality even for in-sample observations. This means that either the moment conditions are wrongly specified or the algorithm is not able to converge. This last statement seems the most plausible. In fact, with a ρ equal to one the function at the kink is not differentiable. Consequently to check the robustness of this result for α we run a second regression setting $\rho = 1.1$. The results for alpha are similar as shown in table 5.1. Whereas the

⁴See section 2 for a comprehensive understanding of the objective function for GMM.

J-stat is still surprisingly high for in-sample observations, the alpha is less than 1/2 as we might expect but seems to have an opposite behaviour against rho.

5.4 Quantile Test on Optimal Forecast

This test was proposed by Patton[32] for the first time. The test is based on $I(y_i - \hat{f}(x_i) < 0)$ described in the previous section. This is a test for optimality under unknown loss functions with the assumption that the loss is homogeneous[32] and the data generating process has dynamics in the conditional mean and variance, or the DGP has dynamics only in the conditional mean and the loss function is a function of the forecast error.⁵ Under the null hypothesis of forecast rationality this variable should be orthogonal to any instrument in the information set. Using simple OLS, the results of this test are comparable to the ones obtained in the next sections, as shown in table 5.2. The prediction and the indicator are significantly correlated each other. Whereas this result may be biased if one of the assumption is not satisfied, it arises several questions about the optimality of the forecast under certain classes of loss function.

	<i>Dependent variable:</i>	
	indicator	$\Lambda'(\alpha = 0.33, \rho = 1.1)$
prediction	-0.0001*** (0.00001)	-0.0001*** (0.00001)
Constant	0.406*** (0.007)	0.072*** (0.012)
Observations	20,685	20,685
R ²	0.007	0.007
Adjusted R ²	0.007	0.007
Residual Std. Error	0.471 (df = 20683)	0.824 (df = 20683)
F Statistic	139.306*** (df = 1; 20683)	138.230*** (df = 1; 20683)
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01	

TABLE 5.2: Tests for optimality: indicator test and Mincer-Zarnowitz test

⁵A comprehensive description of these conditions goes beyond the scope of this thesis. For a comprehensive understanding see [31,32]

5.5 Generalized Mincer-Zarnowitz regression

This test is very similar to the indicator test, and it is historically used to test optimality under MSE. Using the derivative of our asymmetric loss allow for extending this test to a more general class of losses. In particular this is a test on the moment condition for which the derivative of the loss function is orthogonal to any instrument in the information set. Consequently we test whether $\Lambda'(\theta)$ has a certain correlation with the prediction. To run this test we set $\alpha = 0.33$ and $\rho = 1.1$. The result of this test is conditional on the assumption that the loss is a two-parameter loss function and the estimation of the parameters is correct. Whereas this is a weak assumption, given the difficulty of the algorithm to converge when ρ is one, we conduct this test to check the robustness of the previous result, leaving to future research the check of these assumptions. Also in this case we do not need all the variables in the information set, but we can simply use the prediction as dependent variable. ⁶

5.6 Comments and Critical Points

The results show no optimal forecast of the bus service agency under different classes of loss functions. Furthermore, it is clear that the forecast tends to anticipate rather than postpone predicted bus arrival compared to actual bus arrival.

The critical points are the following: with a ρ on the boundaries the minimization problem becomes much more difficult, given a non differentiable objective function and the algorithm implemented in gmm does not perform well. In fact the in-sample J-statistic is surprisingly high. Furthermore, if any of the assumptions underlined does not hold, the conclusions are biased and , in this sense, further investigation seems necessary.

⁶See Appendix for a proof of moment condition 3.

Chapter 6

The Empirical Analysis

In this section I will describe the steps for evaluating different predictive models and for assessing their performance. Furthermore, I will try to understand what are the covariates most correlated with travel time. In total I analyze 20685 observations with different travel time. In figure 4.3 I reported a description of the distribution of travel time. For a complete description of the way the data set has been built see chapter 4.

6.1 Descriptive Statistics

The data frame contains 20685 observations and 21 continuous , two-levels or multi-levels factor variables. Vehicle id contains approximately 1320 levels, trip id 62, stop id 26, cartels 8, traffic 6 (from 0 to 4 with decreasing traffic, -1 information not available), weather 3, prevision of the weather 3, day of the week 7, day 20. The most important dummies and continuous variables are described in the table 6.1.

6.1.1 Variable description

Air correspond to a dummy equal to one if there is an air conditioning system on the vehicle, 0 otherwise. It might be a proxy for new or old vehicles.

Moby and Meb signals if on the vehicle tickets are buyable on board.

Footboard is a dummy equal to one if there is a footboard at the stop, zero otherwise. It can be a proxy for lanes for buses next to that stop (footboard are usually in the middle of the street, where there are preferential lanes).

Waiting time in minutes and in seconds are continuous variables reporting the ex-ante prediction of Atac.

Missing stops is the number of stops that vehicle must go through before reaching the

stop with given id.

At endstop is a dummy variable, equal to one if the bus is at the last stop of the route, zero otherwise.

Travel time is the actual travel time of the bus.

Direction is equal to one if it is outbound, zero if the bus goes inbound. Direction has been computed by merging the data with static data described in chapter 4.

The variables not reported in the table are vehicle id; trip id, which is the route in a given direction; stop id, reporting the id of the stop where the bus is arrived after n seconds of traveltime; traffic.x , for the traffic at end stop at the moment of the query; traffic.y for the traffic at the initial position of the vehicle; weather, variable to control for sunny, cloudy or raining weather; prevision, day-before prevision of the weather; day of the week, with all seven days; banda, three level factors reporting the velocity of the signal; day of the year.

Statistic	N	Mean	St. Dev.	Min	Max
footboard	20,685	0.088	0.283	0	1
waiting_time.min	20,685	18.716	10.428	3	40
missing_stops	20,685	16.131	9.731	1	63
at_endstop	20,685	0.352	0.478	0	1
waiting_time.seconds	20,685	1,122.273	625.939	150	2,429
traveltime	20,685	1,233.151	704.919	25	3,196
direction	20,685	0.453	0.498	0	1

TABLE 6.1: Descriptive Statistics

6.2 Univariate linear regression

In table 6.2 there is a summary of a regression with dependent variable travel time and independent variable the ex-ante prediction of Atac. The result are close to what we might expect. β_1 is close to one, that is , for one minute of expected travel time corresponds approximately one minute of actual travel time on average. Note that β_1 is greater then one, which might mean that the prediction of Atac on average underestimates real travel time. Both the betas are significantly different from zero. β_0 might have several interpretations: for zero seconds of expected travel time, there is approximately one minute of actual travel time. This might be interpreted as structural delay. On the other hand, if we adjust for slack time at the stop it might be not significant anymore. The adjusted R squared in the regression is particularly high, 87 per cent (much in sample variance captured by this linear model). In table 6.3 I showed the mean squared prediction error of the model of Atac and the MSPE of this simple linear

model. The standard deviation of the MSPE (out-of-sample) of the linear model is computed by bootstrapping 200 times the 5-folds cross validate mean squared error. The standard deviation of the 100 MSPE is 1347 and the five per cent confidence interval is (61182, 66464).

	<i>Dependent variable:</i>
	traveltime
waiting_time_sec	1.051*** (0.003)
Constant	53.257*** (3.602)
Observations	20,716
R ²	0.872
Adjusted R ²	0.872
Residual Std. Error	252.600 (df = 20714)
F Statistic	140,636.300*** (df = 1; 20714)
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01

TABLE 6.2: Univariate linear model

	mse Atac	MSPE Linear Model
mse	77,130.370	63,811.490

TABLE 6.3: Performance of Atac and Univariate linear model

6.3 A smoother function

A second approach is to use local regression, and in particular natural cubic splines. As explained in chapter 2, local regression suffers for curse of dimensionality problems and for this reason it is better to run this regression on few variables. Furthermore, it requires careful considerations if factors are involved in the regression in order to understand what measure of distance should be used. For these reasons I decided to run this model on only one variable, predicted waiting time in seconds. In the next sections I will explain why I believe that this variable can be considered one of the most important variables for prediction.

In figure 5.1 I show the behaviour of the MSPE as a function of the degrees of freedom of natural cubic splines. The best natural cubic spline has 10 degrees of freedom.

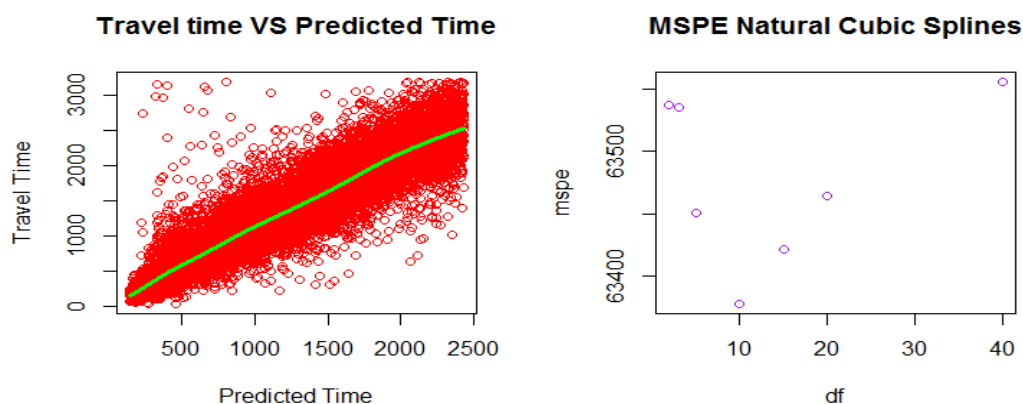


FIGURE 6.1: Spline, EDF VS MSPE

A similar result is for a smoothing spline selected using LOOCV (10.4 DF). The MSPE is 63377. Also in this case I use the technique of bootstrapping to get the SE. Note that the SE is only an approximation because we run the spline with a chosen lambda. The standard error is 1315 and the five percent confidence interval is (60798, 65955). The MSPE is lower than the MSPE of the linear regression but the confidence intervals overlap.

6.4 Multivariate regression with Lasso and Ridge Regression

For multivariate regression I dropped some variables depending on the circumstances. The most complete dataframe that I use has three variables dropped: vehicle id, cartels and day of the year. Whereas cartels had too many missing values, vehicle id had too many levels and it was computationally infeasible to use. The behaviour of the MSPE for lasso as a function of lambda is showed in figure 6.2. The second figure reports the number of variables included in the regression in function of lambda. The MSPE is computed with 5 folds cross validation. The process is the following: for each four out of the five folds, lambda is selected using 5-folds CV on this subset of data, tested on the remaining fold and the process is repeated five times. The SE is computed by bootstrapping 100 times.

To run the regression the data frame has first been converted into a matrix and all factors have been expanded to 0-1 dummies. The matrix has 193 variables and 20685 observations.

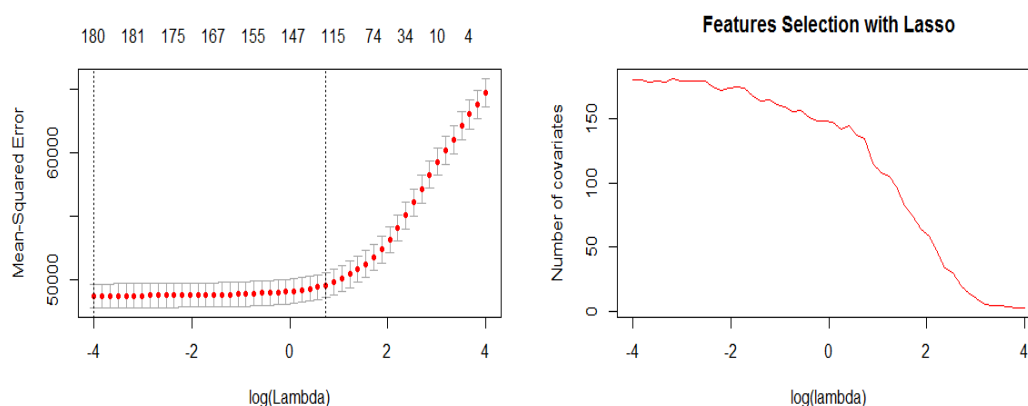


FIGURE 6.2: MSPE and Number of Covariate - Lasso 193 variables

The MSPE of lasso and ridge regression is particularly close each other. The MSPE for lasso is 48292 with SE 1139 and 48282 for ridge regression with SE 1143. The confidence intervals are respectively (46057,50526) and (46041,50524). The number of covariates in the regression for lasso are 173 and for ridge regression 193.

In a second moment I run a regression reducing the number of features to 51. All the dummies for each level of trip id, route id and stop id were dropped. The objective is to understand the cost of dropping this information against the computational gain. In fact, a regression on trip, route and stop id cannot be extended to all buses, route and trips in Rome unless it is carried out a study on each single stop in Rome. The 5-folds cross validated MSPE for lasso is 54977 with SE 1227 and five percent confidence interval (52571,57382). For ridge regression is MSPE 54964, SE 1223 , confidence interval (52566,57363). The MSPE is significantly higher then the one obtained by running a regression on more covariates by two standard deviations. On the other hand, it is still significantly lower then the prediction error of Atac. The lambda for the ridge regression is 0.01, while it is 1.77 for lasso. The number of variables used are 51 for ridge and 39 for lasso.

6.5 Stepwise

With stepwise we obtain similar results, but much more computational intensive. With 5-folds CV on a stepwise on the dataset with 193 covariates the MSPE is 48736 and the SE obtained by bootstrapping 100 times is 1378. The five percent confidence interval is (46035,51437). The MSPE of a stepwise on only 51 covariates is 54817 and the SE is 1322. The confidence interval is (52225,57410). The outcome of a stepwise on 51 variables is a regression function with 34 variables. Here a brief overview of the regression

showed in the appendix:¹: some of them are most of the hours of the day (dummy for each hour), waiting time in seconds and minutes (significant with both positive sign), traffic, rain (significant with positive sign), some days of the week, direction (significant with positive sign), “at end stop” (significant with positive sign), number of stops to go through (significant with positive sign), prevision of cloudy weather (significant with positive sign) and others. Whereas it is difficult to give an interpretation to most of the dummies because the effect must be compared to the average seconds of travel time of all the levels not included in the regression, interesting results are for “at end stop” and “missing stops”, both significant with positive sign. In the next section I will show further results carried out on these variables.

6.6 Pruned tree and Random Forest

Pruned tree is showed in figure 6.3. The only variable used in this case is the ex-ante prediction. For data set with less variance of travel time the results for trees are different as showed in figure 2.2. The MSPE is computed with 5-folds CV on a pruned tree trained on 4 folds. The result is a MSPE of 70608 with SE 1674. The five per cent confidence is (67326,73889). As we might expect the performance of a single pruned tree is much lower compared to linear regression. Running a Random Forest on only 51 covariates (dropping vehicle, stop and trip id) we get an out of bag mean squared prediction error around 4700 with around 400 trees. The SE is 497 and the confidence interval is (46006,47954). To check the result, we run a random forest on a training set with 4/5 observations and test on the remaining out-of sample observations getting a similar result. Note that the error has a positive bias compared to the cross-validated mean squared prediction error. Running a random forest with 193 variables we get an OOB MSPE around 42000 with 490 trees.

6.6.1 Variables Importance

Importance is a measure of how much a certain variable on average increases the purity of the nodes in a random Forest. Purity for linear regression is computed by averaging the MSE between the two nodes. The result on 193 covariates is showed in figure 6.4. In the case of 51 variables we obtain similar results. These variables have been used for three different regression with the first 4, 7 and 10 most important covariates on travel time. For (1) in table 6.5 waiting time in seconds is significantly correlated with actual travel time with a beta close to one (one-to-one relationship). On average one more

¹Note: the error in brackets are always the heteroskedastic robust standard errors.

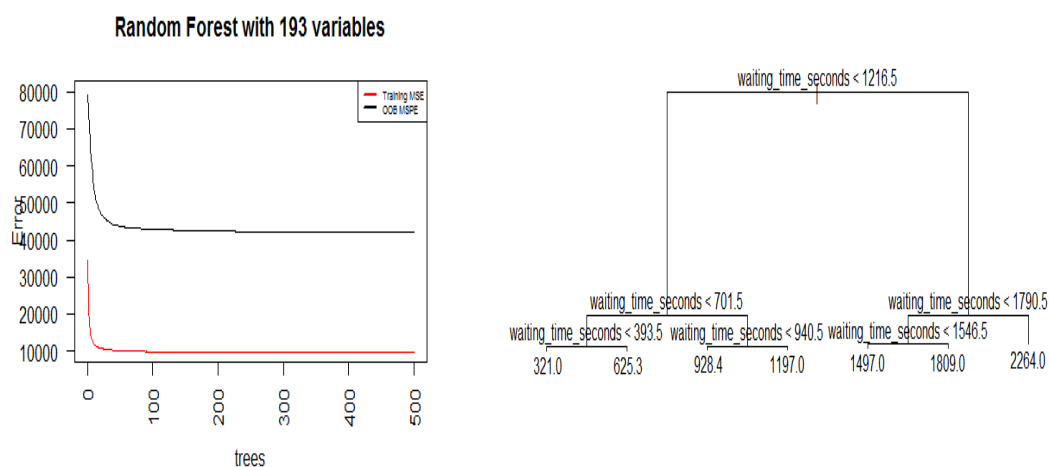


FIGURE 6.3: MSPE of Random Forests and pruned tree

missing stop increases travel time by ten seconds, which makes sense if we consider the few seconds of slack time at the stop. At end stop shows that unscheduled slack time at end stop increases on average travel time by one minute and an half. Finally the constant shows that waiting time at end stop has a minimum value of 20 seconds for a bus with zero predicted waiting seconds, no stops to go through and not at the end stop. Waiting time in minutes is not significant maybe because much of its information is already captured by waiting time in seconds. The understanding of the dummy at end stop may be related to higher slack time then scheduled. On the other hand, it may be positively correlated to travel time only because the ex-ante prediction is not adjusted for slack time. (2) and (3) shows similar results adding more variables. In particular, 4 pm seems to increase travel time by half minute on average, maybe due to higher traffic conditions. Low traffic is negatively correlated with travel time, as we might expect. The only dummy of traffic with positive sign is traffic.y 3 (level 3 of traffic at starting point) which shows contradictory results. One reason may be that the function collect this variable only in an approximate way, introducing bias in the collection of data. A second reason may be that this information does not truly represents traffic conditions. Finally any variables correlated with both the independent variable and the error term may introduce bias. An example may be trip id. A certain trip may be correlated with the traffic and also with travel time.

6.7 Stop 70988: A case study for future research

Why buses at stop 70988 take on average one minute and an half more of travel time ceteris paribus?

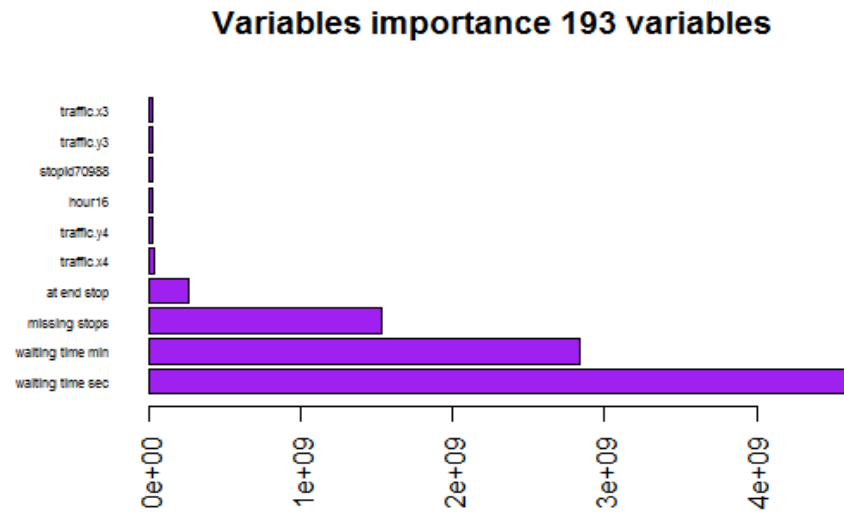


FIGURE 6.4: Variables importance in random forest

	<i>Dependent variable:</i>		
	traveltime		
	(1)	(2)	(3)
waiting_time_seconds	0.938*** (0.099)	0.920*** (0.0982)	0.929*** (0.097)
waiting_time.min	-2.261 (5.976)	-4.068 (5.872)	-4.064 (5.827)
missing_stops	10.617*** (0.371)	13.905*** (0.407)	13.682*** (0.403)
at_endstop	86.187*** (4.616)	103.714*** (4.677)	103.811*** (4.661)
traffic.x4		-100.030*** (4.697)	-239.475*** (27.946)
traffic.y4		-30.221*** (4.287)	64.159*** (12.484)
hour4pm		25.245*** (9.231)	24.845*** (9.199)
id_stop70988			81.386*** (7.469)
traffic.y3			101.821*** (12.476)
traffic.x3			-139.073*** (27.951)
Constant	21.343*** (3.610)	112.704*** (4.948)	146.516*** (28.652)
Observations	20,685	20,685	20,685
R ²	0.879	0.883	0.885
Adjusted R ²	0.879	0.883	0.885
Residual Std. Error	244.968 (df = 20680)	240.818 (df = 20677)	239.238 (df = 20674)
F Statistic	37,648.860*** (df = 4; 20680)	22,364.690*** (df = 7; 20677)	15,890.470*** (df = 10; 20674)

Note: *p<0.1; **p<0.05; ***p<0.01

TABLE 6.4: Linear Regressions - feature selection with RF



FIGURE 6.5: Stop70988

One possible reason is that the bus stop is just before the entrance of an hotel and a religious institute. This means that cars that are entering or exiting from the building may increase slack time of the vehicle at the bus stop. A second possible reason is that on the same street of the stop there are other offices and public buildings (mainly religious ones). In particular few metres before the stop there is the entrance of a public building again on the same side of the street of the stop. In the picture just some are showed and there are others on the same street. Finally, the street has just one line for each side and this may increase traffic during the day.

As just shown, this kind of prediction can be useful to detect also misplaced stops. Probably this stop could be moved few metres onwards or even it could be put in a parallel street to decrease the delay effect on travel time. On the other hand there are many critical points in this analysis. Whereas we control for many other variables, there could be omitted variable bias and further research seems necessary to check the robustness of this result.

	Loss Atac	Loss GMM
mean loss	35800	20988

TABLE 6.5: Test-set error of Atac and GMM under asymmetric loss

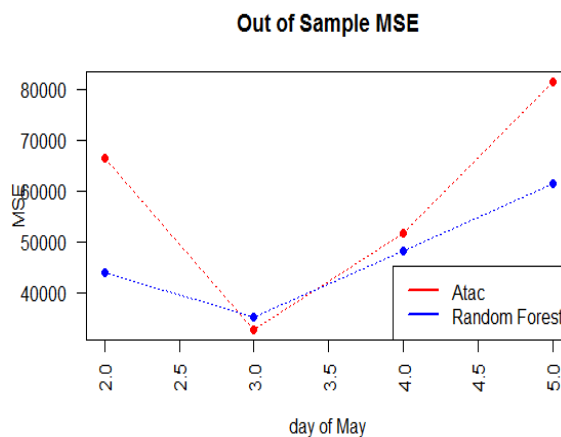


FIGURE 6.6: Out of sample MSE on 2nd,3rd,4th,5th of May

6.8 Test out of sample using Random Forest and MSE

In this section I describe the results of a test conducted out of sample. To run this test it was constructed a random forest on 193 variables using a dataset with data from the 16th of April to the 30th of April. The out of sample data were between the 2nd of May and the 5th of May. The mean squared error is 47727, while the mse of Atac is 65648, and it is slightly higher the the out of bag mean squared error which is around 42000. This result is particularly interesting considering the objective of this study. In fact, it shows that models built on past data can still significantly outperform the prediction of Atac. On the other hand, figure 6.6 shows that for one day out of four this is not true. As we might expect, the gain of a different model out of sample is still significant but lower compared to the results obtained using cross validation. Future research should collect more data to check this result on a larger time gap.

	MSE Atac	MSE Random Forest
mean loss	65648	47727

TABLE 6.6: Performance out of sample of Atac and Random Forest under MSE

6.9 Prediction with a different loss

In this section I develop an alternative linear model using an asymmetric loss. I recall the general definition of a two-parameters loss function defined in chapter 5:

$$\Lambda_i(\rho, \alpha, \theta) = [\alpha + (1 - 2\alpha)I(y_i - \hat{f}(x_i) < 0)]|y_i - \hat{f}(x_i)|^\rho \quad (6.1)$$

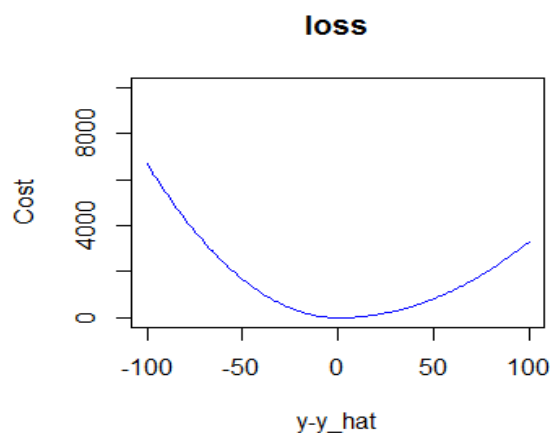


FIGURE 6.7: Loss function used for our prediction

Using GMM the estimation of α under three moment conditions was $\alpha = 0.33$. This means that I weight more negative errors than positive ones. Furthermore I set ρ equal to 2. The reason is simply to get a convex and differentiable functions, although a better value for ρ would be one as showed in the previous chapter. The argument θ that minimizes the loss is found using iterative GMM on 133 variables (they were 193 but to rule out multicollinearity the number of columns of the covariate matrix was reduced to 133).² The moment conditions are:

$$E[h(\gamma, W)] = E[X'(I(y - \hat{f}(x) < 0) - \alpha_0)]|y_i - \hat{f}(x)|^{\rho-1}] = 0 \quad (6.2)$$

The model is correctly identified (number of moments - that is the number of the column of X - equal to the number of parameters). The results show that lines may be positive or negative correlated to traveltime; some stops are negatively correlated with travel time and further investigation seems necessary. Other variables such as the dummy at end stop and the number of missing stops are positively correlated. To study the predictive power of the model we train it on 4/5th of observations and use a test set with the remaining 1/5th. Whereas the error of Atac is 35800, the error of GMM is 20988.³ Also in this case the performance of our model is better than the prediction of Atac.

²See chapter 2 for a comprehensive description of GMM.

³Note that these values cannot be compared to the values of the MSE of the previous sections.

Chapter 7

Conclusions

In this chapter I will briefly describe the core findings of this study, I will underline the critical points and describe the direction that future research may take.

7.1 Achievements

The results show an higher predictive power in terms of out of sample prediction error of all the models that have been constructed compared to the one in use today. In figure 7.1 there is a summary of the MSPE for each model used in this research. The figure reports the cross validated mean squared error of each model. Whereas the MSPE has been computed using 5-folds cross validation for all the models, the only exception is for random forest, for which it was used the out-of-bag mean squared error. Note that the results have a positive bias due to the fact that only 4/5th observations of the dataset has been used for constructing the models. Moreover, empirical evidence shows that this bias tend to be higher for the OOB MSE compared to the CV MSE. As already explained in chapter 2, the 95 per cent confidence interval showed in figure 7.1 is constructed bootstrapping 100 times. Random forest has the highest prediction power; the lowest prediction power is for pruned tree and univariate regression. Even using just few variables, without including stop id, trip id, route and vehicle id, lasso, ridge regression, stepwise and random forest are significantly better then the predictor in use.

The result obtained by running a random forest on data between the 16th and the 30th of April and tested on the 2nd, 3rd, 4th and 5th of May still shows a averaged better performance, but the gain is lower compared to the gain estimated with cross validation.

A second important achievement regards the analysis of the variables. As already shown

in the previous chapter, using the variables' importance reported by the random forest, it is possible to detect particular stops negatively correlated with travel time. This negative correlation can be due to wrong allocation of stops if all the assumptions hold, as showed for the example of the stop 70988. A similar analysis may be done for detecting vehicles with particular delays. In this sense future research should test whether this statement may be correct.

Other variables correlated with travel time are the number of stops before arrival and the dummy at end stop, proxy for slack time of the driver at the end stop. A possibility of this positive correlation is that the prediction of Atac underestimates slack time of the driver. On the other hand, we should be careful with this statement because if the bus arrives at the last stop with delay, then this correlation may be due to this delay and not to higher slack time.

In the last part of the empirical analysis I built a predictor by minimizing a quadratic asymmetric loss function. The asymmetry gave more weight to errors whose prediction overestimate travel time. The reason was because an higher prediction may increase the likelihood that the person goes to the stop once the bus has already gone. I preferred a power of the error ρ equal to 2 simply because it leads to a differentiable loss function. GMM was used to estimate the function that also in this case showed an higher out of sample predictive power compared to Atac. Further research should develop alternative predictors under this new loss.

Chapter 5 was completely devoted to the estimation of the parameters of a generic two-parameters loss function, the asymmetric parameter α and the power of the error ρ under the weak assumption of forecast optimality of Atac. GMM regression with a two-parameter generic loss function was used to estimate these values. The result show an higher weight to negative errors as we might expect and a ρ equal to one. The J-test rejects the null hypothesis of optimality of the forecast of Atac under a generic class of loss functions, by assuming correct specification of the three moments conditions. On the hand the high value of the in-sample J-stat show a problem of convergence of the algorithm probably related to the non differentiability of the loss at ρ equal to 1. To adjust for this problem the power of the error term was set equal to 1.1 but the J-stat shows again similar problems and further investigation seems necessary. An indicator test - introduced for the first time by Patton[31] - and a linear regression test were implemented leading to the rejection of optimality forecast under specific assumptions.

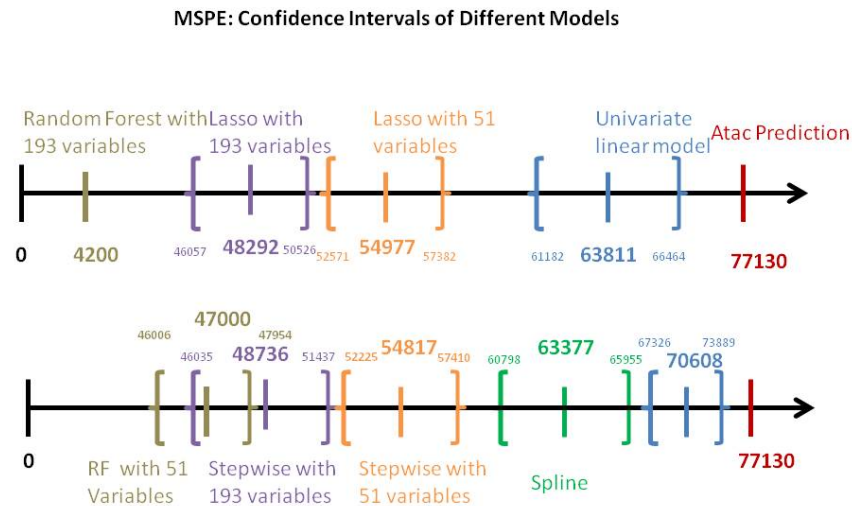


FIGURE 7.1: Summary of Model Performance

7.2 Future Research

Future research should be focused on collecting more data and testing new models. One possible track to follow is to improve the code developed during this research to construct an application able to make real time predictions. Extending the period of data collection is a necessary condition to build a strong predictor. Furthermore, many other sources of real time data can be used to improve the quality of the predictor, such as real time API for weather. Finally, these models may be used for other fields: identification of wrongly collocated bus stops or particularly slow vehicles may be detected in real time using these models. In fact, detection of broken buses can be a new field of research. Id of vehicles may become an interesting variable to control for anomalous behaviour. Abnormal delays may signal technical problems of buses and a certain classifier may be able to detect buses with high likelihood of becoming broken.

7.3 Critical Points

Key questions are: Are the observations on 26 stops representative of all the population? Are the dropped observations key elements that would arise bias in my computation? Are two weeks representative for the whole year? What are the costs in implementing these models to make predictions?

Not all these questions have an easy answer and many others should be added. The first problem come from the collection of data. The way the data set has been built may arise many bias. Not all buses are detected, the data set is conditional on the information provided by Atac. A further problem may arise when inactive buses are detected on their route if the system is not updated in time. In addition, dropped observations may play a crucial role in bus prediction. In fact, as already explained, I dropped almost the 0.5 per cent top percentile to avoid mismatched observations of same vehicles on the same route in different time windows. In addition, two weeks are not representative for all the year and more data should be collected. Variables such as day or hour may play a crucial role if more variance is present for these covariates. Moreover, to work in a proper way the model should use data as close as possible to real time data. Good predictions over time may require to continuously update the model to minimize the out of sample error. In this sense, implementation of different models may require huge costs. This is one reason why in the previous section I tried to decrease the number of covariates used in the regression(from 193 to 51). On the other hand, to work in a proper way , continuous updating of the coefficients using feasible samples of observations may be useful. New research should be focused on collecting new samples of data and to test models during different weeks.

A further problem relies on the reliability of the information provided by Atac. This can be a huge limit if this information is not correctly updated in real time and further investigation sem necessary to explore alternative sources of information.

In addition, there are many limits on the ways variables have been computed. Traffic time is represented by the average speed of buses close to a certain stop, it only covers the starting point and the end point of the trip of the bus and it does not show high variance. Moreover the collection of this variable is conditional on several assumptions. Weather and predicted weather are computed manually and have low variance. Some dummies have been dropped because always equal to the same value. Other factors such as id of the vehicle have too many levels. Cartels have too many missing values and have not been used at all. On the other hand missing values on other variables were very few, around 30 or 40 out of thousands of observations and they were dropped. If missing values were correlated with the output variables this would create bias on the data set.

The data set is not continuous in time because of problems of connection to the server of Atac. This can create further bias if bad connection is correlated with travel time (although it might be unreasonable). Also fatal errors of the machine have interrupted for some hours the collection of data.

7.4 Final Comments

Whether we can exploit the huge amount of data to improve the micro-cosm were we live remains an open question. Whereas data are available, the possibility to store, read and analyze this data can be extremely difficult. A more complex task is then the understanding of this data, which most of the time is an art rather than a science. With this thesis I tried to give my small contribution showing how real time data of bus arrival, apparently useless, can become useful for the people of a city with a bit of efforts. Whereas the choice of the best model may depend on the nature of the problem - see the no-free-lunch Machine Learning theorem - the methodology could be extended also to other area of research - hopefully.

Appendix A

Miscellaneous

A.1 Proof EDF

To prove:

$$\text{trace}(S) = \sum_1^n \text{cov}(\hat{y}_i, y_i) / \sigma^2 \quad (\text{A.1})$$

Assumptions:

- $\hat{y}_i = \sum_{j=1}^n S_{i,j} y_j$
- $n \geq j \geq i \geq 1$
- $(S_{i,j})$ does not depend on Y
- y_1, \dots, y_n are uncorrelated and $\text{var}(y_i) = \sigma^2$

From the assumption we can derive that $\text{cov}(y_i, y_j) = 0 \forall i \neq j$, $\text{cov}(y_i, y_j) = \sigma^2 \forall i = j$.
 $\text{cov}(\hat{y}_i, y_i) = \text{cov}(\sum_{j=1}^n S_{i,j} y_j, y_i) = S_{i,i} \sigma^2 + 2S_{i,j} \text{cov}(y_j, y_i) \forall j \neq i$. By the assumption of iid:

$$\text{cov}(\hat{y}_i, y_i) = S_{ii} \sigma^2 \quad (\text{A.2})$$

$$\text{EDF} = \frac{1}{\sigma^2} \sum_1^n \text{cov}(\hat{y}_i, y_i) = \sum_i^n S_{ii} = \text{trace}(S) \quad (\text{A.3})$$

A.2 Proof Moment Condition

A.2.1 Moment Condition 1

To prove: $E[1 * \lambda(\hat{f}, y, \gamma_0)] = 0$.

Where $\gamma_0 = \underset{\gamma}{\operatorname{argmin}} E[\Lambda(\hat{f}, y, \gamma)]$, then by the first order condition, $\frac{\partial \Lambda(\gamma)}{\partial \gamma} = 0$

$\rightarrow C * \lambda(\hat{f}, y, \gamma) = 0$ where $\lambda(\hat{f}, y, \gamma) = \frac{\partial \Lambda(\gamma)}{\partial \gamma} = 0$ at any local minimum and C is a constant.

A.2.2 Moment Condition 2

To prove: $E[\lambda(\hat{f}, y, \gamma_0)' \Lambda(\hat{f}, y, \gamma_0)] = 0$.

This moment condition is derived because the derivative of a fixed length vector is always orthogonal to the vector itself. Given a vector of fixed length V, then: $|V|^2$ is constant, $\sum v_a^2$ is constant, $\sum \frac{\partial}{\partial t} v_a^2 = 0$, by the chain rule $\sum 2 \frac{\partial v_a}{\partial t} v_a = 0$, $\sum \frac{\partial v_a}{\partial t} v_a = 0$, $V \frac{\partial V}{\partial t} = 0$.

A.2.3 Moment Condition 3

To prove: $E[\hat{y}_i \lambda(\hat{f}, y, \gamma_0)] = 0$.

Under the two parameter class of loss functions, assuming:

$$\hat{f} = X\gamma_0$$

$$\gamma_0 \neq 0$$

$$\lambda(\hat{f}, y, \gamma) = E[\rho X'(I(y - \hat{f}(x) < 0) - \alpha)] |y_i - \hat{f}(x)|^{\rho-1}] = 0 \quad (\text{A.4})$$

Because ρ is a constant this can be written as

$$\lambda(\hat{f}, y, \gamma) = E[X'(I(y - \hat{f}(x) < 0) - \alpha)] |y_i - \hat{f}(x)|^{\rho-1}] = 0 \quad (\text{A.5})$$

By multiplying each side for a fixed valued vector γ'_0 , $\gamma'_0 E[X'(I(y - \hat{f}(x) < 0) - \alpha)] |y_i - \hat{f}(x)|^{\rho-1}] = E[(X\gamma_0)'(I(y - \hat{f}(x) < 0) - \alpha)] |y_i - \hat{f}(x)|^{\rho-1}] = E[\hat{f}'(I(y - \hat{f}(x) < 0) - \alpha)] |y_i - \hat{f}(x)|^{\rho-1}] = 0$

A.3 Stepwise: results

For all other tables see chapter 6.

TABLE A.1: Features selection with Stepwise

	<i>Dependent variable:</i>	
	traveltime	
waiting_time_seconds	0.845***	(0.005)
missing_stops	14.594***	(0.335)
traffic.x4	-203.502***	(21.184)
at_endstop	119.286***	(4.029)
traffic.y1	-316.939***	(22.911)
air	67.208***	(6.128)
traffico.y3	32.801***	(4.048)
hour4pm	82.330***	(7.695)
hour7am	70.803***	(8.236)
hour3am	-156.306***	(18.211)
hour1am	-143.189***	(17.437)
hour2am	-128.249***	(20.689)
hour5am	-59.465***	(10.533)
traffic.x3	-123.275***	(21.212)
Wednesday	-19.757***	(4.679)
hour7pm	57.041***	(8.099)
direction	21.390***	(3.501)
prevision_w1	12.831***	(4.000)
hour6pm	45.145***	(7.327)
hour5pm	41.145***	(6.862)
hour4am	-75.453***	(19.462)
hour3pm	36.803***	(8.576)
hour11am	30.216***	(8.176)
banda.x1	-13.210***	(4.445)
Thursday	-14.272***	(5.361)
hour11pm	-21.011**	(8.861)
hour9pm	-12.723*	(7.613)
traffico.y2	-23.469**	(10.188)
hour10am	19.758***	(7.595)
hour12am	20.472**	(8.604)
Rain	10.159*	(5.697)
hour2pm	15.422	(9.583)
hour8am	17.949	(10.916)
Saturday	8.153	(5.742)
Constant	155.399***	(21.830)
Observations	20,685	
R ²	0.889	
Adjusted R ²	0.889	
Residual Std. Error	234.789 (df = 20650)	
F Statistic	4,876.395*** (df = 34; 20650)	
<i>Note:</i>	* p<0.1; ** p<0.05; *** p<0.01	

A.4 Code

Code for Cross validation (in parallel execution with bootstrapping) is in chapter 2, section 2.2.8; code for queries is in chapter 4, section 4.1. All scripts are available under request.

A.4.1 Code for elaboration of data

The code has the following structure:

During the loop we first read each text file saved during the queries and we store only the observations that are close to the bus stop:

```

| Load static Data
| Create null object:  xx = NULL
| Initialize accumulator:  k = 1
| while try(read(data from query k)) is not error
|   | xx = append by row xx to all buses with 0 minutes of expected arrival
|   |   at bus stop
|   | k = k +1

```

To clean this matrix we drop doubled observations for same vehicle id, same stop id, same day, hour and month:

```

| Create two null objects:  data = notdata = NULL
| for i in 1:  nrow(xx)
|   | cc = xx[i,]
|   | if cc is neither in notdata nor in data
|   |   | obs = matrix with all observations in xx with same vehicle id, trip
|   |   |   id, stop, month, day and hour of cc
|   |   |   dd = vector of time values of each observation in obs
|   |   |   pos = the observation in obs with minimum (as.numeric(time))
|   |   |   obs = all observations in obs minus pos
|   |   |   data = append by row pos to data
|   |   |   notdata = append by row obs to notdata

```

To create the matrix of buses that have still to arrive to the stop we do something similar, but now the distance in minutes is not 0 but C , were C is uniformly distributed between 3 and 39:

```

| Load static Data
| Create null object:  xx2 = NULL
| Initialize accumulator:  k = 1
| while try(read(data from query k)) is not error
|   | c = pick a number uniformly distributed between 3 and 39
|   | xx2 = append by row xx2 to all buses with minutes of expected arrival
|   |   at bus stop equal to c
|   | k = k +1

```

To clean also this matrix, we drop again doubled observations:

```

| Create two null object:  data2 = notdata2 = NULL
| for i in 1:  nrow(xx2)
|   | cc = xx2[i,]
|   | if cc is neither in notdata2 nor in data2

```

```

| obs = matrix with all observations in xx2 with same vehicle id, trip
|   id, stop, month, day and hour of cc
| dd = vector of time values of each observation in obs
| pos = the observation in obs with minimum (as.numeric(time)) (you
|   could pick any observation, not necessarily the min)
| obs = all observations in obs minus pos
| data2 = append by row pos to data2
| notdata2 = append by row obs to notdata2

```

Merge the arrival and the bus with distance C expected minutes from the stop and clean for same observations with same time of arrival, same stop id, same month, day, hour, same trip and same vehicle id (mismatched observations):

```

| rr = merge data with data2 by the id of the vehicle, the id of the trip
|   and the id of the stop
|
| gooddata = all rr where as.numeric(time of arrival) > as.numeric(time of
|   detection)
| Create null vectors: bad = good = NULL
| for i in 1: nrow(gooddata)
|   | cc = gooddata[i,]
|   | if cc is neither in bad nor in good
|   |   | obs = matrix with all observations in gooddata with same vehicle id,
|   |   |   trip id, stop, month, day and hour of ARRIVAL of cc
|   |   | pos = observation in obs with maximum as.numeric(time)
|   |   | obs = obs minus pos
|   |   | good = append pos to good
|   |   | bad = append obs to bad

```

Clean for same observations with same time of detection, same stop id, same month, day, hour, same trip and same vehicle id:

```

| null vectors bad2 = good2 = NULL
| for i in 1: nrow(good2)
|   | cc = good[i,]
|   | if cc is neither in bad2 nor in good2
|   |   | obs = matrix with all observations in gooddata with same vehicle id,
|   |   |   trip id, stop, month, day and hour of STARTING POINT of cc
|   |   | pos = observation in obs with minimum as.numeric(time)
|   |   | obs = obs minus pos
|   |   | good2 = append pos to good2
|   |   | bad2 = append obs to bad2

```

```

├─ Add day, weather, direction to good2
├─ compute travel time as the difference in seconds between time of arrival
  and time of detection
└─ save good2

```

A.4.2 Traffic Function

```

Function(trip id)
├─ Query Trip s3
├─ initialize first accumulator: set y = 1
├─ while y is less than 100
│   ├─ if y==1
│   │   ├─ matrix2 = 1st values of s3
│   │   ├─ if (stop of matrix2 == STOP AND stop distance of xx < 2)
│   │   │   ├─ traffic condition = traffic condition of matrix2
│   │   │   └─ exit the loop: y = 101
│   │   └─ if (yth stop == STOP AND 2*nrows(matrix2) +2 >= stop distance)
│   │       ├─ traffic state = TS in row abs(nrow(matrix2) - stop distance)
│   │       └─ exit the loop: y = 101
│   │   └─ if (STOP in matrix2 AND 2*untilstop + newarrivals >= stop distance)
│   │       ├─ traffic state = TS in last row matrix2
│   │       └─ exit the loop: y = 101
│   └─ else
│       ├─ bind by rows matrix2 with yth values of s3
│       ├─ if yth stop of matrix2 == STOP
│       │   ├─ initialize second accumulator: untilstop = y
│       │   ├─ initialize third accumulator: newarrivals = 0
│       │   └─ y = y + 1
│       └─ else
│           ├─ newarrivals = newarrivals + 1
│           └─ y = y + 1
└─ return traffic state

```

Traffic values are between 0 and 4 in decreasing order of traffic and -1 if they are not available. They depend on the average speed of buses within a given radius from the stop. The input of the query from traffic values must be the id of the trip. The output is a list of dictionaries and values, reporting each stops in the trip and the traffic condition for each stop. The objective of this function is to make a query on the id of the trip for each bus arrival, find the closest stop to the vehicle in that moment and get the traffic condition at that stop. The function builds a matrix with one single stop id

and the corresponding traffic condition in each row. The rows are ordered from the first stop to the last. To save time, the function does not build the full matrix but it stops when it finds the right stop. For seek of brevity the graph reports many abbreviations. STOP is the id of the stop; y, untilstop and newarrivals are three context-preserving accumulators. The first, y, has the same role of x in the previous loop. It tells what is the stop to extrapolate from the list and to append to the matrix. Untilstop tells what is the position in the matrix of S_i - the stop queried in the previous loop. Newarrivals is the number of stops in the matrix after S_i . TS is traffic state. The function exploits an additional information from the values of the bus arrival: the number of missing stops to arrive to S_i , named in the graph stop distance.

I assumed that the vehicle do not change the route between two subsequent trips. One further assumption is that traffic condition reported for stops in the opposite side of a street - same stops of the same route on the trips with opposite directions - have the same traffic condition.

The first if statement checks whether the first stop on the trip corresponds to the closest stop to the vehicle considering both directions. If this is false, the second if statement checks whether S_i is the last element of the matrix and whether the vehicle has a stop distance less then the number of stops reported in the matrix. If this is true, we are sure that the stop of interest is in the matrix no matter what direction the bus has. The function picks the traffic position from the ith element of the matrix, where i is the number of rows minus the number of missing stops. It takes the absolute value because it may be a negative number if the vehicle is in the opposite direction. The second if statement checks whether both S_i and the stop of interest are in the matrix. If this is true it picks the last row of the matrix. In fact this condition is satisfied only when the stop of interest enters in the matrix after S_i , as the last observation.

A.5 Final notes on the models

- For linear regression the error was computed using the heteroskedastic formula;
- For splines both smoothing splines with loocv and cubic splines with 5 folds cv were tested;
- For shrinkage models covariates were standardized as explained in chapter 2.

Bibliography

- [1] Abkowitz, Mark D., and Israel Engelstein (1983): “Factors affecting running time on transit routes,” *Transportation Research Part A: General* 17.2, 107-113.
- [2] Albright, E., and Figliozzi, M. (2012): “Factors influencing effectiveness of transit signal priority and late-bus recovery at signalized-intersection level,” *Transportation Research Record: Journal of the Transportation Research Board*, (2311), 186-194.
- [3] Amita, J., Singh, J. S., Kumar, G. P. (2015): “Prediction of bus travel time using artificial neural network,” *International Journal for traffic and Transport Engineering*, 1(5), 410-424.
- [4] Bin, Y., Zhongzhen, Y., Baozhen, Y. (2006): “Bus arrival time prediction using support vector machines,” *Journal of Intelligent Transportation Systems* 10(4), 151-158.
- [5] Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., and Ratti, C. (2011): “Real-time urban monitoring using cell phones: A case study in Rome,” *Intelligent Transportation Systems, IEEE Transactions on* 12(1), 141-151.
- [6] Carrion, C., and Levinson, D. (2012): “Value of travel time reliability: A review of current evidence,” *Transportation research part A: policy and practice*, 46(4), 720-741.
- [7] Chang, J. S. (2010) : “Assessing travel time reliability in transport appraisal,” *Journal of Transport Geography* 18(3), 419-425.
- [8] Chang, C. Z., Chen, X. M., Wang, M. (2015, May): “Study on Combinational Scheduling Optimization of Bus Transit Rapid Based on Tabu Search Genetic Algorithm,” *Applied Mechanics and Materials* (Vol. 744, pp. 1827-1831).
- [9] Chen, G., Yang, X., Liu, H., Liu, X. (2013, October): “Regression-based approach for bus trajectory estimation,” *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (pp. 1876-1881). IEEE.
- [10] Pierre Chausse (2010): “Computing Generalized Method of Moments and Generalized Empirical Likelihood with R,” *Journal of Statistical Software* 34(11), 1-35.

- [11] Diab, E. I., and El-Geneidy, A. M. (2013): “Variation in bus transit service: understanding the impacts of various improvement strategies on transit service reliability,” *Public Transport*, 4(3), 209-231.
- [12] Duncan Temple Lang: XMLRPC: Remote Procedure Call (RPC) via XML in R.
- [13] Efron, B. (1986): “How biased is the apparent error rate of a prediction rule?,” *Journal of the American Statistical Association*, 81(394), 461-470.
- [14] El Geneidy, A. M., Horning, J., and Krizek, K. J. (2011): “Analyzing transit service reliability using detailed data from automatic vehicular locator systems,” *Journal of Advanced Transportation*, 45(1), 66-79.
- [15] Elliott, G., Timmermann, A., Komunjer, I. (2005): “Estimation and testing of forecast rationality under flexible loss.,” *The Review of Economic Studies*, 45(1), 72(4), 1107-1125.
- [16] Feng, W., Figliozzi, M., and Bertini, R. L. (2015): “Quantifying the joint impacts of stop locations, signalized intersections, and traffic conditions on bus travel time,” *Public Transport*, 7(3), 391-408.
- [17] Friedman, J., Hastie, T., and Tibshirani, R. (2010): “Regularization Paths for Generalized Linear Models via Coordinate Descent” *Journal of Statistical Software*, 33(1), 1-22.
- [18] Friedman, J., Hastie, T., and Tibshirani, R. (2001): “The elements of statistical learning.” Springer, Berlin: Springer series in statistics.
- [19] Gong, X., Guo, X., Dou, X., and Lu, L. (2015): “Bus Travel Time Deviation Analysis Using Automatic Vehicle Location Data and Structural Equation Modeling,” *Mathematical Problems in Engineering*, 2015.
- [20] Gurmu, Z. K., Fan, W. D. (2014): “Artificial neural network travel time prediction model for buses using only GPS data,” *Journal of Public Transportation*, 17(2), 3.
- [21] Hamilton, J. D. (1994): “Time series analysis,” *Princeton: Princeton university press*.
- [22] Henderson, G., and Darapanemi, V. (1994): “Managerial uses of causal models of subway on-time performance,” *Transportation Research Record*, (1451).
- [23] Janson, L., Fithian, W., Hastie, T. J. (2015): “Effective degrees of freedom: a flawed metaphor,” *Biometrika*, asv019..
- [24] A. Liaw and M. Wiener (2002): “M Classification and Regression by randomForest,” *R News*, 2(3), 18–22.

- [25] Lin, W. H., and Bertini, R. L. (2004): “Modeling schedule recovery processes in transit operations for bus arrival time prediction,” *Journal of Advanced Transportation*, 38(3), 347-365.
- [26] Lin, Y., Yang, X., Zou, N., and Jia, L. (2013): “Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China,” *Journal of Transportation Engineering*, 139, 139(11), 1133-1140.
- [27] Hlavac, Marek (2014): “stargazer: LaTeX/HTML code and ASCII text for well-formatted regression and summary statistics tables,” R package version 5.1
- [28] Mazloumi, E., Moridpour, S., Currie, G., Rose, G. (2011): “exploring the value of traffic flow data in bus travel time prediction,” *Journal of Transportation Engineering*, 138(4), 436-446.
- [29] Kalaputapu, R. and Demetsky M. (1995): “Modeling Schedule Deviation of Buses Using Automatic Vehicle-Location Data and Artificial Neural Networks,” *Transportation Research Record*, 1497, 4452.
- [30] Kimpel, Thomas J (2001): “time point-level analysis of transit service reliability and passenger demand”.
- [31] Patton, A. J., Timmermann, A. (2012): “Testing forecast optimality under unknown loss” *Journal of the American Statistical Association*.
- [32] Patton, A. J., Timmermann, A. (2007): “Properties of optimal forecasts under asymmetric loss and nonlinearity” *Journal of Econometrics*, 140(2), 884-918.
- [33] Pinelli, F., Hou, A., Calabrese, F., Nanni, M., Zegras, C., and Ratti, C. (2009, October): “pace and time-dependant bus accessibility: a case study in Rome,” *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on* (pp. 1-6). IEEE.
- [34] R Core Team (2013): “R: A language and environment for statistical computing. R Foundation for Statistical Computing”, Vienna, Austria
- [35] Ragusa ,G(2016). “Econometric Theory, Notes”
- [36] Giuseppe Ragusa (2014): “ase: Collection of useful functions for Applied Statistics and Econometrics”, R package version 1.0
- [37] Revolution Analytics and Steve Weston (2014): “doSNOW: Foreach parallel adaptor for the snow package.”
- [38] Revolution Analytics and Steve Weston (2014): “foreach: Foreach looping construct for R.”

- [39] Brian Ripley (2014): “tree: Classification and regression trees” *R package version 1.0-35*
- [40] Rinott, Y. (2016): “Linear Model, Model Selection, Notes”
- [41] Ryus, P. (2003): “A Summary of TCRP Report 88: A Guidebook for Developing a Transit Performance Measurement System,” *TCRP Research Results Digest* (56).
- [42] Salibian-Barrera, M. (2016): “Methods for Statistical Learning, Notes”
- [43] Schramm, L., Watkins, K., and Rutherford, S. (2010): “Features that affect variability of travel time on bus rapid transit systems,” *transportation Research Record: Journal of the Transportation Research Board*, (2143), 77-84.
- [44] Schmidt P., Katzfuss M., : “Interpretation and testing of point forecast without directive,” *arXiv preprint arXiv:1506.01917*.
- [45] Senevirante, P. N. (1990): “Analysis of on-time performance of bus services using simulation,” *Journal of Transportation Engineering*, 116(4), 517-531.
- [46] Shalaby, A., Farhan, A. (2004): “Prediction model of bus arrival and departure times using AVL and APC data,” *Journal of Public Transportation*, 7(1), 3.
- [47] Strathman, J. G., and Hopper, J. R. (1993): “Empirical analysis of bus transit on-time performance,” *Transportation Research Part A: Policy and Practice*, 27(2), 93-100.
- [48] Venables, W. N. Ripley, B. D. (2002): “Modern Applied Statistics with S. Fourth Edition,” *Springer, New York*. ISBN 0-387-95457-0.
- [49] Wang, J. N., Chen, X. M., Guo, S. X. (2009, October): “Bus travel time prediction model with -support vector regression,” *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on* , (pp. 1-6). IEEE.
- [50] Wang, B., Wang W., Yang, M., Gao, L., (2012): “An approach to bus travel time prediction based on the adaptive fading Kalman filter algorithm,” *Twelfth COTA International Conference of Transportation Professionals*.
- [51] Wooldridge, J. M. (2010): “Econometric analysis of cross section and panel data,” MIT press.
- [52] Zhao, J., Dessouky, M., and Bukkapatnam, S. (2006): “Optimal slack time for schedule-based transit operations,” *Transportation Science* 40(4), 529-539.