Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN
AUTOMATICA E RICERCA OPERATIVA

Ciclo XVII

Settore Concorsuale di afferenza:
01A6-RICERCA OPERATIVA

Settore Scientifico disciplinare:
MAT09-RICERCA OPERATIVA

CLASSIFICATION ALGORITHMS FOR INTELLIGENT TRANSPORT SYSTEMS

Presentata da:
Natalia Selini Hadjidimitriou

Coordinatore Dottorato:	Relatori:
Prof. Daniele Vigo	Prof. Mauro Dell'Amico
	Prof. Daniele Vigo

Esame finale anno: 2015

Abstract

Intelligent Transport Systems (ITS) consists in the application of ICT to transport to offer new and improved services to the mobility of people and freights. While using ITS, travelers produce large quantities of data that can be collected and analysed to study their behaviour and to provide information to decision makers and planners. The thesis proposes innovative deployments of classification algorithms for Intelligent Transport System with the aim to support the decisions on traffic rerouting, bus transport demand and behaviour of two wheelers vehicles. The first part of this work provides an overview and a classification of a selection of clustering algorithms that can be implemented for the analysis of ITS data. The first contribution of this thesis is an innovative use of the agglomerative hierarchical clustering algorithm to classify similar travels in terms of their origin and destination, together with the proposal for a methodology to analyse drivers' route choice behaviour using GPS coordinates and optimal alternatives. The clusters of repetitive travels made by a sample of drivers are then analysed to compare observed route choices to the modeled alternatives. The results of the analysis show that drivers select routes that are more reliable but that are more expensive in terms of travel time. Successively, different types of users of a service that provides information on the real time arrivals of bus at stop are classified using Support Vector Machines. The results shows that the results of the classification of different types of bus transport users can be used to update or complement the census on bus transport flows. Finally, the problem of the classification of accidents made by two wheelers vehicles is presented together with possible future application of clustering methodologies aimed at identifying and classifying the different types of accidents.

Key words: clustering algorithms, hierarchical cluster analysis, Support Vector Machines, traveler behaviour, route choice, real time arrivals, big data, Intelligent Transport Systems, Power Two Wheelers, accident detection

Contents

List of Figures

List of Tables

1 Background

This chapter provides an overview of the main topics that are tackled in the thesis and of the methodologies that have been implemented or considered in the analysis.

First of all, the definition of Intelligent Transport Systems (ITS) is provided together with a classification of different types of ITS services. The rest of the chapter is completely dedicated to the description of a selection of clustering algorithms that are, in turn, further divided into groups.

The selected clustering algorithms have been considered or implemented in the analysis described in the next chapters. The approaches and methodologies were based on the specific needs of the analysis and on the type of dataset.

1.1 Intelligent Transport Systems

Intelligent Transport Systems (ITS) consists in the application of Information and Communication Technologies (ICT) to transport with the objective to offer new and improved services for the mobility of people and freights. There are different definitions of ITS, some of them focus on the applicability of ICT to different transport modes, others highlight the perspective of improving communication between all actors involved in the transport chain and underline the importance of exchanging information to develop cooperative mobility.

ERTICO-ITS Europe, the European Association that promotes the deployment of ITS in Europe, provides a definition of ITS that highlights the importance of *..the integration of information and communications technology with transport infrastructure, vehicles and users. By sharing vital information, ITS allow people to get more from transport networks, in greater safety and with less impact on the* *environment.* The capacity of the transport network can be fully exploited only if the information is exchanged between all the actors.

Similarly, the U.S. Department of Transport (DOT) describes ITS as follows: *ITS improves transportation safety and mobility and enhances global connectivity by means of productivity improvements achieved thorough the integration of advanced communication technologies into the transportation infrastructure and vehicles. Intelligent Transport System encompass a broad range of wireless and wire line communication-based information and electronics technologies.*

The EU Transport Research Knowledge Centre has a broader view of ITS that can be applied to the different transport modes with the objective of improving the management of the transport network and the service of the entire transport system.

ITSs comprise several combinations of communication, computer and control technology developed and applied in the domain of transport to improve system performance, transport safety, efficiency, productivity, service, environment, energy, and mobility. ITS can be applied to the transport infrastructure of highways, streets, and bridges, as well as to a growing number of vehicles, including cars, buses, trucks, and trains. These systems can be used both for passenger and for freight transport. These technologies provide a new means of improving the service quality and management of the transport system.

Finally, the EU project HUMANIST (Human Centred Design for Information Society Technologies) emphasises the application of ICT technologies to transport by defining ITS as the *application of advanced sensor, computer electronics, and communication technologies and management strategies – in an integrated manner – in order to increase the safety and efficiency parameters of the transportation system.*

Furthermore the project proposes a classification of ITS based on the type of installation (in-vehicle system, nomadic device, etc.), type of traveller (i.e. commuter, tourist), type of human-machine interface (i.e. warning message, provision of information, etc.), type of prospect use (traffic information, freight and fleet management, etc.).

An additional classification of ITS systems can be found in the "Market Analysis of the Intelligent Transport System and Services (ITSs) Sector", a report prepared in 2008 by innovITS, UK. The document proposes seven classes of ITS themes based on the product and service they provide: improving network management, improving road safety, improve travellers information, improve reliability and efficiency of public services, reduce the environmental impact, supporting security, automotive telematic, communication services.

ITS are able to modify users' behaviour by providing them information that would be difficult to collect otherwise. New information systems have the capability to guide the travellers to better exploit the existing transportation systems, services and infrastructures. The result of the deployment of ITS is the production of large amount of data that can be used for different purposes, including the study of travellers' behaviour.

This thesis is focused on the classification of ITS data with the aim to describe transport users behaviour by proposing new methodologies of data analysing, to provide information that can be of value to decision makers, transport planners and public authorities, to improve the supply of transport services and the provision of infrastructure.

The two terms *clustering* and *classification* are often used as synonyms. However they have a slightly different meaning in the field of data analysis. The difference relies on the objective of the classification and on the existence of previous knowledge on the structure of the data set.

In this work, the term *clustering* is referred to the algorithms that provide information on the structure of the data without having any previous knowledge on them. The term *classification* is instead used for techniques that imply a good a-priori knowledge and for which the objective is to classify observations based on these knowledges.

1.2 Machine Learning

Machine Learning (ML) is defined by Mitchell (1997) as *the discipline that seeks to find the fundamental laws that underline the learning processes with the objective to create a computer programme that can improve automatically, with the experience.*

Finding the law that underlines a phenomenon is important for several reasons. In the transportation field, for instance, it could be possible to use historical data on traffic flows to forecast future behaviour and reroute traffic, with the final objective of improving congestion on the road network.

One of the main field of research of ML deals with clustering. The problem is to develop algorithms based on the observations, to extract behavioural information or patterns that can describe a phenomenon. ML distinguishes between two main approaches: *supervised* and *unsupervised learning.*

In *unsupervised learning*, the natural pattern is sought directly based on the data structure (Duda et al. (2001)), while in *supervised learning*, a set is provided along with the corresponding categories and associated costs, to train a classification function. The selection of one of the two approaches mainly depends on the a-priory knowledge on the dataset and or on the objectives of the analysis.

The next section describes a selection of well-known clustering algorithms that belong to the two main categories of algorithms described by ML. The classification problems are formulated based on the Linear Programming approach.

1.3 Unsupervised learning

Unsupervised learning approaches seek to find the structure of the dataset without having a priori knowledge on the grouping of data. Hansen and Jaumard (1997) describe clustering from the mathematical programming point of view. They create a list of characteristics that a clustering problem need to tackle and focus on two main relevant aspects for mathematical programming:

- the identification of the constraints,
- the implementation of the algorithms to solve the classification problem.

A general formulation of the two main classes of unsupervised clustering problems, the *Partitional* and *Hierarchical clustering* is provided by Jain et al. (1999) and Xu (2005). The difference between the two approaches relies on the procedures thorough which the clusters are created.

1.3.1 Metric labelling problem

The mathematical generalization of the classification problem is called *metric labelling problem* (Kleinberg and Tardos (2002)). The *metric labelling problem* can be view as a discrete optimization problem where the objective is to optimize a function that assigns labels to objects. The problem is also related to the *uncapacitated quadratic assignment problem* (QAP). Here the objective is to find a matching between a set of n activities and a set of l locations so that the sum of all costs of the activities and the flows between locations are minimized. The problem is NP-hard.

Given a set V of v objects and a set L of l possible labels, the objective of the *metric labelling problem* is to find a mapping function $f\colon V \longrightarrow L$ that assigns a label to the n objects.

The cost of labelling two objects, (u, v), by f(v) and f(u) is:

$$w(u,v)d(f(v),f(u)) \tag{1.1}$$

where $w(u,v)$ is the strength of the relationship between u and v (Chekuri et al. (2001)).

The relation between similar objects can be represented in a graph in which, two vertexes are connected by an edge if they have the same label. Specifically, given an undirected weighted graph G=(V, E), the goal is to find the minimum cost labelling, where the cost can be expressed in terms of different metrics between two connected vertexes. The cost function of the discrete metric labelling problem is:

$$Q(f) = \sum_{u \in V} c(u, f(u)) + \sum_{(u,v) \in E} w(u,v)d(f(v),f(u)) \tag{1.2}$$

The first part of equation 1.2 refers to the vertex labelling cost and it expresses the prior knowledge on the observation and how likely the a label is assigned to an observation. Where $f(u)$ is the function that assigns a label to an observation u so that $c(u,l)$ is the cost of the assignment of a label l to the observation u.

The second part of equation 1.2 is the edge separation costs which defines the relation existing between the objects. The weight $w(u,v)$ is a prior estimate of this relation which expresses how likely two objects are assigned with the same label l. Finally $d(f(v),f(u))$ is the metric to determine the similarity or dissimilarity between pair of objects.

The objective of the problem is to minimize $Q(f)$ such that there is no negative cost of labelling a vertex v with a label l:

$$c(v,l) \geq 0 \quad \forall v \in V, l \in L \tag{1.3}$$

Kleinberg and Tardos (1999) solve the problem in polynomial time $O(\log\ l \log \log\ l)$, where l are the labels. However the best approximation of the problem is $O(log\ k)$ which is the complexity of the algorithm developed by Fakcharoenphol et al. (2003).

1.3.2 Measures of proximity

Jain and Dubes (1988) present different metrics to determine the proximity between observations. The pair comparisons of observations are usually represented in *similarity* or *dissimilarity* matrices.

Given a set of observations, the rows of the matrix are usually called *patterns* or *objects* and the columns *features* or *attributes*:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

The Dissimilarity matrix is expressed as follows:

$$\begin{bmatrix} 0 & d(1,2) & d(1,3) & \cdots & d(1,n) \\ & 0 & d(2,3) & \cdots & d(2,n) \\ & & 0 & \cdots & \cdots \\ & & & & 0 \end{bmatrix}$$

A measure of similarity can be a dummy, a discrete or continuous variable. In case of dummy variables, 1 indicates that two observations are included in a cluster, 0 otherwise. An example of continuous metric is the Euclidean distance.

The indicator of similarity or dissimilarity between i and j patterns must fulfill the following properties:

For dissimilarity measures:

$$d(i,i) = 0 \quad \forall i \tag{1.4}$$

For similarity measures:

$$d(i,i) \geq \max d(i,j) \quad \forall i \tag{1.5}$$

Symmetry:

$$d(i,j) = d(i,j) \quad \forall (i,j) \tag{1.6}$$

Positivity:

$$d(i,j) \geq 0 \quad \forall (i,j) \tag{1.7}$$

The most general form of metrics is the Minkowski distance which is computed as follows:

$$d(i,j) = (\sum_{j=1}^{n} |x_{ij} - x_{kj}|^r)^{\frac{1}{r}} \quad where \quad r \geq 1 \tag{1.8}$$

If $r = 2$, the Minkowski metrics is the Euclidean distance. If $r = 1$, the metrics is called Manhattan distance because it computes the distance that would be travelled to get from one data point to another if the a grid-like path was followed. The larger is r the greater is the weight that is assigned to larger distances.

This metric has to satisfy the *triangle inequality* for which the sum of the lengths of any two edges of a triangle must be greater than or equal to the length of the remaining side:

$$d(i,k) \leq d(i,m) + d(m,k), \quad \forall i,k,m \tag{1.9}$$

However, not all measures must satisfy the triangle inequality. for instance the *mutual neighbour distance* (Abonyi and Feil (2007)) is computed as follows:

$$MND(x_i, x_j) = NN(x_i, x_j) + NN(x_j, x_i) \tag{1.10}$$

where $NN(x_i, x_j)$ is the neighbour number of x_j with respect to x_i.

1.3.3 Hierarchical clustering algorithm

The *Hierarchical clustering* creates a nested series of partitions $C_1, C_2, ..C_m$ of a set of observations X:

$$X = x_1, x_2, ..., x_n \tag{1.11}$$

The partitions satisfy the two conditions:

$$C_i \cap C_j = \emptyset \quad for \quad all \quad i \quad and \quad j \quad from \quad i \quad to \quad m, \quad i \neq j \tag{1.12}$$

and

$$C_1 \neq C_2 ... C_m \equiv X \tag{1.13}$$

Clusters can be represented using sets of observations, matrices, hyperplanes or dendrograms. A dendrogram (Figure 1.1) is a tree representation in which the length of leaves provides a visual representation of the similarity or correlation between observations, and the nodes are the clusters. For instance, observations 2, 4 and 5 are highly correlated, while observations 1 and 3 (and 6) belong to another cluster in which the elements are less correlated comparing to the others.

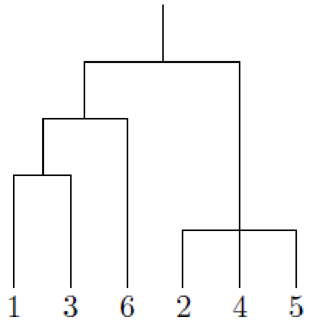


Figure 1.1: An example of dendrogram

Depending on how the nodes are created, there are two possible approaches to hierarchical clustering: a top-down or a bottom-up. The *Agglomerative* clustering starts from the leaves (the bottom) and aggregates the observations until a unique cluster is obtained; *Divisive clustering* starts from the top until all observations are separated into single clusters.

A description of the *Agglomerative Hierarchical clustering* is provided by Johnson (1967). The algorithm starts with each observation that forms a singleton cluster and proceeds as follows:

1. compute a similarity matrix between all pairs of clusters (define a metric),
2. merge the most similar pair of clusters,
3. update the similarity matrix accordingly, by computing the distances between the newly formed clusters (merged clusters cannot be separated),
4. steps 2 and 3 are repeated until a single cluster that includes all the observations remains.

The *Single Linkage method* (Sneath (1957)) merges the groups based on the minimum distances between two clusters, reported in the dissimlarity matrix.

$$d(i+j,k) = \min d(i,k), d(j,k) \tag{1.14}$$

Since the algorithm is based on minimum distancesm it tends to form one large cluster with few small clusters or single elements around it. In the last steps of the analysis, it is possible to have the so called *chaining effect* which is the creation of a sort of bridge between two different clusters that provokes their fusion in one single cluster (see Figure 1.2).

Similarly, the *Complete linkage* merges the groups based on their maximum distance between two objects in two groups:

$$d(i+j,k) = \max d(i,k), d(j,k) \tag{1.15}$$

This method is strongly affected by outliers because it is based on the maximum distances.

The *Ward method* is a different approach comparing to the previous ones because it is based on the analysis of variance (Ward (1963)). It is an agglomerative clustering algorithm because it starts from n clusters of size 1. The method consists in

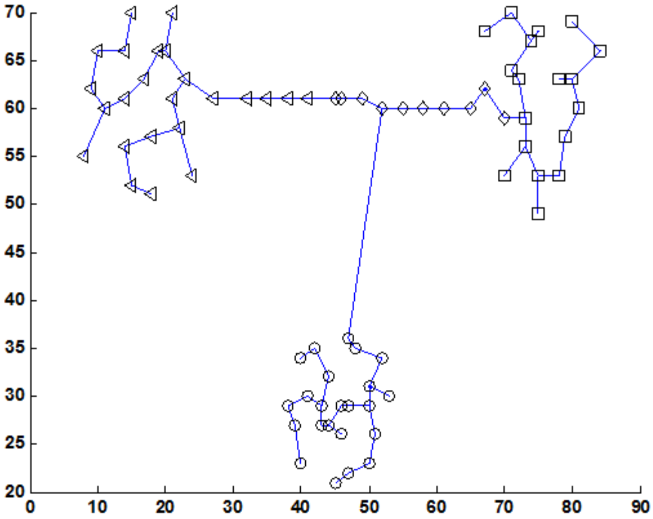


Figure 1.2: Chaining effects

the minimization of the sum of squared errors. If X_{ijk} is the variable k for the observation j in the cluster i, the error sum of squares is:

$$SSE = \sum_i \sum_j \sum_k \mid X_{ijk} - \mu_{ik} \mid^2 \tag{1.16}$$

At each step the algorithm merges the observations that imply the minimum increase of within cluster variance.

The difficulty of hierarchical methods relies on the decision on when to stop the iteration such that an appropriate number of clusters is created. There is not a general rule to take this decision. Generally, the decision is based on dendrogram. However, for large dataset, the visualization of the clusters in a dendrogram becomes difficult and other methodologies should be developed. All Hierarchical clustering methods consist in computing the similarity between all pairs of observations which implies a complexity of at least $O(n^2)$.

1.3.4 Partitioning clustering algorithm

A general definition of the partitioning clustering methods is provided by Jain and Dubes (1988). Given n patterns in a n-dimensional metric space, the method consists in determining a partition of the patterns in K groups, such that the patterns in a cluster are more similar to each other than to patterns in different clusters.

The global optimum of a partitioning cluster can be obtained by enumerating all the possible partitions. Since this is not always feasible because the enumeration of all possible partitions is manageable only for very small partitions, greedy algorithms are often used. For instance an algorithm starts from an initial partition and relocates objects from one cluster to another (i.e. swapping) until the value of the objective function improves. This is one of the most common approach that usually leads to local minima.

k-means

In the k-means algorithm each cluster is represented by its centroid. The k number of clusters to be created is decided a-priori. The k centroids are selected at random and the closest elements to the centroid are grouped into the cluster formed by that nearest centroid. The centroids of newly formed groups are then recomputed and a new iterations starts. Jain and Dubes (1988) underlines that, although many clustering algorithms have been proposed since the first time the k-means algorithm was introduced in 1955, the k-means remains the most used. This evidences how difficult it is to develop a general method that can be deployed for the solution of different problems.

Given a vector of N observations $X = x_1, ..., x_j, ..., x_N$ to be clustered in c_k clusters, the objective of the *k-means* is to minimize the squared sum of errors between the center of the cluster and the observations. Thus, for each cluster the relation is the following:

$$SSE_k = \sum_{x_i \in c_k} \mid x_i - \mu_k \mid^2 \tag{1.17}$$

The objective is to minimize the global function which is an NP-hard problem:

$$SSE = \sum_{k=1}^{K} \sum_{x_i \in c_k} \mid x_i - \mu_k \mid^2 \tag{1.18}$$

The basic algorithm works as follows:

1. select k points as initial centroids,
2. assign all points to the closest centroid,
3. recompute the centroids of each cluster,
4. repeat 2 and 3 until the centroid do not change.

The advantage of this procedure is its simplicity. However the need to determine *a priori* the number of clusters represents a disadvantage in a situation where the underlying structure of data is unknown. Furthermore the algorithm is very sensitive to outliers. On the other hand, the algorithm is computationally efficient so that it is fast in case of large databases. The complexity of the algorithm is O(vlt), where v is the number of elements to be grouped, l is the number of groups and t is the number of iterations.

The main problem of this method is that the algorithm does not guarantee to find global optimum. To improve the results, the algorithm could be run several times in order to select the best among a set of results. Several procedures to improve the results of the clustering have been proposed in the literature. However most of them are based on the selection of the best centroid of the clusters.

Thousands of bariations of the k-means have been proposed in the past fifteen years. For instance Dong and Qi (2009) proposed a new heuristic algorithm to overcome the main disadvantages of the k-means optimization algorithms that consists in the lack of resolution due to the dependability of the solution from the initial centroids that are selected and in the solution itself that consists of local minima. Good results in terms of clusters have been obtained by Fred and Jain (2002) who combined different partitions obtained by the multiple implementation of the k-means to the same dataset. Basically, they created the final groups based on the most frequent partitions. Finally, another example is provided by Mingoti and Lima (2006) who performed a comparison between a selection of non hierarchical and hierarchical clustering algorithms and found that hierarchical clustering and k-means presented similar performance.

1.3.5 Evaluation of the quality of clustering

The accuracy of the partition obtained using a clustering algorithm depends on the method that is used (Mingoti and Lima (2006)) and on the structure and the type of data that need to be grouped. To evaluate the performance of clustering algorithms, artificial data with overlapping and non overlapping clusters are usually created and the result of the clustering is compared to the true simulated data. The simulated data are usually created from a multivariate normal distribution. Milligan (1985), for instance, generated artificial data including different types of errors, and compared the performance of 14 hierarchical clustering algorithms including the k-means. He found that all algorithms were affected by the presence of irrelevant variables but the single linkage and the k-means methods were robust against the inclusion of outliers. Furthermore it was found that the single linkage algorithm suffers from the chaining effects, for which similar objects may be assigned to different clusters (Cormack (1971)).

The *entropy* can be used to measure the quality of clustering. The best entropy is obtained when the cluster includes only one observation. If p_{ik} is the probability that cluster k includes the element i, the entropy of each cluster k is computed as follows:

$$E_k = -\sum_i p_{ik} log(p_{ik}) \tag{1.19}$$

The total entropy is obtained as the weighted sum of the entropies, where the weights are the number of elements included in each cluster:

$$E_{tot} = \sum_k \frac{n_k E_k}{n} \tag{1.20}$$

Finally, the Cophenetic coefficient (Jain and Dubes (1988)) measures is used to evaluate the hierarchical clustering structures. The indicator evaluates the degree of similarity between the similarity matrix and the proximity level when two pairs of observations are grouped in the same cluster for the first time. Specifically the Cophenetic index measures how much a dendrogram preserves the distances between the original data points.

1.4 Supervised learning

The advantage of supervised learning algorithms is that prior knowledge on the classification results can be used to solve the problem (Duda et al. (2001)). Specifically, a set of observations is used to train a classification function. The objective is to find the separating hyperplanes that divide the sample in two or more classes. The distance between each observation and the separating vector has to be maximised.

The following section focuses on the description of Support Vector Machines. The simplest case is when the observations are linearly separable and there are only two classes.

1.4.1 Linear separability

Let two sets P$\subseteq \mathbb{R}^d$ and Q$\subseteq \mathbb{R}^d$. P and Q are linearly separable if there exists a hyperplane that separates them. Figure 1.3 shows a set of observations that are linearly separable (a), and a set which is not linearly separable because different observations cannot be grouped by a line that separates them in two planes (b).

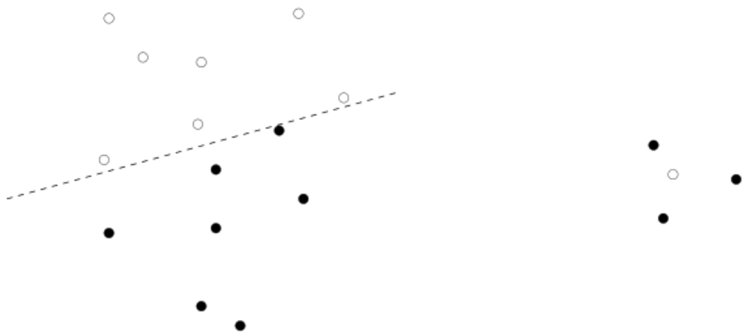


Figure 1.3: (a) linearly separable set - (b) non-linearly separable set

A hyperplane defines two half spaces H_1 and H_2. If a set of observations is linearly separable, each observation must satisfy one of the two conditions:

$$H_1 : x_i \cdot w + b = +1 \tag{1.21}$$

$$H_2 : x_i \cdot w + b = -1 \tag{1.22}$$

To demonstrate that a set of observations is linear separable it is necessary to determine if the hyperplanes exist or not. The test for linear separability can be formalized as a Linear Programming problem where the objective is to find the real numbers $X{=}x_1, x_2, ..., x_N$ such that a specific weight function is maximized subject to a constraint.

$$max \quad \sum_{j=1}^{N} c_j x_j \tag{1.23}$$

subject to:

$$\sum_{j=1}^{N} a_{i,j} x_j \geq b_i \tag{1.24}$$

The intersection of hyperplanes is the feasibility region. If the equation 1.23 is set equal to γ, the objective is to find the maximum value of γ such that the constraint is satisfied. It is not always possible to find an optimal solution. In some cases there might be unbounded solutions or infeasible solutions. In these cases, the set of observations is not linearly separable.

If the dataset is linearly separable, the *perceptron learning algorithm* can be used for the classification. The algorithm converge in a finite number of iterations if there exist a separating hyperplane.

1.4.2 Support Vector Machine

Support Vector Machine (SVM) consists in a function φ that maps data in a number of dimensions that correspond to the number of classes in which the observations have to be classified. The objective is to maximise the distance of separating hyperplane from the closest observations (*support vectors*).

Burger (1998) presents a formulation of the problem that can be seen as a specific case of the *metric labelling problem*. The two labels assigned to the training data can be either positive or negative: $\mathbf{x_i}$ is the vector of observations, y_j is the label assigned to the each observation. When there are only two possible classes, the problem is binary and each observation can have either a positive or a negative value.

$$\{\mathbf{x_i}, y_j\} \quad \text{where} \quad i = 1, ..., N \quad \text{and} \quad y_j \in \{-1, 1\}, \mathbf{x_i} \in R^d \tag{1.25}$$

Training a Support Vector Machine means finding the optimal hyperplane which is the one that has the maximum distance from the training set. The closest observations to the hyperplane are called *support vectors*. The distance between the hyperplane and the nearest negative or positive support vectors is called *margin*. The goal of SVM is to find the separating hyperplane that maximise the *margin* because the larger the margin is, the better a classifier can be generalized.

Figure 1.4 shows the two support vectors separated by a hyperplane. The objective is to find a decision rule for which it is possible to assign each observation to its corresponding hyperplane.

The procedure consists in the identification of the vector $\mathbf{x_i}$ from the origin of the plane to the direction of the observation and its perpendicular projection allows to determine where the observation is placed. The optimization problem consists in finding the vector $\mathbf{w}$ that is perpendicular to the hyperplane.

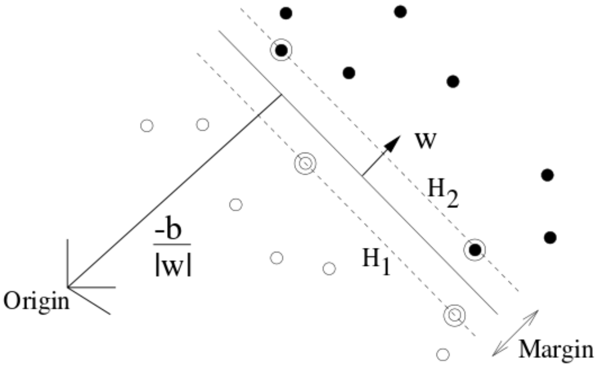


Figure 1.4: Hyperplanes and support vectors. Source: Burger (1998)

The dot product of the two vectors, $\mathbf{w}$ and $\mathbf{x}$, provides the perpendicular projection of the vector $\mathbf{x}$. The decision rule consists in setting a constant, b, that provides an indication on which partition the observation lies on:

$$\mathbf{w} \cdot \mathbf{x} \geq b \tag{1.26}$$

Inequality 1.26 can also be expressed in the form:

$$\mathbf{w} \cdot \mathbf{x} + c \geq 0 \tag{1.27}$$

where c = - b.

The problem is that both, the constant and the perpendicular vector, are unknown. Depending on which of the following two conditions is satisfied, the observation has one of the two labels $\{+1, -1\}$:

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq +1 \quad for \quad y_i = +1 \tag{1.28}$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \quad for \quad y_i = -1 \tag{1.29}$$

The two constraints can be combined into one relation:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 = 0 \tag{1.30}$$

As said, the objective is to maximize the margin included between the support vectors. The more the distance between the support vectors, the better is the classifier. If the sample is not linearly separable, the objective of SVM is to maximize the margin plus a penalty that is applied for pointing the line to the wrong side of the plane.

The measure of the width of the margin is provided by $\frac{1}{||\mathbf{w}||}$ which is the distance that has to be maximized. However this is also equivalent to minimize the squared norm of the perpendicular vector $\frac{1}{2}\left\|\mathbf{w}\right\|^2$ under the linear constraints 1.30. This implies solving a *quadratic optimization problem.*

At this point, it is possible to use the Lagrange multipliers to minimize the objective function without taking care of the constraints that are instead introduced in the objective function.

1.4.3 Lagrangian formulation of the problem

This section describes the procedure to replace the constraints of the equation 1.30 with the Lagrangian multipliers α_i. The training set appears in the form of dot products between vectors thus allowing the generalization to the non linear case.

In this formulation, the constraints are multiplied with the Lagrangian multipliers and subtracted from the objective function (see equation 1.31).

$$L_P \equiv \frac{1}{2}\left\|\mathbf{w}\right\|^2 - \sum_i \alpha_i \left[y_i(\mathbf{x_i}\cdot\mathbf{w} + b) - 1\right] \tag{1.31}$$

The problem is to solve a convex quadratic programming problem in which L_P has to be minimized with respect to **w** and b. The main characteristics of the convex optimization problem is that any local solution is also an optimal solution.

The derivatives of L_P with respect to the vector **w** leads to a linear relation which is a function of the observations:

$$\frac{\partial L_P}{\partial \mathbf{w}} = \sum_i \alpha_i \cdot y_i \cdot \mathbf{x_i} \tag{1.32}$$

The derivative of L_P with respect to b is:

$$\frac{\partial L_P}{\partial b} = \sum_i \alpha_i \cdot y_i = 0 \tag{1.33}$$

To solve the optimization problem, 1.32 is put back into the objective function 1.31 and the so called Wolfe dual is obtained. What is obtained is a quadratic optimization problem with two linear constraints. The problem can be solved with different algorithms. Two of them will be described in the next paragraphs.

$$\min L_P \equiv \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \tag{1.34}$$

subject to:

$$\alpha_i \geq 0 \tag{1.35}$$

and

$$\sum_i \alpha_i y_i = 0 \tag{1.36}$$

This optimization function 1.34 depends on couples of observations $(x_i x_j)$. But most importantly, the optimization problem depends on their dot product. For instance, given two transformation functions, the problem is to maximize their dot product:

$$\varphi(x_i) \cdot \varphi(x_j) \tag{1.37}$$

To maximize 1.37, all is needed is a Kernel function of the two observations. The advantage of this method is that there is no need to know the transformation function of the two observations (Winston (2010)). The kernel function of x_i and x_j is defined as:

$$K_{ij} = k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \tag{1.38}$$

where K_{ij} is the similarity between i_{th} and j_{th} elements.

There is no methodology that describes how to select the Kernel function that allows to obtain a good classification of data. Therefore the procedure usually consists in trying different functions and selecting the one that yields to the best classification results.

The most popular Kernel functions are:

Linear:

$$k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathbf{T}}\mathbf{y} \tag{1.39}$$

Quadratic:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathbf{T}}\mathbf{y})^2 \tag{1.40}$$

Polynomial:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathbf{T}}\mathbf{y})^d \tag{1.41}$$

The Kernel function are used when the dataset is not linearly separable. The function allows to map the data into a multidimensional space. Kernel functions allows to manipulated data as they are projected into a multidimensional space but by operating at its original space (Mitchell (1997)).

Cross-validation consists in splitting the data set in a number of equal partitions. Then each partition of data is used for testing and for training. For instance, two-third of the data are used for testing and one-third for training. Usually, a three-fold cross-validation is performed. Cross validation is used to determine the values of Kernel parameters.

1.4.4 Multiclass Support Vector Machine

Multiclass SVM deals with classification problems where the objective is to classify observations in more than two classes.

The most commonly used approach is the *one-versus-the-rest* which is also called OVA (One-Versus-All). The method consists in solving a number of binary support vector machines. The first step is to train a first positive class, k_1, against all the other classes (N - 1) as they are part of a single negative class. If $f_k(x)$ is the solution of the kth problem, the method consists in the selection of the solution that gives the largest positive value.

An overview of the literature on OVA method is presented by Rifkin and Klautau (2004). They distinguish between two main classes of methods: the *single machine* that consists in solving a single optimization problem and the *error correcting* approaches that consist in the selection of binary classifiers and in the development of methods to combine them.

The other approach is called *one-versus-one* and it consists in training all the N(N-1)/2 couple of classes. The result of the classification is based on the overall results of the classification. That is each observation is assigned to a certain class accordingly to the number of times (or "votes") it has been assigned to that class.

1.4.5 SMO Algorithm

The *Sequential Minimal Optimization* is based on the Chunking algorithm which in the optimization of few variables at time, with the aim to control the dimensionality of the problem. It is in fact computationally more effient to handle matrices with lower dimensions. For instance, a quadratic problem with 20 variables involves a quadratic form of 20 by 20 matrix which is easier to compute respect to a 1000 by 1000 matrix.

The *Sequential Minimal Optimization* algorithm is implemented to solve the minimization of 1.34 subject to the two linear constraints 1.35 and 1.36. The algorithm decomposes the quadratic optimization problem in several sub-problems and, at every step it selects the smallest problem. The smallest problem consists in the two Langrangian multipliers α_i and α_j because with only one multiplier it would not be possible to satisfy the linear constraints. At each step, a couple of multipliers is selected such that the objective function is optimized subject to the constraints and the Support Vectors are updated to reflect the new optimal solution. The SMO consists in solving the two Lagrangian multipliers, in a heuristic to find the two Lagrangian to optimize and in the computation of the threshold b (Platt (1998)).

1.5 Other examples of classification algorithms for ITS data

Classification algorithms have been widely deployed for the analysis of ITS data. One main research area deals with the prediction of travel times on road networks. For instance, Lee et al. (2008) propose a Naïve Bayesian Classification (NBC) for the prediction of link travel time. The method deploys historical traffic data and the result is a label with the most probable velocity for the road link. The Bayesian classification is a probabilistic approach that aims to estimate the underline distribution of the data. The NBC is based on the Bayes formula and it is called Naïve because it assumes the independence of events which is a strong simplification. The assumption of independence of the events allows to multiply their probabilities. The result is the probability of the outcome of an event (i.e. probability of velocity on a road link) without knowing anything about that particular day (Witten et al. (2000)).

The *fuzzy clustering* differentiate from the previous methods on the possibility to create overlapping clusters. The condition of mutual disjunction is relaxed for instance by the introduction of a degree of membership (Hruschka (2009)).

Density based algorithms assumes that the observations are derived from a probability distribution thus the objective is to identify the clusters and to find the parameters of the distribution. The main advantage of this algorithm is that it is able to identify the underlying structure of the cluster and, differently from the other approaches, it does not leads to convex solutions. The main idea is to identify the clusters basing on their density which also allows to identify the noise (i.e. observations that do not belong to any cluster). Given a radius from each observation, the density within it has to be above a certain threshold, so that a minimum number of observations (MinPt) can be included into the cluster.

Ester et al. (1996) describe the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm where the *Eps-neighborhood* of a point u, denoted as $N_{Eps}(u)$ is defined by the relation:

$$N_{Eps}(u) = \{v \in N | dist(u,v) \leq Eps\} \tag{1.42}$$

Since the method consists in the measurement of the density around each point, there will be some of them that will be placed at the border of a cluster where observations are less dense. Therefore the authors introduce the definition of *directly density-reachable* and *density-connected* point. Specifically, a point u is *directly density-reachable* from a point v wrt Eps and MinPt, if:

$$u \in N_{Eps}(v) \tag{1.43}$$

$$|Eps(v)| \geq MinPt \tag{1.44}$$

A point u is *density-connected* wrt Eps and MinPt if there exists a point o such that u and v are both density reachable from o wrt Eps and MinPt.

A good accuracy of the prediction of travel time is obtained using Support Vector Regression (SVR). For instance, Chun-Hsin et al. (2003) apply SVR for time series forecasting with the idea to determine a function that can approximate future values of travel time based on past observations. Another example of travel time prediction is provided by Asif et al. (2012) who use a combination of clustering algorithms to predict traffic in a road network using GPS coordinates. Here the problem is to handle large data set such as the ones produced by GPS coordinates. The Support Vector Regression is initially applied to obtain a large scale traffic prediction. Usually, 50 days of historical data are used to train the classification function and make 10-days predictions. Successively, two unsupervised algorithms, the k-means and the Principal Component Analysis (PCA), are implemented to extract spatial and temporal prediction in large networks. Where the cluster analysis is deployed to group roads with similar prediction performances during different time periods.

One of the objectives of ITS is to reduce traffic congestion. Lee et al. (2011) describe how ITS and classification techniques can help to reduce traffic. If all trip trajectories and destinations are labelled, a classifier could be trained to predict a new vehicle destination. The information on vehicle destination are important when the objective is to reroute traffic in order to reduce congestion.

The method of clustering trajectories consists in the assignment of a class to each trajectory which is characterised by a set of features. The trajectories in a road network can be represented in a graph, where the vertexes are the intersections and the edges are the segments of the trajectory (the features). Furthermore, different trajectories can have similar features and it is important to know the sequence of the features over time. Lee et al. (2011) propose a frequent pattern-based classification of trajectories on the road network. The method allows to take into account the order each road segment is visited and to consider the spatial and temporal information on the travels.

1.6 Conclusion

The chapter has described few main clustering algorithms that can be implemented for the analysis of ITS data. The selection of the most appropriated clustering algorithm should be based on whether there is some knowledge on the structure of the dataset or not. The literature has proposed a general formulation of the clustering problem which is called metric labelling problem. However the clustering algorithm has been developed based on the structure of the dataset and on the objectives of the analysis. If the structure of the dataset is unknown, the best choice is to select unsupervised clustering algorithms. Even within this category of classification algorithms, the are many possible choices and many variants of the most commonly used algorithms that can be found in the existing literature.

It is possible to distinguish between two main approaches, supervised and unsupervised learning algorithms. Two examples of algorithms of unsupervised type are the hierarchical and k-means. The main advantage of the Hierarchical clustering algorithm is that there is no need to indicate the number of clusters that need to be created. However, hierarchical approached have a time complexity with is at least O(n^2), where n is the number of observations to be clustered. This can be problematic especially in case of large dataset such as the ones collected using ITS technologies. Another disadvantage of Hierarchical approaches relies on the fact that is not possible to separate clusters once they have been created which does not allow to improve the solution during the execution of the algorithm. The k-means that takes as input the number of clusters to be created. This can be a considerable disadvantage especially when the objective is to find the natural structure of the clustered data. However the algorithm is efficient, its complexity is in fact $O(tcn)$ where t is the number of iterations, c is the number of clusters to be created, n is the number of observations. The main disadvantage of this algorithm is that it often terminated in local optima. However the exists many proposals to improve the algorithm such as the use of metaheuristics. The quality of clustering can be evaluated using several approaches such as the measure of variance within and between the clusters. However, in case of georeferenced database, it is convenient to visualise the results with a GIS (Geographic Information System).

In case of supervised learning algorithms the structure of the dataset has to be known in advance. The methodology consists in the classification of the entire dataset by training a classification function. The main innovation of Support Vector Machines relies on the possibility to select a kernel function to obtain a classification of the observed data on several dimensions (i.e. depending on the number of classes in which the dataset has to be divided).

The design of a general algorithm that can be deployed for different clustering purposes is a difficult task. This is demonstrated by the fact that the k-means algorithm have been proposed more than 50 years ago but it is the most commonly used in all its variations.

2 Hierarchical cluster analysis of car floating data

2.1 Introduction

An ambition of the local authorities is to be able to predict the drivers' choices so that appropriate measures for dealing with traffic management or infrastructure upgrade can be adopted. Route choice behaviour has been studied in the past from several points of views to derive information on the driver performances and on the elements that determine the drivers' selection process. Lin and Gong (2012) and Maag and Krüger (2010) propose a route choice behaviour-cognitive model to describe the decision making process of drivers.Mahmassani et al. (1990) modelled the interaction of route choices with several variables and found that geography, network condition and information on the traffic conditions had a strong influence on route choices.

GPS coordinates can help to obtain more accurate information on drivers' route choices as evidenced by Schonfelder and Axhausen (2003) who underline the potential of GPS data for travel behaviour research, especially when studying the variability over time. Furthermore the use of Global Positioning System (GPS) navigation systems for private route guidance, has increased the availability of positioning data. For instance, the Borlänge GPS dataset described in Axhausen et al. (2004) involved 260 private and commercial cars for almost two years. The analysis of behavioural patterns of travels using GPS trips is usually built on the Activity-Based Approach that recognizes that transport is a derived demand and that there is a strong relation between travel behaviour and human activities (see Jones et al. (1990)). Basically these studies use explanatory models trying to describe the observed behaviours from a sample of driver's routes. Our work differentiate from these studies in the methodology and in the approach.

We use the GPS tracks recorded during a test of an European project, involving 89 drivers monitored for 17 months using on-vehicle dataloggers which produced 42.000 routes. The first innovation that is proposed is to use the observed data, which span almost all the territory of the test and cover all the daily hours, to compute a good and realistic approximation of the travel speed. Indeed, several researches from the literature compared observed routes with shortest time routes, but the speeds used to compute the shortest path not necessarily fit with the real speeds during the test period. The second innovation is to use empirical GPS data without taking into account the drivers characteristics, but grouping the individuals by similar missions that are identified as moving from a given restricted area to another restricted area. The aim is to identify if there is a common behaviour for several individuals moving between two given zones. This information has great value for local authorities, since it describes the way most of the drivers moves on the territory. The knowledge on how drivers behave between zones of the Province will allow to support local authorities' decisions concerning traffic management. New network configurations could be conceived based on actual drivers route choice behaviour or new information could be included in the traffic models to have more realistic simulation results.

The remainder of the chapter is organised as follows. In paragraph 2.3 and 2.4 the dataset used and the methodology developed in this study are described. Specifically, an iterative method to compute average travel velocities from observed data is described as well as how the four optimal alternatives are computed. Section 2.5 presents the results of the analysis derived from the comparison between the characteristics of the observed and optimal alternative paths. Furthermore route choice behaviour in geographical macro areas is analysed using shortest path optimization. Finally, a clustering procedure of paths' origins and destinations is implemented to identify groups of habitual paths and a new index is proposed. A further analysis of the road links with the higher deviation from the optimal alternatives provides important information on drivers' behaviour. In this work, route, path and travel are used as synonyms.

2.2 Literature review

The first theoretical studies of traffic flows go back to Wardrop and Whitehead (1952) which propose mathematical methods based on shortest paths and on statistical analysis. Wolfe (1956) improved upon these methods and modeled the traffic equilibrium as a *Quadratic Assignment Problem*, one of the hardest problems in combinatorial optimization (see, e.g., Burkard R.E. (2009)). These models are based on the assumption that there is a sort of cooperation between drivers, which can resemble the modern *Swarm Intelligence* (Beni and Wang (1993)), that induces the drivers to choose routes giving a local minimum for the system, in which no individual can reduce its travelling time without increasing the travelling time of other individuals. These models, however, do not consider that the driver choices may be affected by factor as familiarity with a route, avoiding congestion, avoiding traffic lights, directness of the route, which are not related to a global objective function, but to personal preferences. Papinski and Scott (2011) studied some of the factors affecting route choice by considering a sample of 31 individuals who stated that their objective is to minimize their home-to-work travel time and the analysis proved that they selected direct routes and routes that allow to avoid congestion.

Information on the paths followed by a drive can be collected either using a questionnaire or by automatically collecting information using GPS data. A survey allows to gather information on the purpose of the travel, the motivation of specific choices, and other factors that do not have an immediate dependence from position and time. The collection of GPS trajectories offers the possibility to deploy more accurate and complete information on travel characteristics such as the spatial location of the travel, the travel time or the timestamp. There is a reduced cost and increased accuracy, compared to manual surveys, but no direct access to personal motivations. Some works, see, e.g. Wolf et al. (2001), Axhausen et al. (2004), Tsui and Shalaby (2006) are focused on the development of methodologies trying to infer from GPS trajectories additional information as the purpose of the trip. However these works cannot explain all cases, as when the trip terminates in the middle of a road or in unidentifiable location. Moreover the sample used is generally small (9 individuals in Tsui and Shalaby (2006) and 24 from the Borlänge dataset). This work uses the entire cleaned dataset and an aggregate statistical analysis is performed to understand the driver's behaviour. Specific driver's choices are recorded and observed, but do not affect the final results when it is possible to prove that the result is statistically significant.

To observe routes the researcher have mainly adopted shortest distance and shortest time paths for comparison (see, e.g., Prato and Bekhor (2006), Jan et al. (2000), Schüssler and Axhausen (2009), Sun et al. (2014), Zhu and Levinson (2012)). Shortest paths with turn-penalty and stop lights-penalty have been also used, for instance by Bekhor et al. (2006) who evaluated frequent routes from the Massachusetts Institute of Technology faculty with a sample of 188 respondents, corresponding to 91 origin-destination pairs. Papinski et al. (2009) observed thirty-one drivers from Ontario, Canada, and performed a survey combined with GPS data collection to detect preferred routes. They found that respondents preferred routes were the ones that allowed to minimize time, the number of traffic lights or stop signs, avoid traffic congestion and maximize the directness of the route.

Papinski and Scott (2011) give a resume of the route choice attributes used in the literature and propose a GIS-based toolkit for route choice analysis which consider up to 40 variables. It is worth noting that the 40 variables are obtained as attributes of the shortest distance and shortest time paths. Using a sample of 237 observed routes and a pair test-t the authors compared observed routes to the optimal alternatives and found that drivers choose paths that are significantly longer if compared to the optimal paths in terms of distance and travel time. Zhu and Levinson (2012) demonstrate using GPS coordinates that drivers do not select shortest paths but select routes basing on travel time. To compute shortest time paths they estimate average travel time on the road network. New methodologies to estimate average link travel time using probe vehicles have been proposed by Hellinga et al. (2008) and Jenelius and Koutsopoulos (2013). While other studies are focused on the analysis of correlation of arc travel speeds (see Bernard and Axhausen (2006)).

Trip variability has been studied by Huff and Hanson (1986) who used a contingency table to compare different attributes such as trip purposes, mode, trip distance and arrival time. Pas (1983) implemented cluster analysis to identify groups of similar daily travel activities patterns. Pas and Sundar (1995) found high day-to-day variability in trip frequency and this variability was almost the same for home-based and non home-based trips.

Temporal changes within individuals and the factors that influence change are the subject of longitudinal studies (see, e.g., Fitzmaurice et al. (2008)). In transportation studies, longitudinal analysis could be performed using repeated measures of GPS data that allow to identify the position of a the driver who performs the same activity during different times and days. Few studies focus on temporal variability of individuals' travel activities. For instance, Pas and Koppelman (1987) studied the day to day variability of trips during five days and found that interpersonal variability was higher for individuals who do not have to carry on daily activities. Hanson and Huff (1982) detected high degrees of repetition on daily travel activities and applied clustering techniques to classify homogeneous groups in terms of travel behaviour using multi-day travel data (see Hanson and Huff (1986)). Hanson and Huff (1988) studied the systematic components of day-to-day variability of travel behaviour and found that a seven-week observation period was not enough to capture systematic variability. Buliung et al. (2000) analysed spatial repetitive location choices during a week by type of activity and found differences in spatial variability and transport mode choice between week-day and week-ends. Thus they conclude that policies aimed to reduce week-days private travels would be not effective for the week-ends.

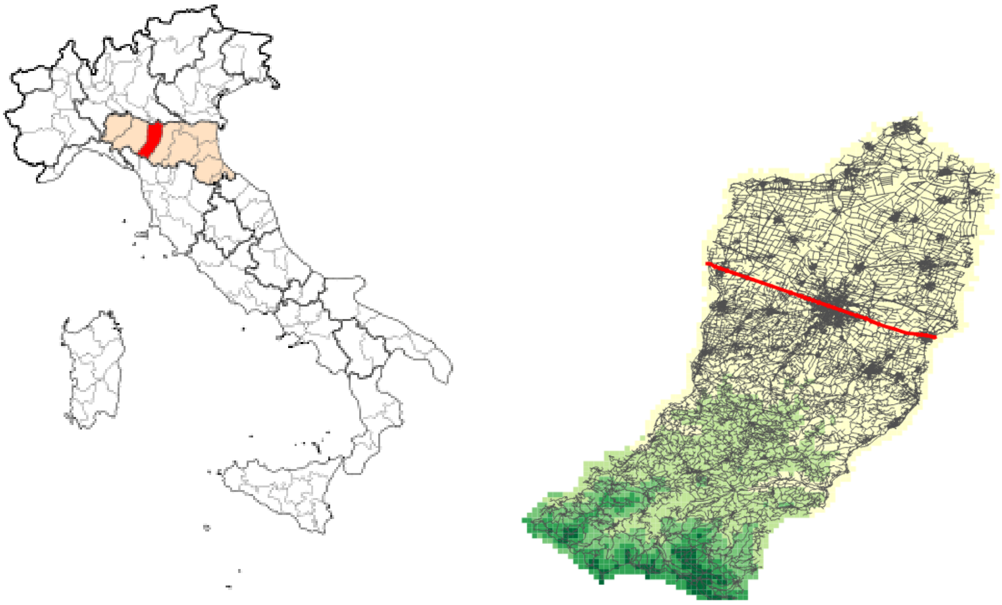
To analyse drivers' path choice variability, Jan et al. (2000) used the GPS coordinates of 216 drivers who performed 3000 trips in Lexington, Kentucky. They compared drivers taking identical trips and made considerations on the seasonality of the departure time and on the working and holiday periods to improve traffic assignment models. They conclude that trips made by the same driver were consistent over time and trips made by different drivers had some deviations.

Schonfelder and Axhausen (2003) used the Mobidrive survey (see Axhausen et al. (2002)) conducted in 1999 in two German cities during a period of 6 weeks on 361 persons. They found that the 70% of trips covered 2-4 locations and 90% of trips were made to the same 8 locations. They also detected that drivers choose between a small number of possible paths, that is individuals show well defined travel habits. Schlich and Axhausen (2003) analysed trip variability using travel diaries filled in during nine weeks and they evidenced a lack of good similarity measures of trips characteristics. Löchl and Schönfelder (2005) explored the potential of longitudinal travel data to provide a more complete overview of the interaction between the travel environment, the plan for the activities and their actual execution.

2.3 Data set and GIS tools

The data used for this analysis derives from the European project TeleFOT (Field Operational Tests of Aftermarket and Nomadic Devices in Vehicles). The objective of the project was to test the effects of nomadic devices on driving tasks. Two devices were used in the Italian pilot: a data-logger installed into the vehicle of 89 drivers and a nomadic device (smatphone) that provided to the driver a static navigation support system and a speed-limit/speed-alert information system. The data registered by the data-logger, that consist of low frequency GPS coordinates collected during a period of 17 months are considered. The GPS coordinates were stored every 200 seconds on average which corresponds to about 2 kilometers.

The drivers are a sample of residents in the Province of Reggio Emilia located in the Emilia Romagna Region of Italy (Figure 2.1), selected basing on their availability to participate in the test, on kilometers driven per year (minimum 10.000 kilometers), age, gender and profession and among drivers who had at least three years driving



(a) Province of Reggio Emilia

(b) Reggio Emilia territory

Figure 2.1: Province and Territory of reggio Emilia.

experience. The Province area can be inscribed in a rectangle with edges of 30 Km and 75 Km, respectively. The area is mainly plain except for the Southern part which is closer to the Apennines (green area in Figure 2.1b). The public transport network is not very well developed in terms of frequency, and private transport is largely used. The via Aemilia crosses the region from East to West and in the middle there is the main town, namely Reggio Emilia. (Figure 2.1b).

Although the entire data set included some trips performed anywhere in Italy, only the paths entirely included within the Provincial area were selected because the objective is to determine route choice variability of repetitive travels over time. Paths registered during the week-end (Saturday and Sunday) were excluded to simplify the identification of repetitive travels. Very short paths with less than 5 GPS coordinates have been also discarded, and the remaining paths have been checked for consistency. This filtering process resulted in the extraction of 41.960 paths performed during the observation period of one year and half (1st September 2009 - 31st January 2012) by 89 drivers. The total number of km travelled is 588.193. To the best of our knowledge this is one of the largest dataset used in literature to analyse driver behaviours.

The analysis was performed using the free and open source Geographic Information System QGIS 2.2, the object-relational database PostgreSQL 9.1 and its extension

PostGIS 2.0 that adds support for geographic objects. The GIS allows to manage large amount of spatial information and to visualize the variability of route choices in terms of selected arcs.

The shapefile of the Province of Reggio Emilia has been taken from the OpenStreetMap archive (www.openstreetmap.org) which is an open source project of geographic data created by a community of volunteers. The road network is represented in the map by polylines connected to an attribute table that includes information on the route id, the typology, the name of the road, the maximum speed on the arc and the indication of the road directions (one way or two ways). Since these data are provided by volunteers, the file had some missing data and the arcs where sometimes disconnected. Hence, a series of adjustments and congruence checks have been performed before using the map for the study.

2.4 Methodology

The proposed methodology is based on the deep use of GIS. In a first phase the map of the territory is enriched with additional data that can be derived from the observed paths. Than a few shortest path algorithms were applied using the map data, to determine optimal alternatives to the observed ones. Finally statistical tools were used to analyze the path variability and to try to explain the driver preferences.

2.4.1 Estimation of average travel speed

The huge amount of information in the dataset give the possibility to innovate upon the existing travel speed computation, by calculating road speeds during the test period. Indeed, several works in the literature compared observed routes with shortest time routes, but these optimal paths are computed using data which are independent from the test and not necessarily fit with the real speeds during the test period. The thesis proposes to use the observed data, which span almost all the territory and cover all the daily hours, to compute a very good approximation of the real travel speeds during the test period.

The methodology proposed aims to obtain the estimates of speed for each hour and arc of the road network using GPS coordinates from observed paths, as probe vehicles. It starts by computing the speed in correspondence of each GPS coordinate as the ratio between the incremental distance from the previous GPS point and the corresponding incremental time. Next a GIS iterative spatial query is run and each GPS coordinate is associated with all the arcs within 100 meters from the GPS point so that no map matching algorithm was needed. Here the assumption is that the same GPS coordinate can be associated to more than one arc so that closed arcs can have the same average velocity. For each link-hour pair the speed is computed as the average of the speeds from the associated GPS points. Successively, the arcs with missing data are associated with the estimated average velocities of the similar arcs, in terms of road type and time range, located within 500 meters. The same procedure has been run for each time range. An additional iteration was run for the arcs with missing velocities located within 1000 meters. A detailed explanation of the procedure to estimate average velocities using observed data is reported in Chapter 3. The distribution of velocities over the road network is represented in Figure 2.2

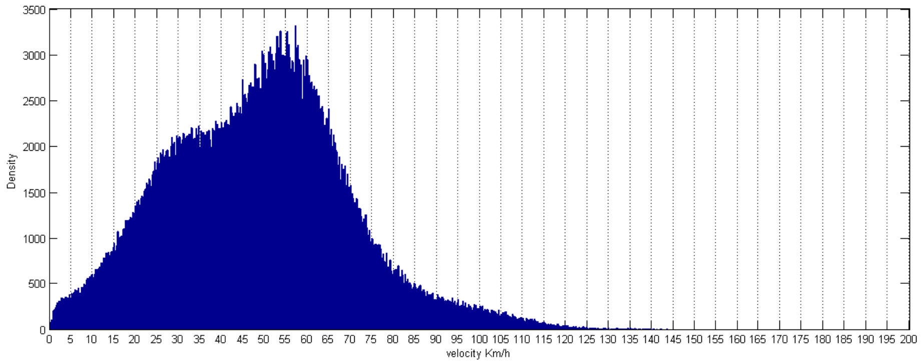
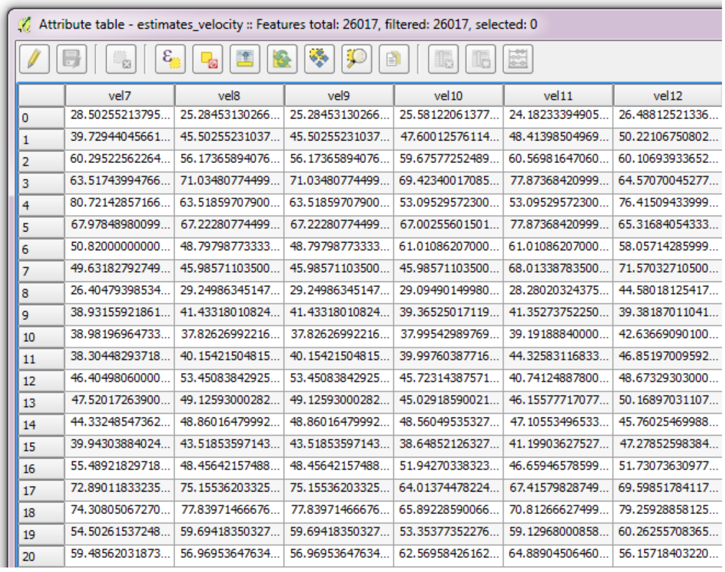


Figure 2.2: Distribution of velocities

The analysis of the GPS coordinates performed to estimate the arc speed has been useful also to manage missing data and errors. One of the arc's attribute in the OpenStreetMap network is the type of road. This attribute includes the *pedestrian* type, but some roads marked as pedestrian have been traversed by the users while driving. So, after the spatial query was run to associate the GPS coordinates to the road network, the road typology was converted from *pedestrian* to *road* for each arc where at least one GPS point were located. An attribute table of 24 additional columns was added to the shape file, each giving the average speed on a hourly range.

Figure 2.3 shows the attribute table with the estimated velocities. Each row of the table indicates a polyline of the road link, while the columns report the estimated average velocities in each hourly time range (i.e. vel8 is the time range 8-9 AM).



Attribute table - estimates_velocity :: Features total: 26017, filtered: 26017, selected: 0

	vel7	vel8	vel9	vel10	vel11	vel12
0	28.50255213795...	25.28453130266...	25.28453130266...	25.58122061377...	24.18233394905...	26.48812521336...
1	39.72944045661...	45.50255231037...	45.50255231037...	47.60012576114...	48.41398504969...	50.22106750802...
2	60.29522562264...	56.17365894076...	56.17365894076...	59.67577252489...	60.56981647060...	60.10693933652...
3	63.51743994766...	71.03480774499...	71.03480774499...	69.42340017085...	77.87368420999...	64.57070045277...
4	80.72142857166...	63.51859707900...	63.51859707900...	53.09529572300...	53.09529572300...	76.41509433999...
5	67.97848980099...	67.22280774499...	67.22280774499...	67.00255601501...	77.87368420999...	65.31684054333...
6	50.82000000000...	48.79798773333...	48.79798773333...	61.01086207000...	61.01086207000...	58.05714285999...
7	49.63182792749...	45.98571103500...	45.98571103500...	45.98571103500...	68.01338783500...	71.57032710500...
8	26.40479398534...	29.24986345147...	29.24986345147...	29.09490149980...	28.28020324375...	44.58018125417...
9	38.93155921861...	41.43318010824...	41.43318010824...	39.36525017119...	41.35273752250...	39.38187011041...
10	38.98196964733...	37.82626992216...	37.82626992216...	37.99542989769...	39.19188840000...	42.63669090100...
11	38.30448293718...	40.15421504815...	40.15421504815...	39.99760387716...	44.32583116833...	46.85197009592...
12	46.40498060000...	53.45083842925...	53.45083842925...	45.72314387571...	40.74124887800...	48.67329303000...
13	47.52017263900...	49.12593000282...	49.12593000282...	45.02918590021...	46.15577717077...	50.16897031107...
14	44.33248547362...	48.86016479992...	48.86016479992...	48.56049535327...	47.10553496533...	45.76025469988...
15	39.94303884024...	43.51853597143...	43.51853597143...	38.64852126327...	41.19903627527...	47.27852598384...
16	55.48921829718...	48.45642157488...	48.45642157488...	51.94270338323...	46.65946578599...	51.73073630977...
17	72.89011833235...	75.15536203325...	75.15536203325...	64.01374478224...	67.41579828749...	69.59851784117...
18	74.30805067270...	77.83971466676...	77.83971466676...	65.89228590066...	70.81266627499...	79.25928858125...
19	54.50261537248...	59.69418350327...	59.69418350327...	53.35377352276...	59.12968000858...	60.26255708365...
20	59.48562031873...	56.96953647634...	56.96953647634...	62.56958426162...	64.88904506460...	56.15718403220...

Figure 2.3: Estimated velocities - road link and hourly time range

2.4.2 Computing optimal alternatives

The analysis of route choices of the drivers is performed by comparing the observed routes to their optimal alternatives. The measures that are considered in the computation of the optimal paths are: distance, travel time, number of traffic lights and turns.

To compute the optimal alternatives, an algorithm is developed that takes as inputs the GIS map (in shapefile format) and the dataset with the GPS coordinates corresponding to the observed paths, and produces four shapefiles for each observed paths, representing the optimal alternatives with respect to the four measures.

To define the origin and destination of the optimal alternatives for each observed path, the algorithm associates the first and last GPS point of a route to the closer point on an entity in the map, that is, an arc or a vertex. Note that this point do not necessarily coincides with the initial or ending vertex of an arc, but can be in the middle of the arc. Than the algorithm creates a graph from the shapefile and applies an optimization algorithm for each measure. An inverse mapping from the graph to the shapefiles is used to identify the optimal path on the GIS map.

To compute the optimal paths the A^* shortest path algorithm is implemented, see Hart et al. (1968), which is a speed up of the Dijkstra algorithm (Dijkstra (1959)). A minimum distance path is the alternative route that the driver would have to choose to reach the destination traveling for the minimum number of kilometres.

Table 2.1: Turn Penalties

Angle ranges (°)			Penalties (sec.)
0.0	$\leq \alpha <$	22.5	0.0
22.5	$\leq \alpha <$	45.0	3.0
45.0	$\leq \alpha <$	67.5	5.0
67.5	$\leq \alpha <$	90.0	7.0
90.0	$\leq \alpha <$	112.5	10.0
112.5	$\leq \alpha <$	135.0	13.0
135.0	$\leq \alpha$		15.0

The minimum time path is the fastest route to reach the destination starting from the origin at the specific time and using the estimated average speed for the corresponding time range.

The third set of optimal alternatives allow to avoid as much as possible the stops at traffic lights. To compute these paths, an additional layer of the GIS map has been created which includes information on the location of traffic lights. These information have been provided by the Municipality of Reggio Emilia in jpg format and transferred by hand to a shapefile. While building the graph, the algorithm adds to the vertices the information on the possible existence of traffic lights and assign to the vertex a penalty. In absence of a statistical study about the time spent at traffic lights in the Province of Reggio Emilia, this value was set to about 60 seconds. This constraint is introduced only if the objective is to minimize the travel time. The measure that is minimized is the total sum of the travelling times along the arcs plus the time penalties on the vertices.

The fourth alternative consists in the selection of the path with the minimum number of turns along the route. The aim is to find the straightest path to the destination. This is obtained with the assignment of a time penalty to the trajectory and by solving a shortest path problem. Specifically, if the driver's trajectory is straight, no time penalty is applied. If the choice of the next arc implies the driver to turn an angle intersection, time penalties are applied. Penalties are set based on an empirical investigation with a panel of drivers. Table 2.1 shows the penalties applied to different rotation angles: no penalty is given to angles between 0 and 22.5 degrees and the maximum penalty is given to angles grater than 135.0 degrees.

A fundamental difference between the computation of the optimal path with traffic lights and the computation of the optimal path with turns has been found. Indeed the traffic light penalty is independent of the way a vertex is reached, so it can be simply summed to each entering (or leaving) arc of a vertex. Instead, the turn penalty directly depends from the pair of arcs used to enter and exit from a vertex. In this case the Dijkstra algorithm is no longer valid (simple examples can prove that it may fail to find the optimal solution). The problem has been studied by Volker (2008). The author proposes an offline method which consists in transforming the original graph in an edge-based graph, and an online method based on a modification to the Dijkstra method obtained by using additional vertex labels, and an hybrid methods. Since there were not strict time constraints for running the algorithm, the offline method was run which is the simplest to be implemented. In few words it consists in building a graph where each original link is associated to a node. Two nodes are linked by an arc if the corresponding original links share a vertex. The cost of the arc is the sum of the cost of the first link plus the cost of the turn (see Volker (2008) for more details).

2.4.3 Indicators and comparison methodology

To analyse the paths characteristics global indicators which have been frequently used in the literature are adopted. These indicators are:

(i) the route length,

(ii) the total travel time,

(iii) the Route Directness Index (RDI),

(iv) the average, maximum and minimum speeds on a link of the path. The indicators of velocity of the optimal alternatives have been computed by selecting the arc with the minimum and maximum velocity and by computing the average velocities of all the selected arcs.

The RDI has been used by Mortenson et al. (2011) who studied seven connectivity indicators and found that the RDI was the most robust to measure the connectivity. The RDI is computed as the ratio between the route length, and the Euclidean distance of the origin-destination points. Values close to one indicate that the route is almost a straight path, while higher values indicates that the route follows a path away from the straight one.

These indicators can be calculated independently for each observed or optimal path. Initially, observed and optimal routes are compared using their global indicators. Descriptive statistics is used to derive preliminary information on the driver behaviours. In a second step the influence of the area in which the origins and destinations are located is considered, by distinguishing GPS coordinates in the flat or in the mountains.

To further improve the analysis a clustering method is run to separate the routes in groups sharing similar missions, i.e., having close origins and close destinations. The results of this analysis, performed on a large dataset, can give very important information on the main drivers choices and motivations, which could be very valuable for local authorities that have to design urban and infrastructural plans, so as for road and traffic authorities.

2.5 Results

This section reports on the outcomes of the analysis performed on the large dataset.

2.5.1 Global indicators for the observed paths and their optimal alternatives

The first analysis concerns the distribution of trips during the test period. Figure 2.4 shows the number of departures for each day of the period under observation, by excluding Saturdays and Sundays to select more regular travels. The Figure 2.4 shows negative peaks in correspondence to National holidays, Christmas and the summer break.

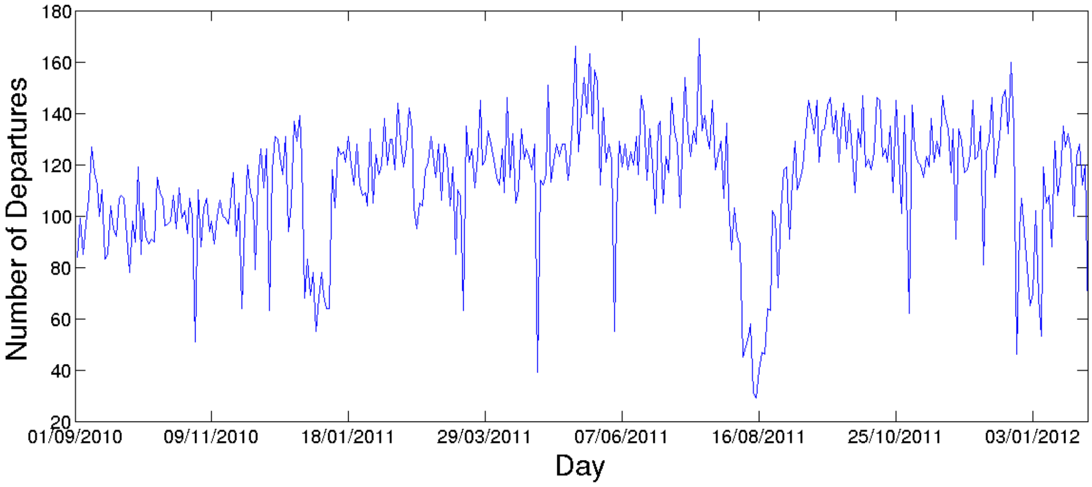


Figure 2.4: Total number of observed paths (1/09/2010 - 31/12/2012)

Next a table for each of the four indicators of paths characteristics is provided. The rows report separately the values of the indicators for the observed paths and for the four optimal alternatives. The descriptive distance indicator (see Table 2.2) shows that observed routes are constantly longer than their optimal alternatives, independently of the measure that has been optimized.

Table 2.2: Indicators of the length of paths (Km)

	Min	Average (± std.)	Max
Observed path	2.336	14.018 ± 6.773	50.191
Shortest distance path	1.409	11.811 ± 6.328	45.876
Shortest time path	1.409	12.526 ± 6.859	48.505
Shortest path with turns penalties	1.409	12.684 ± 6.884	50.038
Shortest path with traffic lights penalties	1.409	12.695 ± 6.792	48.505

Table 2.3 shows the average travel time for all paths of the sample. Also in this case the observed routes are consistently slower than all the optimal alternatives, independently of the measure that has been optimized. Indeed, the observed paths are, on average, 3-4 minutes longer compared to the optimal paths. These four minutes of difference are particularly significant for the minimum time route, but in this case there is probably a bias due to some setup and finishing operation that the driver makes to start and terminate his/her driving.

Table 2.3: Indicators of travel time (h:mm:ss)

	Min	Average (± std.)	Max
Observed path	0:05:43	0:19:03 ± 0:7:38	1:03:51
Shortest distance path	0:01:46	0:16:03 ± 0:7:30	0:56:03
Shortest time path	0:01:46	0:15:18 ± 0:7:04	0:53:13
Shortest path with turns penalties	0:01:46	0:15:34 ± 0:7:08	0:53:53
Shortest path with traffic lights penalties	0:01:46	0:15:28 ± 0:7:03	0:53:42

An evidence of the different types of routes that are selected by the drivers compared to the optimal alternatives is also provided by an indicator of the circuicity of the path defined as the distance traveled by a vehicle divided by the Euclidean distance computed between the start and the end of the path (see Axhausen et al. (2004)). In this case the geodesic distance is computed. Table 2.4 shows that the observed paths are on average more circuitous comparing their optimal alternatives.

Finally, Table 2.5 reports on the speed indicator.

Table 2.4: Indicators of RDI

	Min	Average (± std.)	Max
Observed path	1.0	1.6 ± 0.5	5.0
Shortest distance path	1.0	1.2 ± 0.1	3.9
Shortest time path	1.0	1.3 ± 0.1	3.9
Shortest path with turns penalties	1.0	1.3 ± 0.2	4.1
Shortest path with traffic lights penalties	1.0	1.3 ± 0.2	3.9

Table 2.5: Indicators of Velocity (Km/h)

	Min	Average	Max
Observed path	10.0	50.9	139.8
Shortest distance path	10.0	44.0	132.0
Shortest time path	11.0	47.6	132.0
Shortest path with turns penalties	11.0	47.4	129.0
Shortest path with traffic lights penalties	11.0	48.0	137.0

The average and maximum velocities are higher for observed paths comparing to their optimal alternatives, so it is possible to conclude that drivers, on average, select routes that are longer, more circuitous and for which the travel time is higher, but that allow for higher speeds.

2.5.2 Route choices analysis in macro regions

In this paragraph, the relationship between route choice behaviour and the geographical characteristics of the Province is analysed by the comparison between observed and optimal alternatives. To this aim the Province is divided in two main areas the plain and the mountainous areas (Figure 2.5). The plain area is further divided in two smaller areas separated by the via Aemilia (Figure 2.6). Route choices are analysed based on the location of their origin and destination to determine if the territorial characteristics influence route selection.

To group paths based on their spatial location, a spatial query is run on the start and end coordinates of the paths over the two identified areas of the Province. The analysis is performed on three groups of paths called *Plain*, *Mount* and *Plain-Mount*. The different patterns of drivers route choices are visualized as groups of gaps defined as the difference between the length of the observed path and the length of the optimal paths.

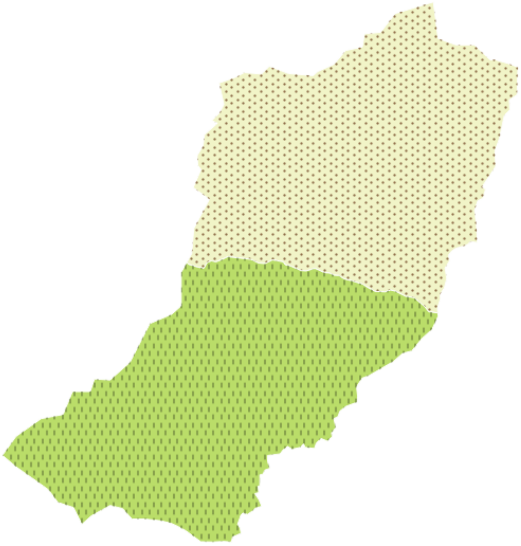


Figure 2.5: Mount and Plain areas

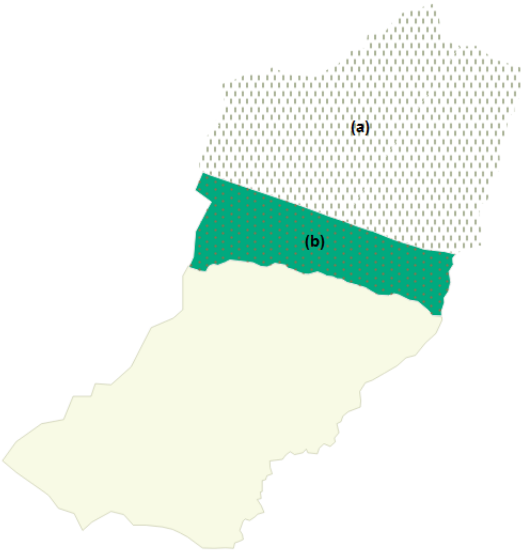


Figure 2.6: Plain area divided by the via Aemilia

The second analysis is carried out on paths located in the two flat areas of the Province separated by the via Aemilia, called *Plain(a)*, *Plain(b)* and *Plain (ab)*. The different patterns of drivers route choices are visualized as groups of gaps defined as the difference between the length of the observed path and the length of the optimal paths.

In Figure 2.7 each box displays an optimal alternative (minimum distance, minimum time, turns and traffic lights penalties) and the location of the origin and the destination. The bottom and top of the boxes refer to the first and third quartiles, the band inside the box gives the second quartile (the median), while the ends of the whiskers represent the maximum and the minimum values of the gap.

The boxes of the paths that start and end in the plain areas are taller comparing to the other two groups of paths (see x-axes) meaning that, in terms of route selection, there is higher variability of route choice. In the mountainous area and for the paths that start in the plain area and end in the mountains and viceversa, there is low degree of variability of gaps and the median is close the zero axis. Moreover, almost all boxes are divided by a line, the median, which is positioned closed to the bottom of the box meaning that the drivers have similar behaviour at certain parts of the scale but not in the remaining part in which route choices are more variable so that different routes are, for instance, selected with lower probability. The paths with gaps equal to zero are not considered in this analysis.

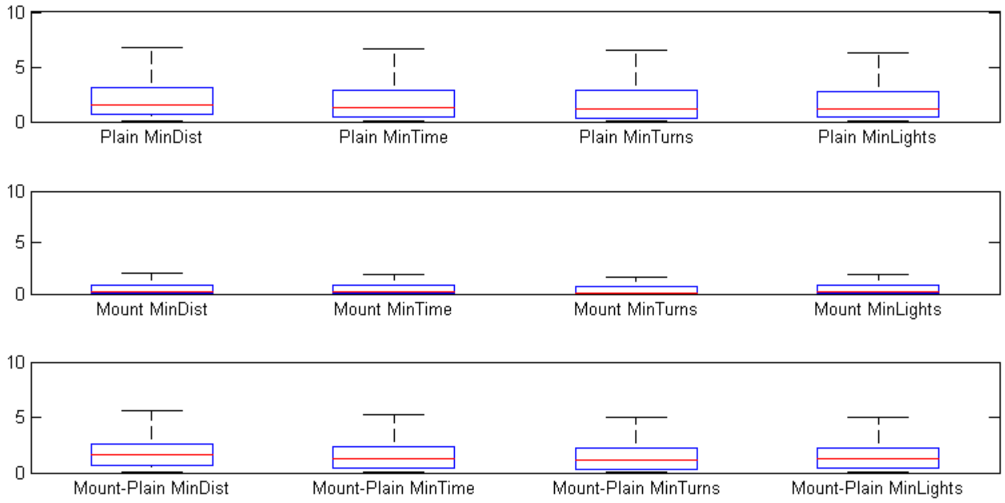


Figure 2.7: Boxplots of gaps (Mountainous and Plain areas)

All groups can be modelled as a Generalized Pareto distribution which is often used to model exceeding values over a thresholdGrimshaw (1993) and its parameters can be estimated using the maximum likelihood. Figure 2.9 shows that the gaps of paths with origin and destination within the plain areas have higher variability and the exceeding values are lower comparing to the other two groups of paths shown in Figure 2.10 and Figure 2.11. However it has to be noticed that fewer paths belong to this group (865) compared to the paths included in the other two groups of paths, the *Plain*(about 35.000) and the *Mount-Plain* (about 6.000).

This means that drivers who travel in the mountains select routes that are more similar to their optimal alternatives, while drivers who travel in plain areas have more choices so that they do evaluate several options that are in general different from the minimum distance paths.

Overall, route choice variability is lower in the mountains while more heterogeneity is found in the plain areas. No difference in route choice variability is found between the two different plain areas of the Province (a and b of Figure 2.6) that are characterised by similar distributions of the gaps.

2.5.3 Clustering of repetitive travels

This subsection aims to describe regular patterns of drivers' route choices in the context of repetitive activities. To identify similar travels in terms of spatial location of the origins and the destinations, an agglomerative hierarchical cluster

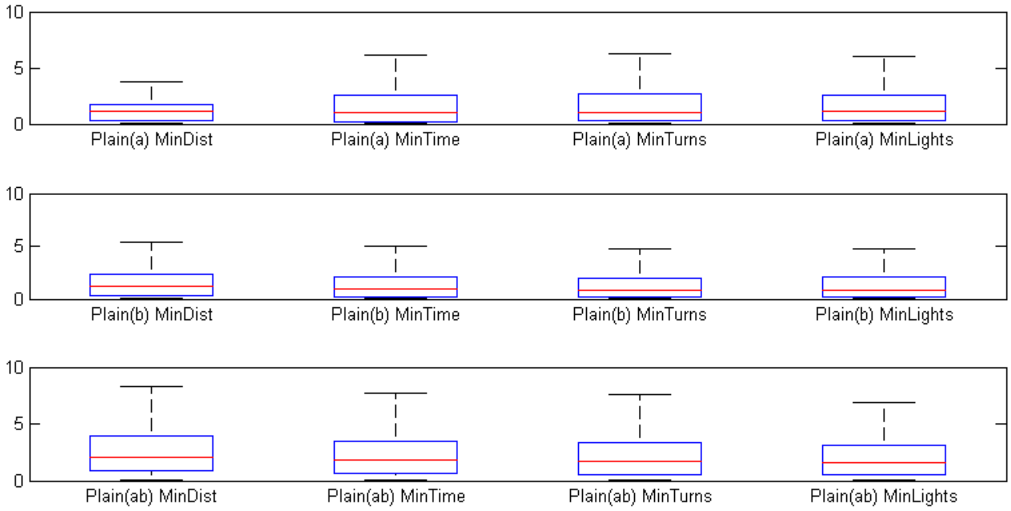


Figure 2.8: Boxplots of gaps

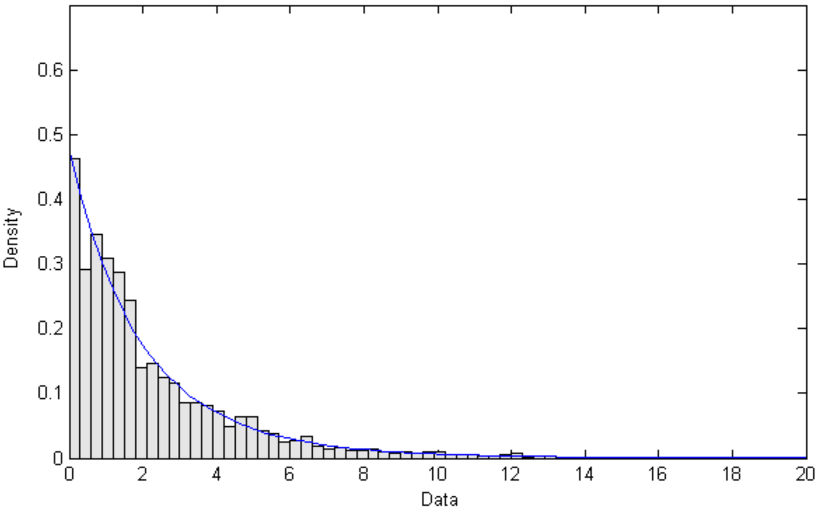


Figure 2.9: Generalized Pareto distribution of paths with origin and destination in the plain area - Min Distance

analysis (Ward (1963)) is run on all the selected paths using Matlab2014a. The Euclidean distance is used to define the different degrees of similarity among couple of coordinates of origin and destination. The measures of similarity are synthesized into two squared matrices with one row and one column for each path in which each element of the matrix indicates the Euclidean distance between a couple of path origins (resp. path destinations). To simultaneously take into account the origin and the destination, a third matrix is obtained by summing up the previous matrices. The single Linkage method, Sneath (1957), which is also called *nearest neighbour clustering*, consists in the creation of an initial number of clusters formed

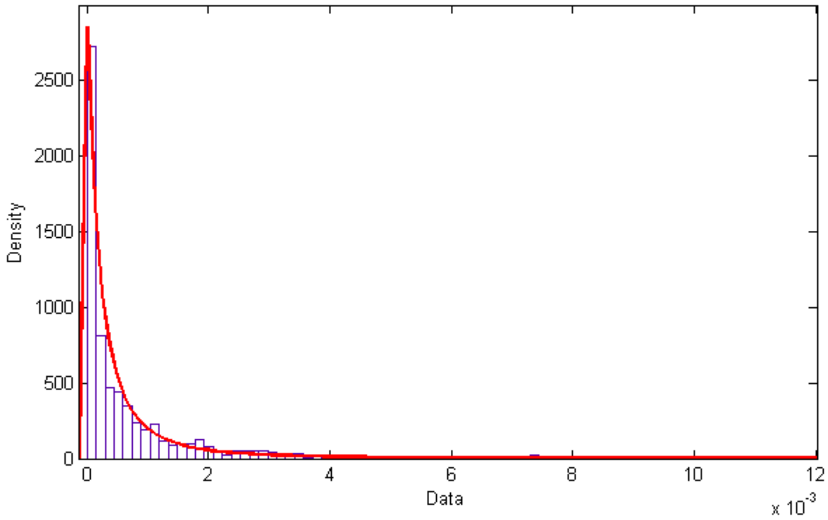


Figure 2.10: Generalized Pareto distribution of paths with origin and destination in the mountainous area - Min Distance

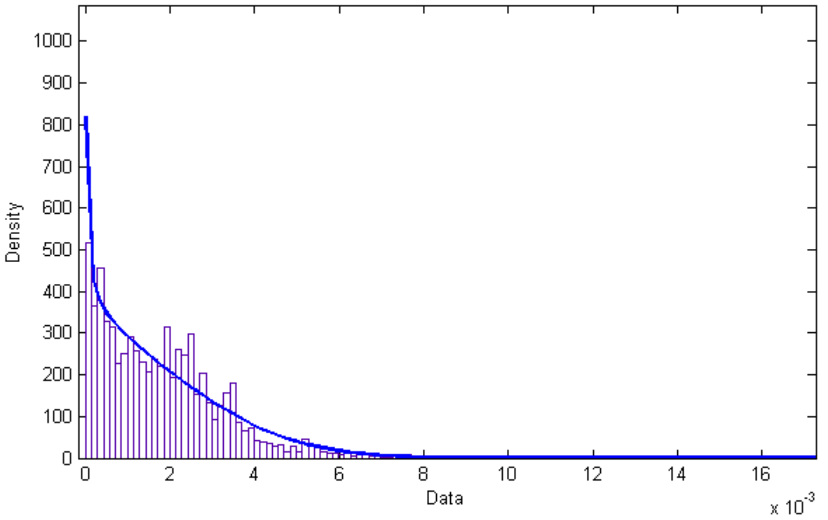


Figure 2.11: Generalized Pareto distribution of paths with origin in the mountainous area and destination in the plain area - Min Distance

by the number of elements to be grouped. Successively, clusters are merged based on their similarity. For instance, if two clusters are both formed by one path and these paths are similar in terms of the position of the origin and the destination, these two cluster are merged into one. The procedure is iterative and the algorithm stops when a maximum number of clusters is reached. Specifically, a constraint is added such that, for each driver, the cluster of paths can not exceed one third of the total number of elements to be clustered. The result of the cluster analysis are 14.464 clusters each including a different number of elements. The 77% of these clusters are formed by one element so that no repetitive travel exist. To have significant data all clusters with less than 30 paths are disregarded. The result of the selection are the 126 homogeneous clusters with an average of 108 paths for cluster, a maximum of 359 and a minimum of 32.

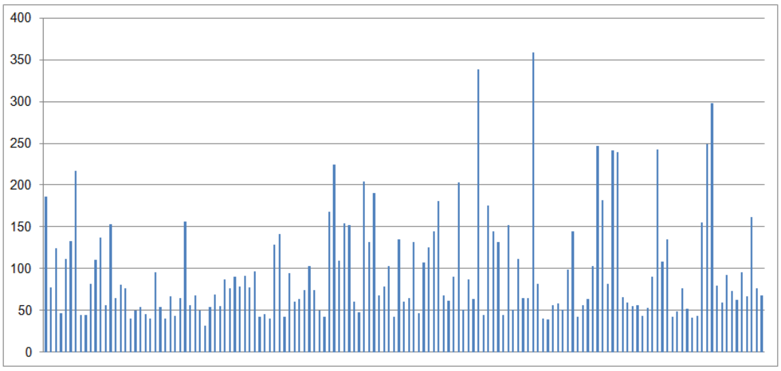


Figure 2.12: Number of paths in each cluster

Figure 2.12 shows the total number of paths included in each cluster. The number of clusters is about one third the total number of paths performed by each driver.

2.5.4 Geographical Route Directness Index

In this subsection a new indicator of the directness of the paths is proposed with the aim to quantify the deviation of the observed path from its optimal alternative. The indicator is called Geographical Route Directness Index (GRDI) and it is defined as the measure of the standardised distance between the observed and the optimal paths' length.

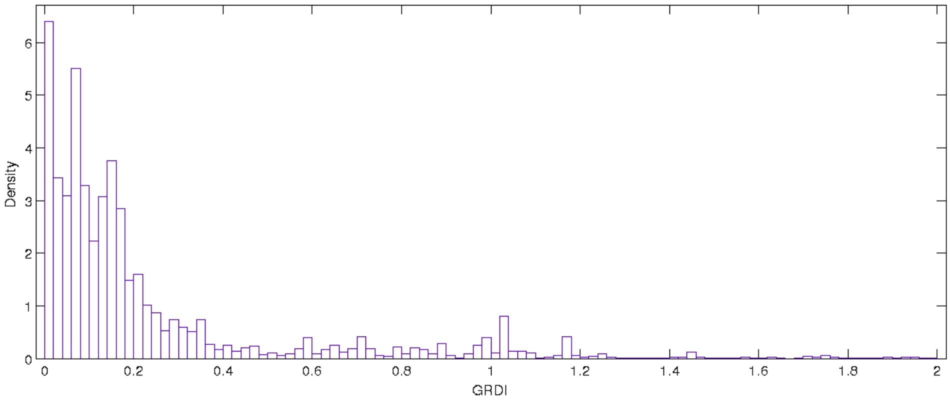


Figure 2.13: Distribution of the Geographical Route Directness Index

The index provides a measure of the connectivity of the path in relation to the optimal alternative in terms of minimum distance. The minimum value of the indicator is 0 when the observed and the optimal paths have the same length and the same origin and destination. The maximum value of the index is shown in Figure 2.13 that reports the distribution of the Geographical Route Directness Index for all groups of repetitive travels. The Figure shows that there are only few paths that have a length which is twice the length of the optimal path. These paths are only the 1.1% so that they can be considered as outliers and are removed from the analysis.

2.5.5 Analysis of variance of repetitive travels

This section aims to determine the relationship among different groups of repetitive travels and route choice behaviour. To this aim a one-way analysis of variance of the GRDI is performed to analyse the difference on the averages among different groups of repetitive travels. The analysis of variance is a statistical technique that allows to compare the averages between three or more groups (Fisher (1921)).

Figure 2.14 shows the distribution of the GRDI within each group of paths and it evidences that, although the means of the majority of groups is included within the range of the index of 0 and 0.5 (y-axis), the distribution of the index has a different behaviour depending on the group membership. In the 14% of the cases the mean of the GRDI is greater than 0.5 and, also in this case, the distribution of the values of the index changes from group to group.

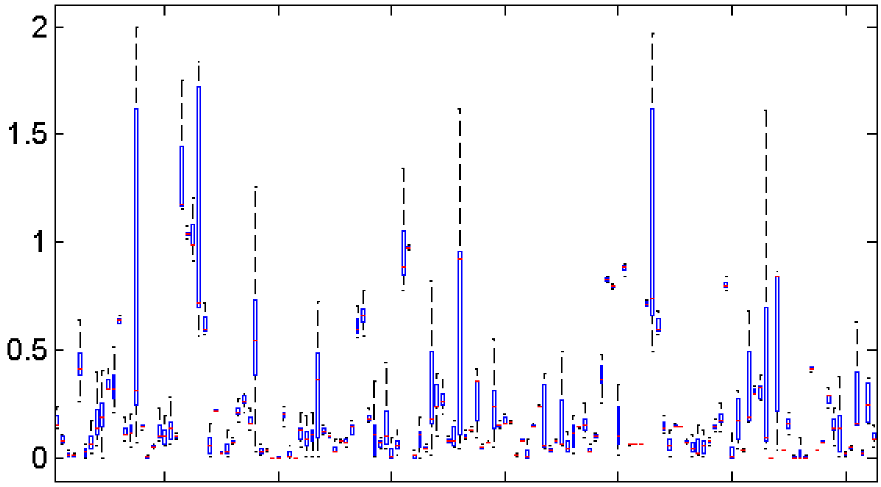


Figure 2.14: Boxplots of repetitive paths'clusters

The Figure 2.15 shows the results of the one-way analysis of variance performed on each group of repetitive travels (y-axis). The Figure evidences the average values of the GRDI by group (round circle) and graphically presents the confidence interval as a line that crosses the circle. The 60% of the tests have averages of the GRDI significantly different from at least one other group of paths, while the result of the remaining 40% couple comparisons are not significant. Moreover, the 15% of paths have GRDI means that are greater than 0.5, while the other groups of paths have values of the GRDI which are closer to zero. Each group of repetitive travels have therefore different GRDI averages that probably meaning that route choices are most likely influenced by individual habits or characteristics, or by other unobservable variables.

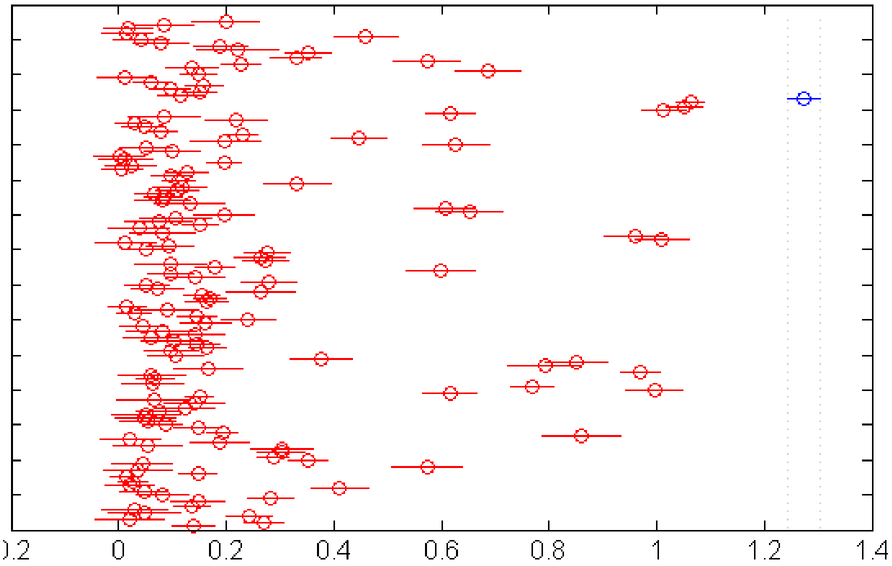


Figure 2.15: Multicomparison among the GRDI means of the clusters of repetitive travels

Successively, the statistical test on the difference between the averages of the GRDI index is performed by grouping the paths by the geographical location of their origin and destination. The three geographical areas are the same as the ones described in section 2.5.2. The results of the test are shown in Table 2.6 in which a multicomparison of the three groups of paths is presented. The first row of the table shows the comparison among the group of paths that have both ODs located in the flat area (*Plain*) and the group of paths with the origin located in the plain area and the destination on the mountain or viceversa (*Mount-Plain*). The estimated mean difference of group *Plain* and *Mount-Plain* is -.0960, and a 95% confidence interval (CI) for this difference is [-0.1120, -0.0800]. The interval does not contain the 0 and the test is significant (p-value=0.00<0.05) so that the means of the two groups are different. The same result is obtained if comparing the groups *Mount-Plain* and *Mount* (third row). Finally, the second row of the table shows that the two groups of paths that have both ODs located either in the plain area (group named *Plain*) or in the mountains (group named *Mount*) have similar averages. Thus it is possible to conclude that travels made on similar territorial characteristics have similar average of the Geographical Route Directness Index.

Table 2.6: Results of the one-way analysis of variance by group of ODs

Group1	Group2	CI (Lower)	mean difference	CI (Upper)	p-value
Plain	*Mount-Plain*	-0.1120	-0.0960	-0.0800	0.00
Plain	*Mount*	-0.0082	0.0091	0.0263	0.4343
Mount-Plain	*Mount*	0.0906	0.1050	0.1195	0.00

Table 2.7 shows that the average GRDI of the group of travels performed during Monday and Thursday are significantly different. Moreover the comparison among all couples of day of the week shows that no group have means of the GRDI that are significantly different. Thus we may conclude that, at the beginning of the week, the average GRDI is higher comparing to the other days of the week (average GRDI on Monday=0.250), as soon as the drivers approaches the week-end, they start to select routes that are more similar to their minimum distance alternative (average GRDI on Thursday = 0.219). Finally, the index slightly increase on Friday.

Table 2.7: Results of the one-way analysis of variance by day of the week

Group1	Group2	CI (Lower)	difference	CI (Upper)	p-value
Wednesday	Thursday	-0.0034	0.02	0.0434	0.1341
Wednesday	Friday	-0.017	0.006	0.0291	0.953
Wednesday	Monday	-0.034	-0.011	0.0119	0.6858
Wednesday	Tuesday	-0.0209	0.0018	0.0245	0.9995
Thursday	Friday	-0.0377	-0.014	0.0098	0.4937
Thursday	**Monday**	**-0.0547**	**-0.031**	**-0.0074**	**0.0032**
Thursday	Tuesday	-0.0417	-0.0182	0.0052	0.2096
Friday	Monday	-0.0404	-0.0171	0.0063	0.268
Friday	Tuesday	-0.0273	-0.0043	0.0188	0.987
Monday	Tuesday	-0.0102	0.0128	0.0358	0.5506

Table 2.8 shows the results of the analysis of variance performed among couples of groups identified by the season during which they have been performed. The table indicates that the means of GRDI in Autumn and Spring are significantly different. Specifically, the mean of GRDI computed for the group of paths performed in Autumn is lower (average GRDI in Autumn=0.23) compared to the average GRDI computed on the paths performed during the Spring (average GRDI in Spring=0.25). Similarly to the results of the analysis of variance performed on paths grouped by the day of the week, there is a significant difference among drivers who

select routes at the beginning of the working year (Autumn) and route selection performed at the end of the working year (Spring).

Table 2.8: Results of the one-way analysis of variance by season

Group1	Group2	CI (Lower)	Mean difference	CI (Upper)	p-value
Summer	Autumn	-0.0194	0.00075	0.0209	0.9997
Summer	Winter	-0.027	-0.0056	0.0158	0.909
Summer	Spring	-0.0424	-0.0203	0.0017	0.0832
Autumn	Winter	-0.0244	-0.0063	0.0117	0.8048
Autumn	**Spring**	**-0.0399**	**-0.0211**	**-0.0023**	**0.0207**
Winter	Spring	-0.0349	-0.0148	0.0054	0.2346

Finally, the last Figure 2.16 shows the distribution of the GRDI during different time of the day. During peak-hours the boxes are shorter comparing to the other time of the day, showing fewer variability of route choices with averages that are closer to zero. In this case it is possible to conclude that, during peak hours, the drivers tend to select routes that are more similar to their optimal alternatives in terms of distance and that these routes are similar between each other. Moreover an increased heterogeneity can be observed during the non-peak hours.

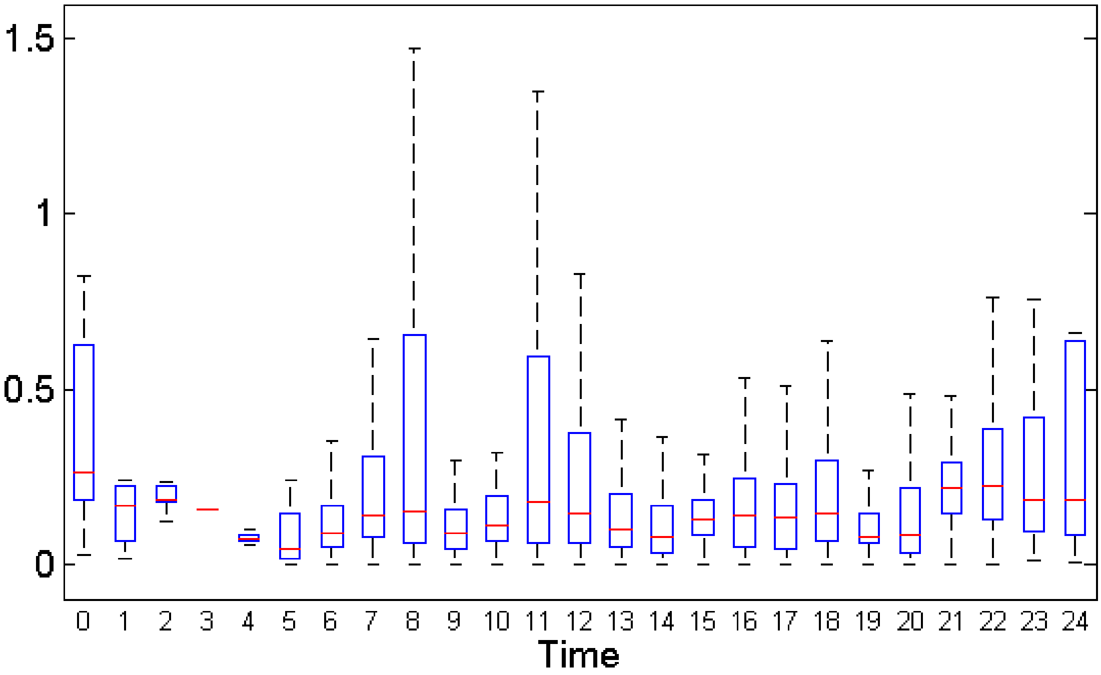


Figure 2.16: Average variability of the GRDI during different time of the day

Overall, the results of the tests performed on the different paths grouped by different categories show that route choice behaviour probably depends on the morphological characteristics of the territory. Moreover, the time of the day, the day of the week and the season may lead to different aggregate behaviours.

2.5.6 Analysis of route choices by the GRDI

Figure 2.17 shows the detail of the city center and it evidences the roads that are characterized by high averages of the GRDI. The first road is the ring, located on the North-West of the city center and it shows that when drivers need to cross the city center, they select longer but faster roads as explained in section 2.5. Specifically, when there is a faster alternative such as the one offered by the ring road, drivers select routes that have a length that is on average twice the shortest distance path. Therefore the connectivity between the West and the East areas of the central area of the Province is ensured by the a fast lane alternative that allows to avoid the traffic of the city center.

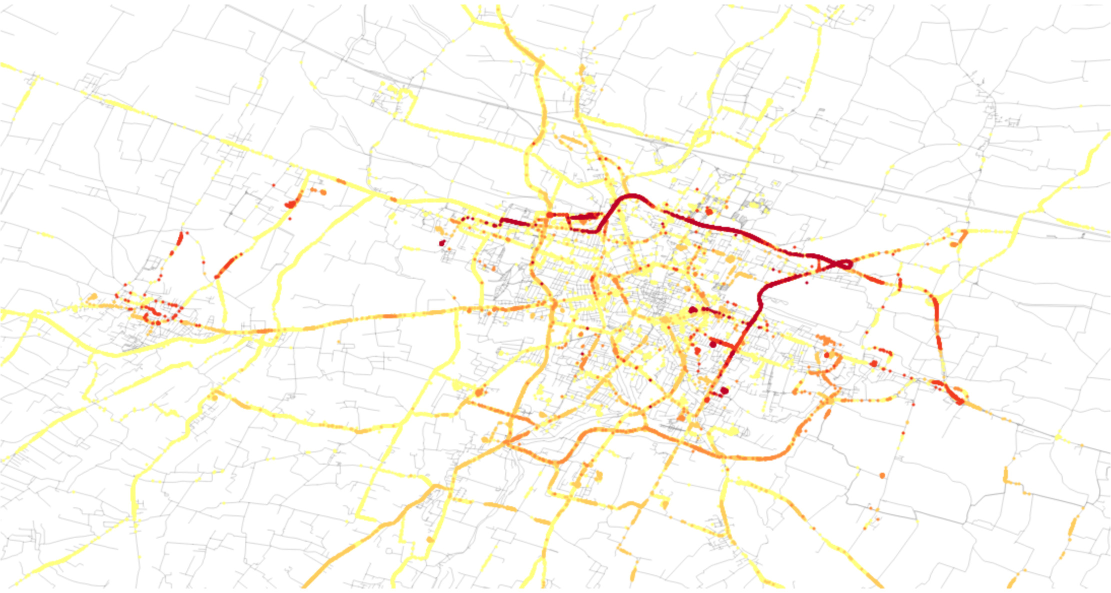


Figure 2.17: Mean of the GRDI - Minimum distance - City center

A second area that is characterized by high values of the GRDI is located near to Cavriago, a small town located in the West of Reggio Emilia. There, drivers who performs repetitive travels do not cross the center of the town which is the shortest in terms of km but they select the route that turns around it. Basing on these considerations, it is possible to conclude that habitual drivers who need to cross a dense populated areas or a town, prefer routes that allows to by-pass the city center.

To explain the high values of the GRDI in certain arcs of the road network, in particular along the ring road, the frequencies of selected alternatives of the groups of repetitive paths that have at least one GPS coordinate on the ring road are analysed. The groups of repetitive travels that have these characteristics are 25. Each group has an average of 105 paths, with a maximum of 338 and a minimum of 40. In the 64% of the groups the driver select in more of the 88% of the cases the route which is on average longer of 5 Km than the shortest distance alternative and that passes by the ring road. The other groups of repetitive travels are more heterogeneous in terms of route choices so that more alternative routes are selected (from 2 to 4).

Table 2.9: Selected clusters

Clusterid	Approx number of alternative paths	Frequency of selection of the bypass path	Avg Km of the observed paths using the ring	Avg Km of the optimal path	Additional Km traveled using the ring	Number of paths in the cluster
10057_6	2	1%	22	14	8.0	74
10031_4	3	1%	20	11	9.0	68
10015_36	3	3%	23	14	9.0	137
10108_25	2	4%	12	7	5.0	112
10130_136	3	4%	25	14	11.0	108
10125_22	2	5%	12	4	8.0	59
20014_6	2	6%	33	18	15.0	63
10059_2	3	32%	17	5.7	11.3	50
10130_23	3	35%	30	14	16.0	135
10024_232	2	88%	25	21	4.0	96
10101_2	2	88%	10	5	5.0	175
10055_2	2	96%	12	8	4.0	129
10073_2	1	99%	10	7	3.0	135
10094_22	2	99%	8	4.7	3.3	64
10024_81	2	99%	26	21	5.0	54
10094_17	2	99%	10	4.7	5.3	87
10073_34	1	100%	7	7	0.0	61
10053_18	1	100%	20	19	1.0	40
10101_41	1	100%	10	5.7	4.3	132
10094_6	1	100%	9.4	4.6	4.8	338
10108_34	1	100%	12	7	5.0	65
10040_37	1	100%	12	6	6.0	69
10055_36	1	100%	15	7.8	7.2	43
10091_6	1	100%	15	5.6	9.4	203
10101_4	3	100%	17	5.5	11.5	145

Table 2.9 reports the quantitative characteristics of each of the selected cluster. All clusters have at least one path that passes by the ring road. The average optimal length of each group ranges from 4 Km to 21 Km which evidences that the groups of repetitive travels are very different from each other in terms of origin

and destination. Specifically, some of the groups have the origins and destinations located near the city center while others do not. The common characteristics is that almost all groups include paths that, if the origin and the destination are connected with a straight line, this line crosses the city center of Reggio Emilia.

Finally an explanation of why the values of the GRDI are higher along the ring road comparing to the other road links is provided. Overall there is a strong negative correlation (-0.59%) between the number of times (the frequency) the driver decides to travel by the ring road and the additional number of Km this choice implies to travel. Furthermore, the drivers prefer a faster alternative (the by-pass) if this implies to travel on average 5 additional Kilometres. Traversing the ring road becomes inconvenient for the driver if this choice implies an average increase of 10 Km of the travel especially in case of short distance trips (about 5 Km). This behaviour explains why the value of the GRDI is on average higher on the ring road. That is the drivers who select the ring road as an alternative path, travel a number of km that are higher comparing the shortest distance path. Furthermore the index provides an indication of the road links which the drivers are available to select even though the choice implies a consistent deviation from the optimal alternative path.

2.6 Conclusions and Future Research Directions

This work analysed the spatial relationship existing between the location of the origin and destination and route choices using GPS coordinates that allow to represent the trajectories of paths made by the sample of drivers. The possibility to obtain information on the exact position of drivers during an extended period of time is offered by the increasing use of Intelligent Transport Systems that, in this study, consists of the installation of dataloggers into several vehicles of private citizens living in the Province of Reggio Emilia. The study analysed the variability and deviation of route choice from the optimal alternatives during a period of seventeen months.

With respect to behavioural findings, this research evidences a positive trend in the variation of route choices performed in different macro areas of the Province by highlighting the higher heterogeneity of route choices in the plain areas. This study has also shown that higher deviations from the optimal alternatives refer to paths that involve the selection of road links that allows for higher speeds thus suggesting that the road infrastructure does influence the choice of the route. The results of this study suggest that the newly introduced Geographical Route Directness

Index could be used to identify the road links where different assumptions should be taken into account in the implementation of traffic models and large dataset should be used to have more significant results.

From a planning and policy perspective the study of route choice behaviour over time and its relationship between infrastructure and paths characteristics may help to develop new strategies that are more dynamic and new transport solutions capable to meet travellers needs.

3 Assessing the consistency between clustered and modeled route choices

3.1 Introduction

In traffic engineering different assumptions on user behaviour are adopted in order to model the traffic flow propagation on the transport network. In this paper classical hypothesis on the user route choice are compared with real observations, pointing out the error related to using these approximations in real practice. If this problem is already well known in literature, (Abdel-Aty et al. (1997), Jan et al. (2000)) only few works are available, which provides quantitative and empirical analysis of the discrepancy between observed and modelled route choices. This is mainly related to the complexity of collecting data: to analyse route choice it is necessary to have observations for a large time period, since observing trajectories for the single user on a specific day could not be enough. Information is required for several days in order to analyse the repetitiveness and understand which elements influence this choice. This is mainly related to the complexity of collecting data: to analyse route choice it is necessary having observations for a large period of time, since observing trajectories for the single user on a specific day could not be enough. Information is required for several days in order to analyse the repetitiveness and understand which elements influence this choice.

3.2 Literature review

Several studies on route choices deal with the development of algorithms for the generation of the choice set. Given an origin and a destination, the problem consists in the identification of the sub-set of alternatives that are most likely to be considered by the drivers. These may vary during the day and across days, depending on habits, past experiences, information acquired before and during the trip, etc. Furthermore the alternatives to be identified by the algorithm have to be heterogeneous in order to properly represent the variety of available choices. The goodness of the solution is usually evaluated using the overlap percentage between the alternatives generated by the algorithm and the observed paths.

In Scott et al. (1997) it is presented an overview of different algorithms for the identification of the best alternatives (i.e. in terms of cost, distance, time, etc.). Two main categories of algorithms are identified: the link elimination (Azevedo et al. (1993)), and the k-best paths (van der Zijpp and Catalano (2005)). The first approach consists in the identification of the optimal paths, from which one or more links are eliminated. For instance, when a driver wants to avoid a specific road link, this is eliminated and the new alternative is computed. The second approach consists in the identification of the k-best paths. The disadvantage of this second method is that it is not very efficient and it tends to create similar alternatives. They also propose a constrained k-path algorithm to find the set of alternative routes that are most likely selected by the drivers, through identifying the number of alternatives, which are not overlapping or that are not circular. The method proposed by Akgun et al. (2000) consists in the measurement of the spatial dissimilarity of alternatives after the k-shortest paths are computed. The dissimilarity of the paths is obtained by randomly selecting the paths from the set such that the dissimilarity, in terms of overlapping, is maximized.

As first step of this work, a comparison between these results and information from previous studies is performed. Among the different works reported in the literature, Bekhor et al. (2006) found that the 37% of respondents selected the shortest time paths (90% of overlapping was required). Similarly, Prato and Bekhor (2006) found that the travellers who selected the shortest time path were only 43.3%. An empirical analysis of route choice behaviour is presented in Zhu and Levinson (2012). The authors study habitual driver behaviour using GPS coordinates registered during three weeks. Analyses from this database highlight how results from the deterministic route choice models do not match with the observed paths. Another study on the perception of travel time Vreeswijk et al. (2013), consistently with the previous analyses, found that 41% of drivers minimize time, while 80% of drivers minimize distance. They realized also that having a more direct connection or a faster route influences the perception of travel times. Finally, Parthasarathy et al. (2013) point out that the structure of the network influences the perceived travel time too. They used a dataset of GPS coordinates and a survey to find significant differences in travel time perception based on the characteristics of the road network. Results presented in this work differ from the ones reported in the literature because only the 26% of the paths selected by the drivers overlaps the shortest time alternative (at least 80% of overlapping) as discussed in the section that describes the results of the analysis.

This work differentiates from the others in the estimation of the mean speeds, used to compute the path travel time, which is different in each time interval and for each link of the road network. In the literature, the shortest path evaluation is usually based on the maximum speed for each road link. In this work, optimal time alternatives are evaluated based on observed data; the real shortest path is thus obtained for the specific time interval. Secondly, the GPS dataset is collected during a year and half (September 2010 – January 2012) thus allowing to have a real observation of the habitual driver behaviour over a significant amount of time. Hence it is possible to compare the characteristics from simulated alternatives with the observed ones. Our analyses are carried out in the morning peak period, in which congestion is observed. This allows to evaluate not only the difference between the observed and the shortest path, but also between the real subset of used routes and the modelled set at equilibrium. The organization of this chapter is as follows.

The contribution of the chapter is twofold: firstly, it is evidenced haow drivers do not select the shortest time path so that the probability of selecting the shortest path alternative is much lower. Hence the analysis of route reliability of the path travel time, is relevant to understand the behaviour of drivers and the possibility to have reliable results from the route choice models.

3.3 Description of the dataset

3.3.1 Low-frequency GPS coordinates

The dataset used in this study consists of low-frequency GPS coordinates and refers to paths performed by 89 drivers in the Province of Reggio Emilia, Italy, during the period 1st September 2010 – 31st January 2012. Data were collected using a data logger installed into the vehicle. An extract of the file with the observed data is shown in Table 3.1. The origin and destination of the observed paths have been identified using the panel id. A panel id = 2 indicates the ignition of the engine, a panel id = 0 shows that the engine has been turned off and a panel id = 1 shows that the vehicle is turned on. The *deltapos* and *deltatime* indicate, respectively, the incremental time (seconds) and distance (metres) from the previous GPS coordinate. Since these measures have been obtained thanks to a map matching algorithm implemented by the data provider, *deltapos* and *deltatime* are deployed to compute the velocity of the vehicle.

Table 3.1: Raw data

userid	date	time	lat	lon	deltapos	deltatime	idpanel	roadtype
ITFOT01P001	09-12-10	7:49:41	44654579	10762002	0	1442673	0	2
ITFOT01P001	09-12-10	7:56:42	44660682	10739462	2058	421	1	2
ITFOT01P001	09-12-10	7:58:21	44667855	10715771	2041	99	1	0
ITFOT01P001	09-12-10	8:00:25	44676864	10692059	2132	124	1	2
ITFOT01P001	09-12-10	8:01:49	44694834	10687143	2104	84	1	2
ITFOT01P001	09-12-10	8:03:13	44704200	10667055	2039	84	1	2
ITFOT01P001	09-12-10	8:04:36	44711408	10644036	2020	83	1	0
ITFOT01P001	09-12-10	8:06:18	44716269	10622375	2010	102	1	0
ITFOT01P001	09-12-10	8:09:53	44712125	10603574	1954	215	2	0
ITFOT01P001	09-12-10	12:47:35	44711889	10603224	0	16662	0	0
ITFOT01P001	09-12-10	12:52:17	44716127	10622541	2009	282	1	0
ITFOT01P001	09-12-10	12:53:51	44711173	10644557	2033	94	1	0
ITFOT01P001	09-12-10	12:55:09	44703337	10668823	2137	78	1	2
ITFOT01P001	09-12-10	12:56:28	44692451	10687030	2122	79	1	2
ITFOT01P001	09-12-10	12:57:52	44675740	10694178	2062	84	1	2

The database includes more than 57.000 observed paths distributed rather uniformly in the Provincial territory. Data were collected in the context of the European Community FP7 project TeleFOT, which aimed at testing the impact of in-vehicle and nomadic devices on usability, behaviour, incidents, safety, green driving, efficiency and the impact on the transport system. Systematic trips are detected using a hierarchical clustering approach, which run over the origin and destination pairs (see Chapter 2). The result consists of 119 clusters of repetitive trips for a total of 13.766 paths. Each cluster includes a set of paths made by the same driver during the observation period. The dataset allows to have an overview of the systematic choices made by the users in terms of route choice, travel time, day identification and departure time. Clusters of repetitive travels in terms of similar origin and destination are generated using the single linkage method Sibson (1973)). Similarities between origin and destination pairs are summarized in a matrix of distances measured through the Euclidean metric. Initially, each observation forms a cluster. Successively, step by step, the nearest observation pairs are merged into a new cluster. The cophenetic correlation coefficient, which is an indicator of the goodness of the clusters structure Jain and Dubes (1988) and varies between -1 and 1, shows values that are above 0.7.

3.3.2 Estimated velocities

To determine the optimal path in terms of time, the average link speed is estimated using observed data. First of all, the velocities computed using the observed data have been separated based on their timestamps. Observed GPS coordinates have been imported into Postgresql 9.3 with PostGIS 2.0 extension, in twenty-four different tables corresponding to the observed velocities in the 24 hourly time ranges. The GPS coordinates have been converted into PostGIS geometries. An example of string that corresponds to the GPS coordinates: 10.553625, 44.612527 is the following:

0101000000FA7E6ABC741B25406075E448674E4640

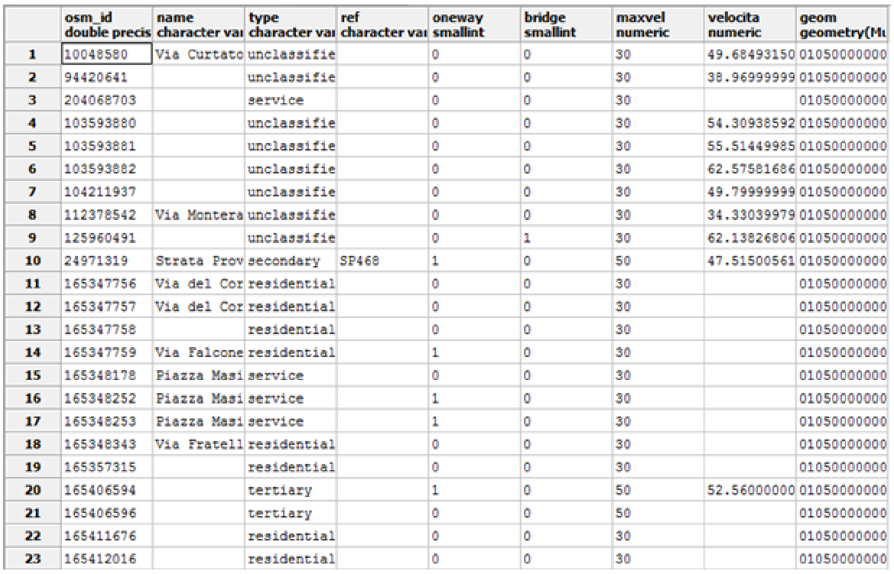
The Openstreetmap shape file has also been imported into the PostGIS database.

Listing 3.1: Estimated velocities from GPS data - Phase 1

```
SELECT
      roads.osm_id,
      roads.name,
      roads.type,
      roads.ref,
      roads.oneway,
      roads.bridge,
      average_velocity.maxvel,
      average_velocity.velocita,
      roads.geom

    FROM
        (SELECT
          match.osm_id,
          match.name,
          match.type,
          match.oneway,
          match.bridge,
          veltype.maxvel,
          AVG(match.vel) AS velocita,
          match.geom
        FROM
            (SELECT
               roads.osm_id,
               roads.name,
               roads.type,
               roads.ref,
               roads.oneway,
               roads.bridge,
               roads.geom,
               telefot8.vel
            FROM
              telefot8,
              roads
            WHERE
              ST_DWithin(telefot8.geom,roads.geom,0.001)) AS match
            LEFT JOIN veltype ON match.type=veltype.type
        GROUP BY
          match.osm_id, match.name, match.ref, match.type, match.oneway, match.
            bridge, veltype.maxvel, match.geom
        ORDER BY
          match.name, match.type) AS average_velocity
FULL OUTER JOIN roads ON average_velocity.osm_id=roads.osm_id;
```

From line 25 to 38, the query assigns the observed GPS coordinates of the time range 8-9 AM (telefot8) to the shape file of the road network (roads). In order to match the coordinates to the road network, line 38 associates all the coordinates to the arcs that are located within 100 metres from the coordinate. The distance is expressed in terms of degrees (0,001) because the geographical projection WGS84, EPSG: 4326 has been used. Line 39 assigns the maximum velocities to the road links with missing data based on the typology of road. Line 40 - 41 computes the average velocities on each arc using the GPS coordinates assigned to that arc. A table called *average velocity* is created as shown in Figure 3.1.



	osm_id double precis	name character vai	type character vai	ref character vai	oneway smallint	bridge smallint	maxvel numeric	velocita numeric	geom geometry(Mu
1	10048580	Via Curtato	unclassifie		0	0	30	49.68493150	01050000000
2	94420641		unclassifie		0	0	30	38.96999999	01050000000
3	204068703		service		0	0	30		01050000000
4	103593880		unclassifie		0	0	30	54.30938592	01050000000
5	103593881		unclassifie		0	0	30	55.51449985	01050000000
6	103593882		unclassifie		0	0	30	62.57581686	01050000000
7	104211937		unclassifie		0	0	30	49.79999999	01050000000
8	112378542	Via Montera	unclassifie		0	0	30	34.33039979	01050000000
9	125960491		unclassifie		0	1	30	62.13826806	01050000000
10	24971319	Strata Prov	secondary	SP468	1	0	50	47.51500561	01050000000
11	165347756	Via del Cor	residential		0	0	30		01050000000
12	165347757	Via del Cor	residential		0	0	30		01050000000
13	165347758		residential		0	0	30		01050000000
14	165347759	Via Falcone	residential		1	0	30		01050000000
15	165348178	Piazza Masi	service		0	0	30		01050000000
16	165348252	Piazza Masi	service		1	0	30		01050000000
17	165348253	Piazza Masi	service		1	0	30		01050000000
18	165348343	Via Fratell	residential		0	0	30		01050000000
19	165357315		residential		0	0	30		01050000000
20	165406594		tertiary		1	0	50	52.56000000	01050000000
21	165406596		tertiary		0	0	50		01050000000
22	165411676		residential		0	0	30		01050000000
23	165412016		residential		0	0	30		01050000000

Figure 3.1: Estimated velocities - Phase 1

A selection of the pedestrian roads have been converted to *road* when GPS coordinates were located over the link.

Listing 3.2: Correct road typology

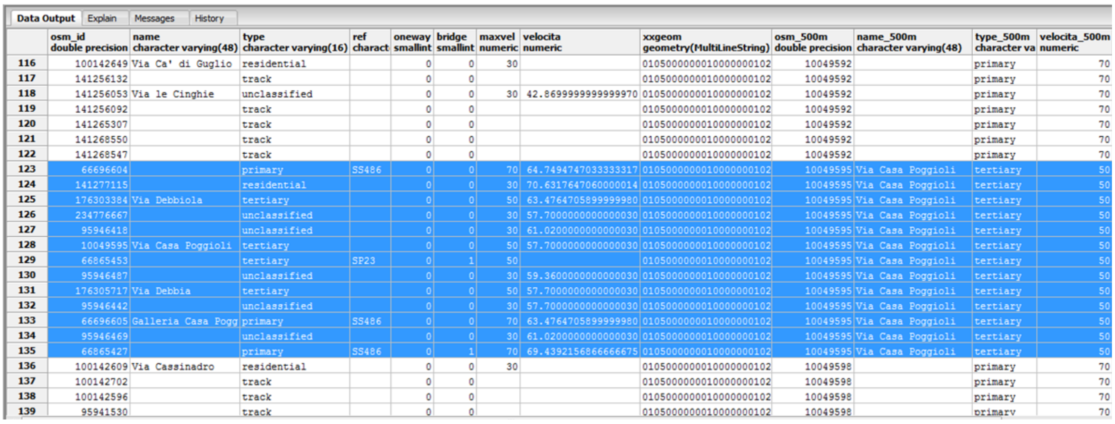
```
UPDATE average_velocity
SET type='road'
WHERE (average_velocity.type = 'path' OR average_velocity .type = 'track' OR
  average_velocity .type = 'footway' OR average_velocity.type = 'cycleway' OR XX.
  type = 'pedestrian' OR average_velocity.type = 'steps') AND average_velocity.
  velocita IS NOT NULL;
```

Successively, an iterative spatial query of the road links with estimated velocities was run over the road links without estimated data. The table that is obtained by running the query 3.3, associates each arc to its corresponding neighbouring arcs of the same network (within 500 metres).

Listing 3.3: Find all road links located within 500 meters

```
CREATE TABLE nearroad AS(
SELECT
    average_velocity.osm_id,
    average_velocity.name,
    ass.osm_id AS osm_500m,
    ass.name AS name_500m,
    average_velocity.type,
    ass.type AS type_500m,
    average_velocity.maxvel,
    ass.maxvel AS velocita_500m,
    average_velocity.velocita,
    average_velocity.geom
FROM
    average_velocity, average_velocity AS ass
WHERE
    ST_DWITHIN(average_velocity.geom,ass.geom,0.0045));
```

The output of the query is presented in Figure 3.2 where the table have been ordered by road links ids included within 500 metres (*osm500m*). The Figure shows that several estimated velocities (column *velocita)* are assigned to the same road link included within 500 metres. For instance, the road link called *Via Casa Poggioli* is located within 500 metres with respect to several links that are listed on the left side of the table (i.e. see *Via Debbia*, *Via Debbiola* in column *name.*



	osm_id double precision	name character varying(48)	type character varying(16)	ref charact	oneway smallint	bridge smallint	maxvel numeric	velocita numeric	xxgeom geometry(MultiLineString)	osm_500m double precision	name_500m character varying(48)	type_500m character va	velocita_500m numeric
116	100142649	Via Ca' di Guglio	residential		0	0	30		0105000000010000000102	10049592		primary	70
117	141256132		track		0	0			0105000000010000000102	10049592		primary	70
118	141256053	Via le Cinghie	unclassified		0	0	30	42.8699999999999970	0105000000010000000102	10049592		primary	70
119	141256092		track		0	0			0105000000010000000102	10049592		primary	70
120	141265307		track		0	0			0105000000010000000102	10049592		primary	70
121	141268550		track		0	0			0105000000010000000102	10049592		primary	70
122	141268547		track		0	0			0105000000010000000102	10049592		primary	70
123	66696604		primary	SS486	0	0	70	64.7494747033333317	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
124	141277115		residential		0	0	30	70.6317647060000014	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
125	176303384	Via Debbiola	tertiary		0	0	50	63.4764705899999980	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
126	234776667		unclassified		0	0	30	57.7000000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
127	95946418		unclassified		0	0	30	61.0200000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
128	10049595	Via Casa Poggioli	tertiary		0	0	50	57.7000000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
129	66865453		tertiary	SP23	0	1	50		0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
130	95946487		unclassified		0	0	30	59.3600000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
131	176305717	Via Debbia	tertiary		0	0	50	57.7000000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
132	95946442		unclassified		0	0	30	57.7000000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
133	66696605	Galleria Casa Pogg	primary	SS486	0	0	70	63.4764705899999980	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
134	95946469		unclassified		0	0	30	61.0200000000000030	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
135	66865427		primary	SS486	0	1	70	69.4392156866666675	0105000000010000000102	10049595	Via Casa Poggioli	tertiary	50
136	100142609	Via Cassinadro	residential		0	0	30		0105000000010000000102	10049598		primary	70
137	100142702		track		0	0			0105000000010000000102	10049598		primary	70
138	100142596		track		0	0			0105000000010000000102	10049598		primary	70
139	95941530		track		0	0			0105000000010000000102	10049598		primary	70

Figure 3.2: Estimated velocities - Phase 2

The last query, therefore, in the specific example of Figure 3.2, selects the road links of type *tertiary* from the column *type*, because *Via Casa Poggioli* is *tertiary* (see column *type500m*). And it assigns the average velocities selected from column *velocita* to the link *Via Casa Poggioli* as shown in query 3.4.

Listing 3.4: Iterative query to estimate velocities over missing links - Phase 2

```
SELECT
    osm_500m,
    name_500m,
    type_500m,
    (AVG(velocita)*(maxvel/velocita_500m)) AS speed,
    assgeom
FROM nearroad
WHERE type_500m = type
GROUP BY osm_500m,name_500m,type_500m,nearroad.maxvel,nearroad.velocita_500m,
  assgeom
```

A last query is required to estimate the last few arcs with missing velocities. These are the arcs that did not have any near by GPS coordinate. Figure 3.3 compares the missing arcs (left map in yellow) to the GPS coordinates that were available for the estimation of average velocities (map on the right).

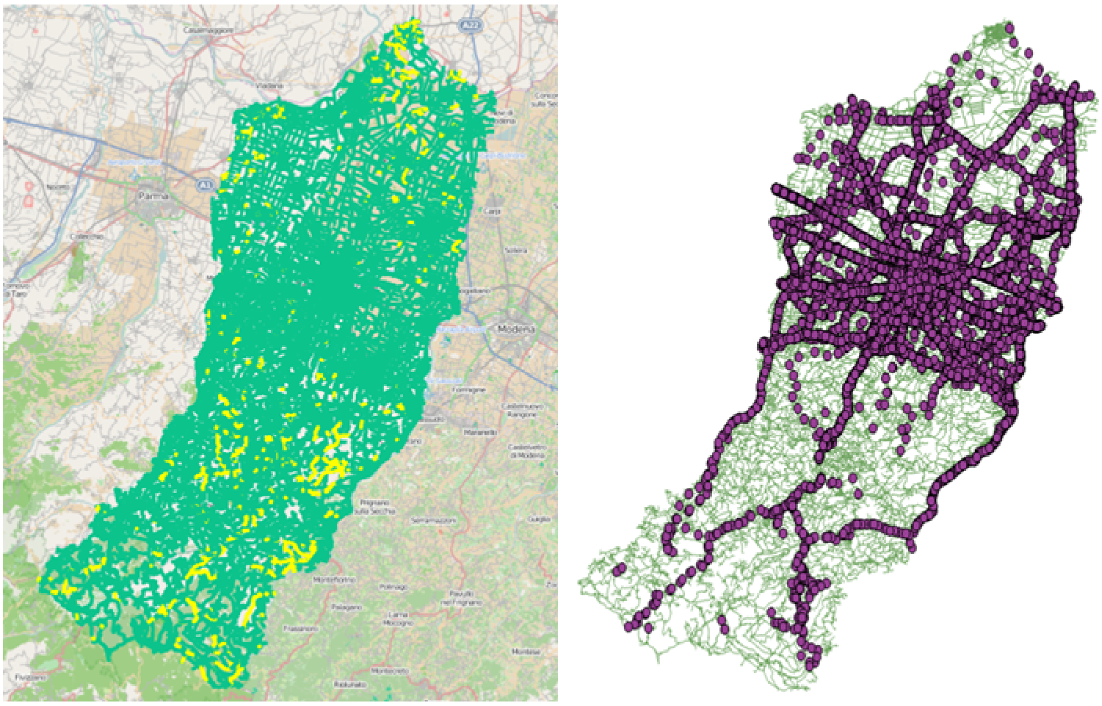
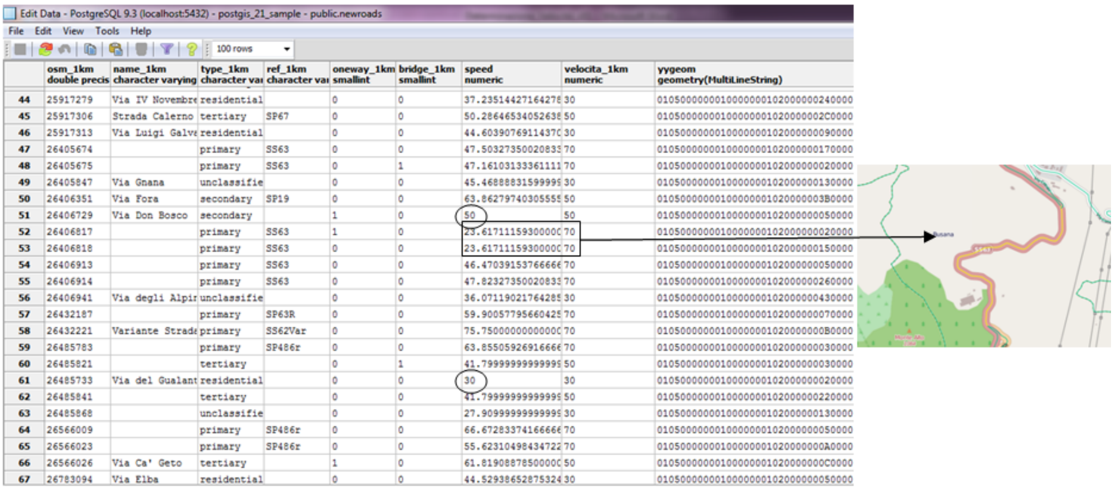


Figure 3.3: Estimated velocities - Phase 2

The procedure of Phase 2 is thus repeated (query 3.3 and 3.4), this time with a range of 1 Kilometers instead of 500 meters.

The result is an almost complete map of estimated average velocities. The procedures described above are repeated for all time ranges and merged. The result is a shape file associated with an attribute table with road links as rows and estimated average velocities as columns. Since there were still few remaining arcs, especially in the mountainous areas, without any estimation, the last Phase consisted in the imputation of missing data using nearest-neighbour method. The method replaces missing data with the corresponding value from the nearest-neighbor column using the Euclidean distance as metrics. If the corresponding value of the next near



Edit Data - PostgreSQL 9.3 (localhost:5432) - postgis_21_sample - public.newroads

	osm_1km double precis	name_1km character varying	type_1km character vai	ref_1km character vai	oneway_1km smallint	bridge_1km smallint	speed numeric	velocita_1km numeric	yygeom geometry(MultiLineString)
44	25917279	Via IV Novembre	residential		0	0	37.2351442716427$	30	0105000000010000000102000000240000
45	25917306	Strada Calerno	tertiary	SP67	0	0	50.2864653405263$	50	01050000000100000001020000002C0000
46	25917313	Via Luigi Galve	residential		0	0	44.603907691143 7C	30	0105000000010000000102000000090000
47	26405674		primary	SS63	0	0	47.5032735002083$	70	0105000000010000000102000000170000
48	26405675		primary	SS63	0	1	47.1610313336111	70	0105000000010000000102000000020000
49	26405847	Via Gnana	unclassifie		0	0	45.4688883159999$	30	0105000000010000000102000000130000
50	26406351	Via Fora	secondary	SP19	0	0	63.8627974030555$	50	01050000000100000001020000003B0000
51	26406729	Via Don Bosco	secondary		1	0	50	50	0105000000010000000102000000050000
52	26406817		primary	SS63	1	0	23.617111593000OC	70	0105000000010000000102000000020000
53	26406818		primary	SS63	0	0	23.617111593000OC	70	0105000000010000000102000000150000
54	26406913		primary	SS63	0	0	46.4703915376666€	70	0105000000010000000102000000050000
55	26406914		primary	SS63	0	0	47.8232735002083$	70	0105000000010000000102000000260000
56	26406941	Via degli Alpir	unclassifie		0	0	36.0711902176428$	30	0105000000010000000102000000430000
57	26432187		primary	SP63R	0	0	59.9005779566042$	70	0105000000010000000102000000070000
58	26432221	Variante Strade	primary	SS62Var	0	0	75.750000000000OC	70	01050000000100000001020000000B0000
59	26485783		primary	SP486r	0	0	63.8550592691666€	70	0105000000010000000102000000030000
60	26485821		tertiary		0	1	41.7999999999999$	50	0105000000010000000102000000030000
61	26485733	Via del Gualant	residential		0	0	30	30	0105000000010000000102000000020000
62	26485841		tertiary		0	0	41.7999999999999$	50	0105000000010000000102000000220000
63	26485868		unclassifie		0	0	27.9099999999999$	30	0105000000010000000102000000130000
64	26566009		primary	SP486r	0	0	66.6728337416666€	70	0105000000010000000102000000050000
65	26566023		primary	SP486r	0	0	55.6231049843472$	70	01050000000100000001020000000A0000
66	26566026	Via Ca' Geto	tertiary		1	0	61.819088785000OC	50	01050000000100000001020000000C0000
67	26783094	Via Elba	residential		0	0	44.5293865287532$	30	0105000000010000000102000000050000

Figure 3.4: Estimated velocities - Phase 3

neighbour is also missing the next nearest neighbour is used. In practice, the nearest method assigns the same values to adjacent time ranges, which is an acceptable hypothesis especially, for velocities on mountainous road links.

The result of the query is a road network that includes information on average velocities for all the twenty-four time ranges and for each link. The road network is then converted into a graph of about 800.000 nodes and about 2.700.000 links.

3.3.3 Shortest time alternatives

Time-based shortest path between each observed origin and destination is computed using an A* algorithm (Hart et al. (1968)) that has a better performance in terms of time with respect to the Dijkstra (Dijkstra (1959)). The algorithm exploits the average velocities, estimated using the observations, on the road links to compute the optimal path in terms of time. For each of the 13 thousand paths, the length and the travel time have been measured. Furthermore, a GIS (Geographical Information System) shape file for each shortest path has been created to allow the visualization of the path with the GIS. One of the main advantage of the GIS is the possibility to compare observed and optimal alternatives in terms of route choice based on the visualization of the spatial characteristics of the paths. A map matching over the road network would, therefore, allow to identify the selected paths (Miwa et al. (2012)). In this work, GIS information is used to compare simulated and observed alternatives.

3.4 Methodology

This section outlines the methodology proposed to compare the observed alternatives to the shortest time paths and describes how the simulated paths have been obtained and compared to the observed alternatives. The comparison between the observed and shortest time alternatives has been performed using a spatial query that was run using PostgreSQL 9.0 with the PostGIS 2.0 extension, which allows to perform queries on tables containing geometry information. All the shape files of the shortest time alternatives and the coordinates characterizing the observed routes have been imported in the database. The query counts the number of GPS coordinates (for each observed path) that intersect the corresponding optimal alternative path, represented as a polyline. The procedure to compare the two datasets consists in counting the number of points, for the observed path – which are included within few meters from the optimal alternative (50 meters). The second step of the analysis consists in the comparison between observed paths and modelled routes, obtained with one of the most popular planning tools, PTV Visum (PTV (2012)). The same network has been imported and calibrated into the simulator. Since the average velocities were available for all time ranges and most of the observed alternatives were performed during the morning peak, the calibration phase focuses on the 8-9 AM time period. Information about speeds and observed links flows have been used, aiming at simulating the congestion and reproducing realistic link flows. As first step, artificial OD flows have been loaded on the network in order to reproduce observed link flows. For the structure of the data, it was not possible to use traffic counts inside an optimization problem, so the aim in this phase was only to reproduce reasonable link flows on the network. This configuration has been used as starting point for the real calibration phase, which has been performed using measured speeds, derived by GPS coordinates. The problem has been formulated as an optimization problem, in which the distance between simulated and observed speeds was minimized using a deterministic gradient-based approach. The output is a calibrated network, in which the error between simulated and observed speeds is less than 6% (according to the RMSE metric). For the route choice analysis, a small demand is assigned on the loaded network. Three OD pairs have been selected, excluding the urban area, in order to avoid having a high number of similar alternatives. The Visum's default route choice models are used in this study. During the calibration phase, a “Deterministic User Equilibrium” approach has been preferred, while to analyse the route choice model, a stochastic assignment method has been used to propagate the vehicles on the loaded network. In this part we describe the stochastic model, which is the one used to obtain our results, according to Visum. The stochastic assignment assumes that users choose the shortest path, but it considers the possibility to have a different perception of travel time for different users, so the shortest path is not uniquely defined for all the demand. The demand for each route is thus distributed according to a distribution model which establishes the share of demand. In the experiments, we used the Kirchhoff distribution model to simulate the choice. The output of the model consists of 171 simulated paths for all the three OD pairs. The set of modelled alternatives have been compared to their corresponding observed paths in terms of length, travel time and overlapping percentage. The modelled alternatives have also been loaded into the GIS using the node coordinates for the selected links. The last part of this work focuses on the reliability index, which has been computed for each path. The indicator used is the one proposed by Kaparias et al. (2008), which provides a measure of how early/late a driver arrives at destination by providing two measures: the lateness and the earliness reliability indexes.

$$r(l) = exp\left[\frac{1}{2} * T_{log(l)} - z * \frac{\alpha}{2} * \sqrt{T_{log(l)}}\right] \tag{3.1}$$

Where $T_{log(l)}$ is a dimensionless variation logarithm, computed using average and variance travel time. The main characteristic of the index is that it does not depend on the length of the link. Therefore if a link is divided into two without changing its physical characteristics, the measure of reliability remains unaffected. In this work, only the lateness reliability index is considered to evaluate observed and modelled route choices because it should be more important for the driver how late the alternative route allow to arrive at destination. The objective of this analysis is to explore the possibility to consider into the route choice model the reliability explicitly as parameter, evaluating the influence of this element on the user route choice.

3.5 Results

Results allow to make considerations on the preferences of habitual drivers, when they make decisions on alternative routes during the peak hour, for a specific OD pair.

Figures 3.5, 3.6 and 3.7 show the results of the first analysis, which takes into account the entire dataset of clustered paths comparing them to their optimal shortest time alternative. The bullets are the low frequency GPS coordinates, the

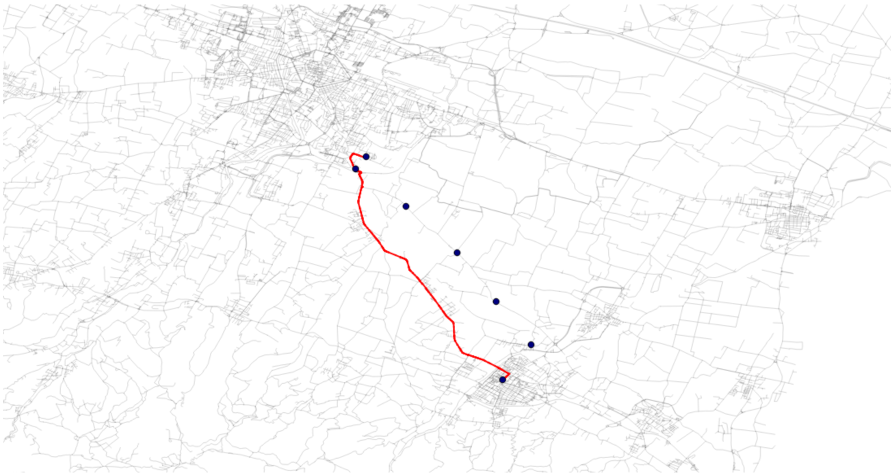


Figure 3.5: Overlap 43%

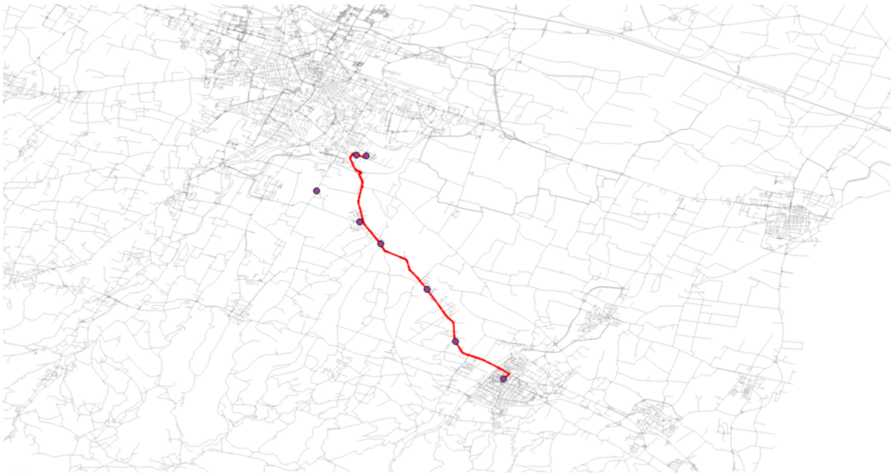


Figure 3.6: Overlap 75%

line represents the shortest time path. The figures show the selection made by the same driver during two different days and between the same origin and destination pair. The corresponding optimal path, in terms of travel time, has been computed using the hourly average velocities, therefore the timestamp of the origin zone for the observed path is used as reference point by the algorithm when velocity hourly time range is selected. In the two examples, however, the optimal alternatives do not differ from each other. This is often the case since drivers tend to travel between the same OD during similar hours of the day. The same analysis has been done for all the observed paths between the three OD pairs.

Table 3.2 shows results for different overlapping percentages for all the 13.766 paths. The values indicate that only ¼ of the alternatives overlap the shortest path for

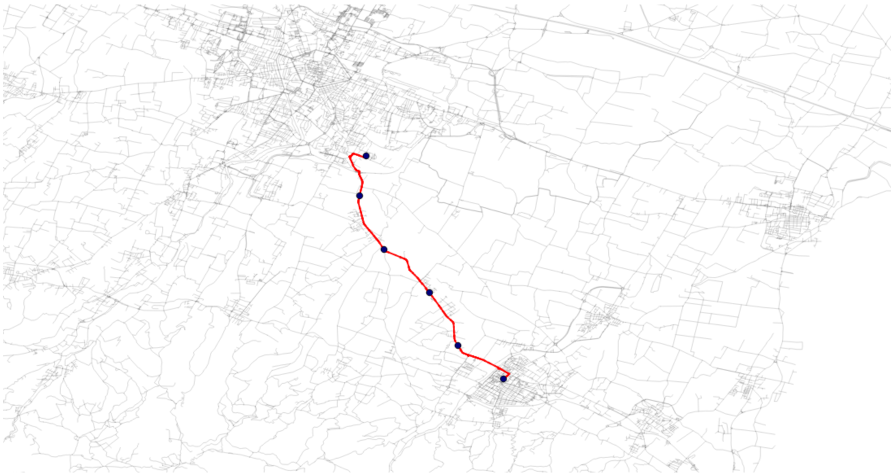


Figure 3.7: Overlap 100%

more than the 80%. This result is considerably different from the values reported in the literature, where about 40% of the users select paths that overlap the shortest time path for more than the 90%. It is relevant to point out that, to the best of our knowledge, this is the first work in which the comparison between optimal and observed alternatives considers a congested network, and the real shortest path time is computed using observed speeds on the congested network.

Table 3.2: Overlaps percentage of paths

Overlap	***%***
100%	15.07%
90-99%	1.46%
80-89%	9.62%
70-79%	9.57%
60-69%	11.10%
50-59%	4.89%
40-49%	12.17%
30-39%	13.63%
20-29%	11.11%
10-19%	4.52%
0-9%	0.03%

The next step of this work consists in the comparison between the observed and simulated route choices. The authors aim is investigating both the reasons for the anomalous result obtained in the previous step, both to analyse the model capability in replaying the route truly chosen by the users. Figures 3.8, 3.9 and 3.9 show the results of the analysis of a sample of three OD pairs. The observed route, the simulated route and the reliability factor are shown for each OD pair. The thicker lines indicate the alternatives that overlap with the observed routes. The graphs show that the selected observed alternative is provided by the model but the model assigns, generally, a low probability of selection because the cost is higher. Figure 3.8 indicates that overall, there are 6 different modelled alternatives between OD1 and that the driver has selected one of them in all cases. The reliability of the alternative selected by the driver is 87.3%. However, Visum has assigned only the 10% of probability to that observed path.

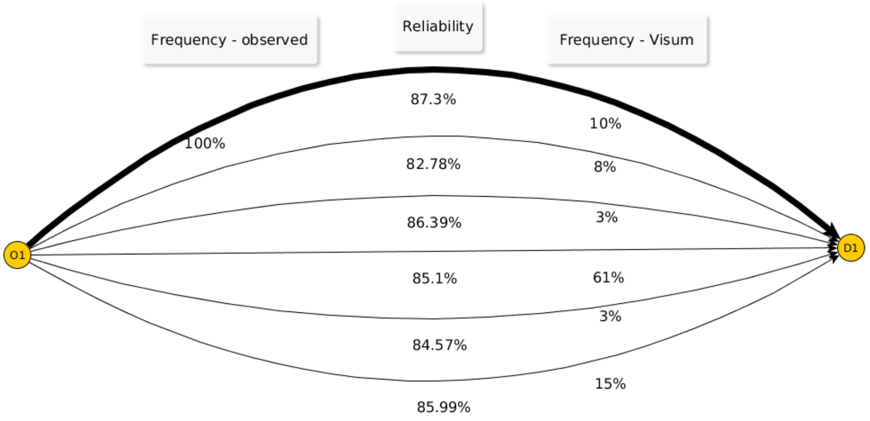


Figure 3.8: OD1

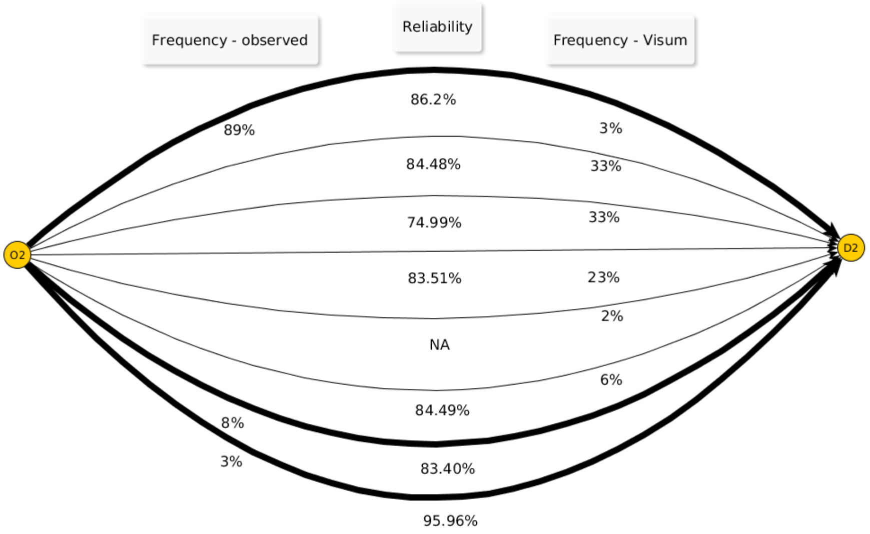


Figure 3.9: OD2

Figure 3.10 shows a similar structure of route selection. The driver has always selected the same alternative that has the highest value of the reliability index

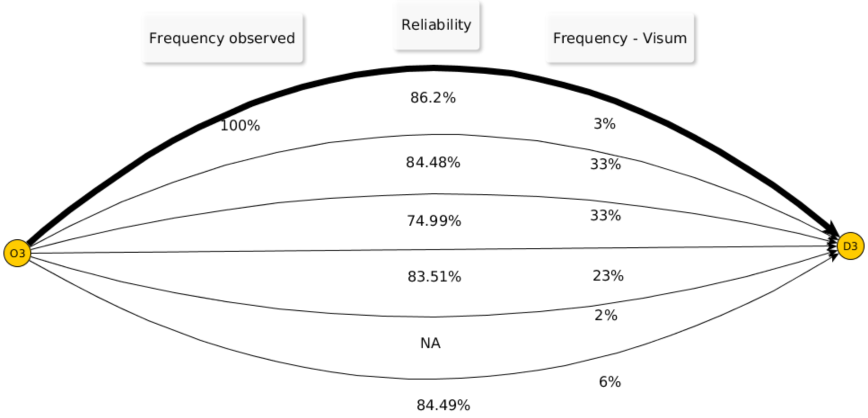


Figure 3.10: OD3

comparing to the other alternatives (86.2%), while only few users (3%) choose this alternative in the model. Finally, Figure 3.9 shows an example with different characteristics in which the observed routes are three. In the real case, the driver selects one preferred alternative with a probability of 89%. One of the modelled alternatives does not have a measure of reliability. The reason is that, this alternative has been selected only once by the model so that, the variance of travel time is zero.

Figure 3.11 shows the comparison of average travel time for each couple of origin and destination. As specified before, the travel time is obtained, both for simulated and observed paths, importing in the GIS the coordinates and computing the travel time through the measured speeds. In this way, the comparison is not affected by the approximation of the model.

In all three cases, the average travel time of the modelled alternatives is lower compared to the travel time of the observed routes. This confirms that drivers prefer routes that are more costly in terms of time. With reference to OD3 of Figure 3.11, the difference of travel time between the observed and optimal alternatives is much higher comparing to the other groups of ODs. Here the reason is that four of the observed alternatives are much longer than any of the modelled ones because the driver decided to turn around the city centre instead of selecting straightest routes. Consequently, the travel time is also higher. Figure 3.12 shows the three observed alternatives of OD3 and the corresponding shortest time path (continuous line). One of these alternatives is longer comparing to the others because it turns around the city centre. This alternative will be never created by the model.

The shortest time path overlaps the observed alternatives only partially. Further-

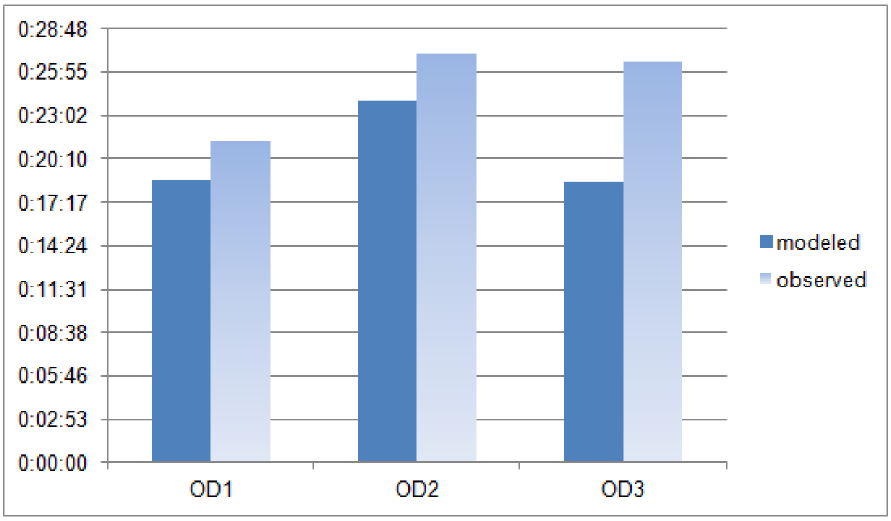


Figure 3.11: Modelled and observed average travel time

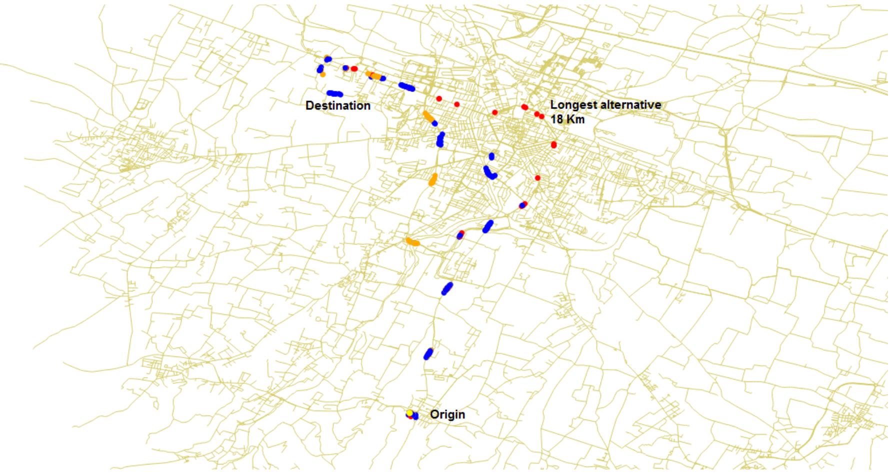


Figure 3.12: OD 3 – observed alternatives and shortest time path

more, among the modelled routes, there is one alternative that completely overlaps the selected routes as shown in Figure 3.10.

The previous analyses have shown that drivers may take into account the reliability of an alternative in route choice. Here three paths, performed in a very similar time range (10-10.40 AM) of three different days, for each alternative are selected to analyse more in detail the relation between drivers' route choices and reliability of a path.

Figure 3.13 shows the three observed alternatives of OD1. It must be noticed that in Figure **??**, only one observed alternative was considered for the analysis because of the very little difference between the observed and optimal alternative. The observed paths differ from each other only in the last section. More specifically, the first section (Part 1) of OD1 has three different alternatives that have been selected by the driver in three different days; the last section that arrives at destination (Part 2) is in common for all the alternatives.

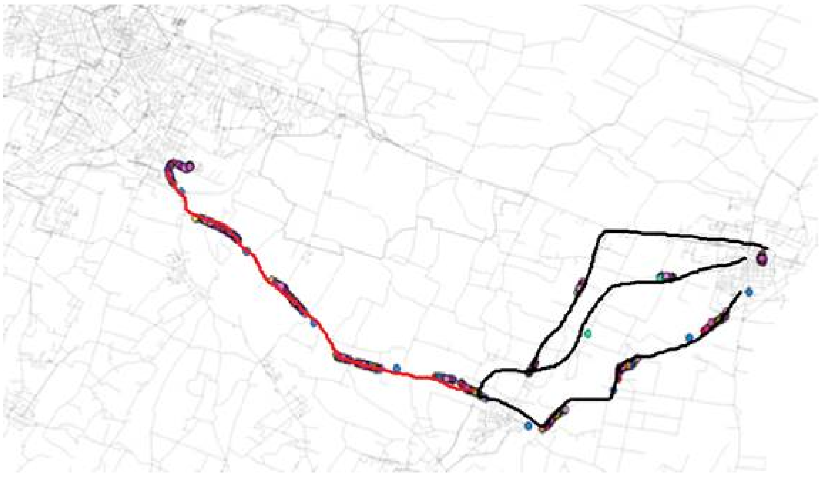


Figure 3.13: OD1 - observed alternatives

This last analysis is focused on the first section of the path which is characterised by three alternatives and, it consists in the comparison with the reliability computed on the common section. Table 3.3 reports the reliability of each of the three alternatives (part 1). The common part (part 2) has a reliability of 73.72%. These figures indicate that the reliability of the three alternatives is, in all three cases, lower comparing to the common section. Moreover the higher value of the reliability (61.69%) corresponds to the most frequent observed alternative. Thus, the paths that are closer to the urban centre seem to be less reliable and the driver prefers alternatives that are in general more reliable.

Table 3.3: Reliability of the three alternatives

Part 1	Reliability
OD1_1	59.36%
OD1_2	60.42%
OD1_3	61.69%

3.6 Conclusion and future work

This chapter presented an empirical tests on the assumption of using the shortest path in the reality. In the first part, results are obtained analysing a significant number of different paths, for different OD pairs. The analysis, among more than 14000 repetitive trips, shows that only a small part of the users (about 25%) select the shortest path for their trip. By the comparison of the shortest path and of the network model, it is possible to argue that, while simulators prefer to load the more capacitated roads, in reality drivers prefer local roads, spending more time to reach the destination but avoiding the congestion. Another element that is not well represented in the simulator is the tendency to choose longer routes. Several users prefer to take a longer route, bypassing the city and coming back using a sort of *circular* route, in order to avoid the most direct and, so, the most congested route. For this observation, the first conclusion is that many real routes cannot be replayed in the model, since users tend to use routes that strongly differ with respect to the *best* one, while the model prefers as direct as possible alternatives and routes that are similar to the shortest path. Furthermore, while drivers have the tendency to choose totally independent alternatives, the modelled ones are similar and strongly correlated. Another relevant element considered in this work is the reliability index, defined as the day-to-day travel time variance. Looking at the results, users have the tendency to accept some more delay in their travel time if the route presents a higher reliability. Stochastic assignment models should consider a reliability parameter to penalize the travel time on those routes which are not reliable, pushing the demand on those which are the most. Results clearly show that the greatest share of observed users moves on the most reliable path. For those OD pairs in which the difference in travel time is limited, all the observed users move on the same path. Presented results show group of paths with the common origin and destination, as explained in the methodology section. Routes, even if very similar, are slightly different in reality. As pointed out, the main difference is generally in the beginning and the ending of the route, since origin and destinations are often different.

Results from this work are very relevant in understanding the limits of the actual route-choice within well-established assignment models. More elements that may have an influence on the route choices should be investigated: among others, the overlapping between round trip routes and symmetric ODs, the repetitivity in the day to day route choice, or if route choices are influenced by trip chains, or if there are topological elements that influence decisions.

4 Multiclass SVM of LiveBus Arrivals users

4.1 Introduction

In order to offer public transport services that meet citizens' needs, Public Authorities and transport planners need to offer new services that allow the users to easily access the information. The Countdown service in the city of London provides digital signs at stops. The same information can also be retrieved using a pc or a smartphone. The service provided by the local public transport authority, Transport for London (TfL), is called LiveBus Arrivals and it relies on the iBus AVL system. The service provides information on the real time arrival of a bus line at stop in the London network. The user accesses the TfL website, using a web browser on a PC or a dedicated app for smart phones. Requests can be of four types:

- The user types the bus number and, if there are two possible route directions, chooses one. The output is a list of stops' names and the user needs to select one;
- The user types the stop number and gets a list of buses approaching the stop with their corresponding time of arrival;
- The user types the name of the stop and will get the same output as above;
- The user writes the number of bus line and gets a map of the route.

One of the first work on transit passengers and real time information was proposed by Hickman (1997) who described a path choice model by incorporating real time information. He introduced stochastic and time-dependent travel time in the path choice model. The objective was to take into account the influence of a real time information system on alternative origin to destination choices. However the results showed that real time information does not provide significant benefits for transit passengers path choices in terms of reduction of travel time. In a recent work, Cats et al. (2011) propose a dynamic transit path choice model and evidence that potential time savings are associated with the provision of real time information. Trozzi et al. (2013) developed a time dependent route choice model to determine the impact of the count down system under overcrowding scenarios. They found that live information do not lead to a reduction of travel time but information modify travel behaviour of passengers who prefer less congested interchange stations.

Other studies focus on the development of algorithms and interfaces for real time transit planner. For instance, Jariyasunant et al. (2010) developed a mobile real time system to implement a K-shortest path algorithm using bus arrivals predictions. The user types the origin and the destination and obtains a personalised shortest path on a map. The results of performance analysis show only a marginal improvement of travel time.

Most of the studies aimed at analysing OD flows in public transport explore the possibility to use mobile phone data or smart cards to analyse and describe the patterns that characterise people behaviour. For instance, Munizaga and Palma (2012) estimated a multimodal public transport OD matrix from SmartCard data and found that their method allowed to estimate 80% of the total boardings. Although the dataset did not included information on the destination, the alighting station has been estimated by looking at the passengers' position and the time of the next boarding.

The requests made by the users on the real time arrivals at a bus stop have been, until now, mainly deployed to analyse the influence of this type of information on the use of bus transport and on the estimation of the perception of waiting time. The effects of real time information displays at stops on waiting time, on easiness to use the system, on willingness-to-pay, on mode choice and on customer satisfaction has been analysed by Dziekan and Kottenhoff (2007). They found that the provision of real time information at stop has the capability to reducing perceived waiting time. Additionally, they found that it is five time cheaper to introduce real time information systems with the aim to reduce perceived waiting time than increasing the frequency of a public service. Finally, they show how travellers react at real time information by running when they see that there are only few minutes left until the train departs. Tang and Thakuriah (2012) analysed longitudinal data on bus ridership and related them to employment rate, gas prices, weather conditions and other socioeconomic characteristics to study the impact of real time information on public transport usage. They found variations over time on bus ridership due to the installation of real time information but they were not able to provide conclusion on geographical variations.

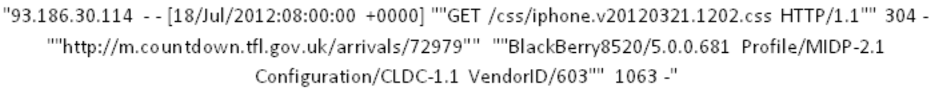
The present work explores the possibility to replace, enrich or complement the census data on the origin and destination (OD) flows between different boroughs of the city of London. Several studies are focused on the possibility to replace survey performed by doing interviews. Becker et al. (2011a) used mobile phones call records to explain the dynamic of people in the city. They propose innovative ways to visualize cellphone activities and a classification based on phones usages to explain the city dynamics. In a second publication of the same year (Becker et al. (2011b)), the authors analyse drivers' route choices using mobile phones data and they present an algorithm to match the data to the road network to measure traffic volumes. Finally, they validate the results of their analysis using the statistics issued by the transportation authority. Ratti et al. (2006) present an exploratory analysis of urban patterns in the city of Milan using mobile phones with location data. They underline the need to study the dynamics of people over time because this would allow to understand the reactions due to external factors such as a disaster, a football match, the introduction of new public transport services, etc. An overview of possible applications and use of cell phones data for public transport is provided in Holleczek et al. (2014). The authors use mobile phones and smart card data to estimate the population of public transport users and they study the OD flows between the districts of the city of Singapore. The final aim of the study is to identify the weak links of the public transport network during different peak hours.

Hardy (2012) analysed Live Bus Arrivals requests by time of the day, location and type of channel used to retrieve the information (PC, a smart phone or the SMS service). The study showed that the demand for real time information was higher during peak hours meaning that the service is mostly used when waiting time is longer. Watkins et al. (2011) carried out a study on waiting time perception for users with and without real-time information. They discovered that perceived waiting time was 30% less for users of real time information. Additionally they found that mobile real time information reduce the actual waiting time because users arrive at the stop closer to the bus arrival instead of being there well in advance.

4.2 The dataset

The data analysed in this work refers to period included between the 16th July 2012 and 10th August 2012. The week under examination covers the final stages of preparation for the Olympic Games that were held in London between the 27th July and 12th August 2012. The period corresponds also to the last week before most schools closed for summer holidays.

The original file consists of about 4 million rows for each observed day. Each string includes the IP address which usually refers to one or a set of requests sent by the same user, the time and the day of the request, the bus and/or the stop number, the cellphone model and the browser installed. An example of string is shown in Figure 4.1. The format of the file includes univocal keywords and non structured system calls from which it is necessary to identify and extract the needed information such as stop codes and line buses.



```
"93.186.30.114 - - [18/Jul/2012:08:00:00 +0000] ""GET /css/iphone.v20120321.1202.css HTTP/1.1"" 304 -
""http://m.countdown.tfl.gov.uk/arrivals/72979"" ""BlackBerry8520/5.0.0.681 Profile/MIDP-2.1
Configuration/CLDC-1.1 VendorID/603"" 1063 -"
```

Figure 4.1: Original Data

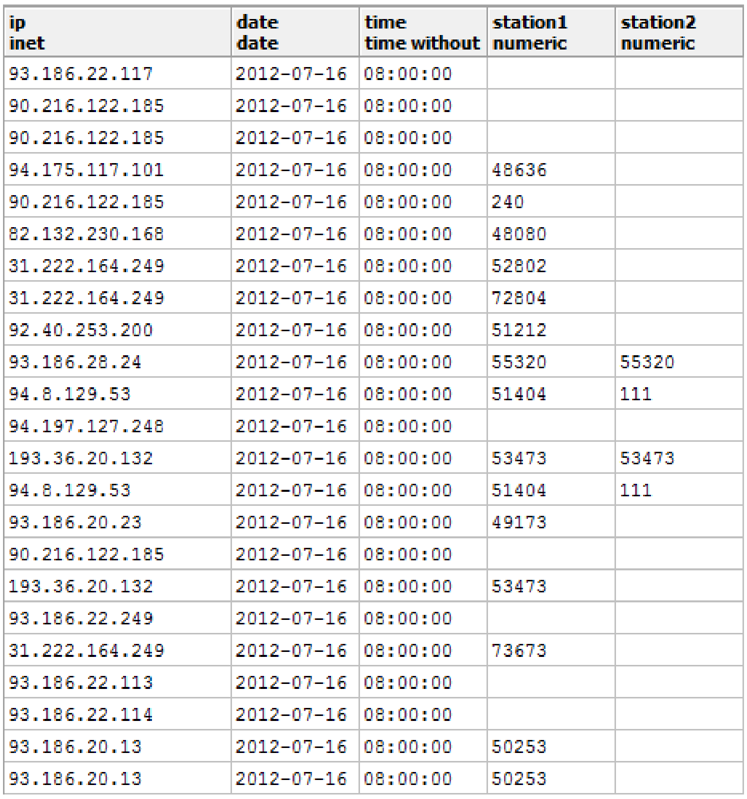
The example shows a request for real time arrivals at Chatteris Avenue (stop number 72979). The output provided to the user consists of information related to three bus routes due to arrive at Langbourne Place (72979): the 256, 499 and 174. This single string does not provide any information on which bus routes the user is waiting for.

Table 4.1: Type of string

	String
1	/arrivals/
2	/stopsNearLocation/
3	stop=
4	/myStops/
5	searchTerm=
6	showJourneyPattern/
7	/route/
8	/stopBoard/
9	/stopId/

To analyse bus passengers' behaviour using Live Bus Arrivals data, the first step was to extract the number of stops and/or bus routes from each string. As a general rule, buses and stops are always preceded by one of the keywords listed in Table 4.1. The resulting file was then imported into the relational database Postgresql 9.3 for data cleaning and elaboration. The output is a list of requests with several empty rows, bus routes and stop codes either in one of the last two columns of the table as reported in Figure 4.2. In the second phase, bus routes and stop codes have been positioned in their corresponding columns and duplicates or empty rows have been eliminated. A screen-shot of cleaned data is shown in Figure 4.3 in which five different users have requested either for a bus line or a stop arrival. Two requests for time of arrival at Cloister Gardens (48215) were asked by the first user (2.25.69.66) at 6:41 and 6:45. Another couple of requests are associated to the same IP address and have been sent at 18:02 for stop Oakmead Gardens (50049). Although the two stops are very close to each other, and the two IP addresses are equal it is not possible to know with certainty if the two requests have been sent by the same user. The Mobile Network Operator assigns different IP addresses to mobile users especially when they access the web or when they change their position. Since the same IP address can be assigned to different users, it cannot be deployed as unique id identifier. Nevertheless the IP address can help to identify multiple consecutive requests sent by the same user in a short period of time. To reduce data misinterpretation, in this work the IP address will be used to identify a single user only when the requests are sent within a hour. With reference to the other users of Figure 4.3, multiple requests were sent within one or fifteen minutes. Specifically, user 2.25.70.14 by sending multiple requests for the same stop and bus route in a time range of about fifteen minutes has probably boarded at Clamp Hill (57184) on bus 258. The time elapsed between the last and the first request can be used as an indicator of user's waiting time at stop. Or as an indicator of perceived waiting time longer than expected.

Finally a further data elaboration was required to help in the analysis. First of all requests in each hourly time range have been grouped and georeferenced. Moreover, for each hourly range and for each user, repeated requests for a stop and/or for a bus have been counted. This information is included in an additional column named "count".



ip	date	time	station1	station2
inet	date	time without	numeric	numeric
93.186.22.117	2012-07-16	08:00:00		
90.216.122.185	2012-07-16	08:00:00		
90.216.122.185	2012-07-16	08:00:00		
94.175.117.101	2012-07-16	08:00:00	48636	
90.216.122.185	2012-07-16	08:00:00	240	
82.132.230.168	2012-07-16	08:00:00	48080	
31.222.164.249	2012-07-16	08:00:00	52802	
31.222.164.249	2012-07-16	08:00:00	72804	
92.40.253.200	2012-07-16	08:00:00	51212	
93.186.28.24	2012-07-16	08:00:00	55320	55320
94.8.129.53	2012-07-16	08:00:00	51404	111
94.197.127.248	2012-07-16	08:00:00		
193.36.20.132	2012-07-16	08:00:00	53473	53473
94.8.129.53	2012-07-16	08:00:00	51404	111
93.186.20.23	2012-07-16	08:00:00	49173	
90.216.122.185	2012-07-16	08:00:00		
193.36.20.132	2012-07-16	08:00:00	53473	
93.186.22.249	2012-07-16	08:00:00		
31.222.164.249	2012-07-16	08:00:00	73673	
93.186.22.113	2012-07-16	08:00:00		
93.186.22.114	2012-07-16	08:00:00		
93.186.20.13	2012-07-16	08:00:00	50253	
93.186.20.13	2012-07-16	08:00:00	50253	

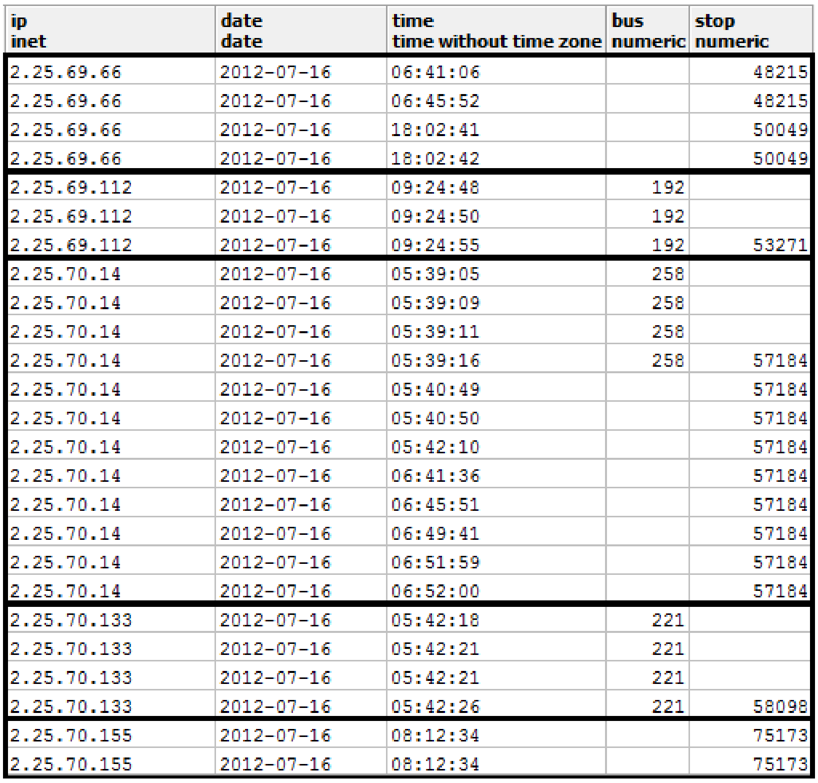
Figure 4.2: Extracted data

4.3 Classification of Live Bus Arrivals users behaviour

This section describes user behaviour by the identification of few main groups of customary users of the Live Bus Arrivals service. These groups belong to two main categories based on the information Live Bus Arrivals users provide by sending their requests for time of arrival at one or more stops or for a bus route. These information can be complete, partial or on bus-to-bus transfer. The possibility to have complete information on passengers trips depends on two factors: 1. the accuracy of the information requested by the passenger and 2. if the time of arrivals is asked for more than one stop, the possibility to unequivocally identify the bus route passing by the requested stops.

4.3.1 Complete information

In case of complete information, the most accurate data that a user can provide consists of the code number of boarding and alighting stops and the bus route. Some



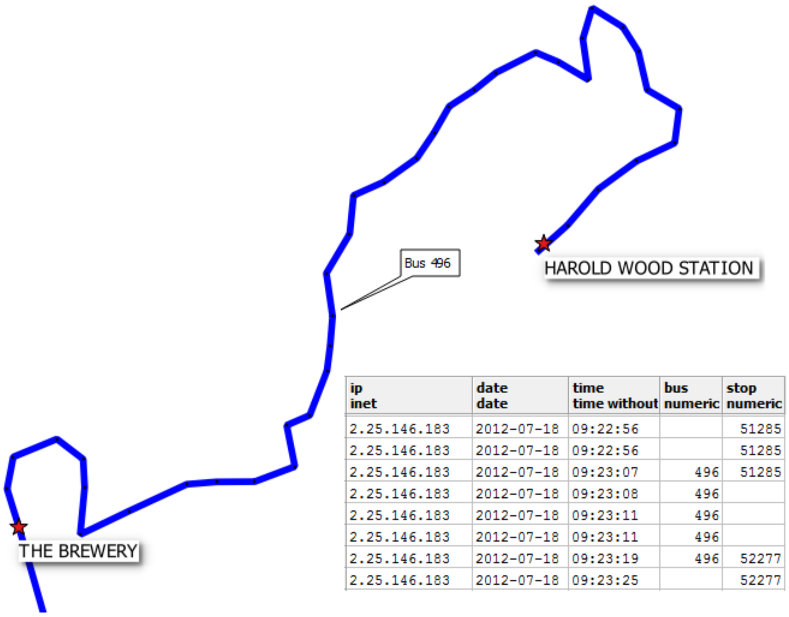
ip inet	date date	time time without time zone	bus numeric	stop numeric
2.25.69.66	2012-07-16	06:41:06		48215
2.25.69.66	2012-07-16	06:45:52		48215
2.25.69.66	2012-07-16	18:02:41		50049
2.25.69.66	2012-07-16	18:02:42		50049
2.25.69.112	2012-07-16	09:24:48	192	
2.25.69.112	2012-07-16	09:24:50	192	
2.25.69.112	2012-07-16	09:24:55	192	53271
2.25.70.14	2012-07-16	05:39:05	258	
2.25.70.14	2012-07-16	05:39:09	258	
2.25.70.14	2012-07-16	05:39:11	258	
2.25.70.14	2012-07-16	05:39:16	258	57184
2.25.70.14	2012-07-16	05:40:49		57184
2.25.70.14	2012-07-16	05:40:50		57184
2.25.70.14	2012-07-16	05:42:10		57184
2.25.70.14	2012-07-16	06:41:36		57184
2.25.70.14	2012-07-16	06:45:51		57184
2.25.70.14	2012-07-16	06:49:41		57184
2.25.70.14	2012-07-16	06:51:59		57184
2.25.70.14	2012-07-16	06:52:00		57184
2.25.70.133	2012-07-16	05:42:18	221	
2.25.70.133	2012-07-16	05:42:21	221	
2.25.70.133	2012-07-16	05:42:21	221	
2.25.70.133	2012-07-16	05:42:26	221	58098
2.25.70.155	2012-07-16	08:12:34		75173
2.25.70.155	2012-07-16	08:12:34		75173

Figure 4.3: Cleaned Data

variants are also possible which provide complete information such as passengers asking information not only for bus arrivals at origin and destination but also at intermediate stops. Complete information can be obtained if a passenger specifies the stop of origin and destination and there is only one line passing by both stops. Figure 4.4 displays the table with the list of requests for real time arrivals of a bus route (496 to Harold Wood), two stops, Harold Wood Station (51285) and The Brewery (52277), and a graphical representation of the requests. These types of requests allow unambiguous interpretation and can be used as disaggregated data on passengers' origin and destination.

4.3.2 Partial information

Partial information on trips is obtained when a passenger asks about the time of arrival of a bus at the stop of boarding or indicates both the stop number and the bus route. This is the case in which a user accesses the Live Bus Arrivals system by typing the number of bus and selects one of the stops from the list. Although the destination is not known, these data can provide information on stop departures and/or on route's flows.



ip inet	date date	time time without	bus numeric	stop numeric
2.25.146.183	2012-07-18	09:22:56		51285
2.25.146.183	2012-07-18	09:22:56		51285
2.25.146.183	2012-07-18	09:23:07	496	51285
2.25.146.183	2012-07-18	09:23:08	496	
2.25.146.183	2012-07-18	09:23:11	496	
2.25.146.183	2012-07-18	09:23:11	496	
2.25.146.183	2012-07-18	09:23:19	496	52277
2.25.146.183	2012-07-18	09:23:25		52277

Figure 4.4: Complete information

Repeated requests. During the week under observation, the 10,2% on average of the Live Bus Arrivals users have sent more than 5 repeated requests between 7 and 8 a.m. When asking for real time arrivals, users continuously update the system, probably until the bus arrives. Figures 4.5 shows the average number of requests sent between 7 and 8 a.m. during Monday - Friday. Large circles indicate high averages of repeated requests (more than 20 requests within a hour for arrivals at the same bus stop). High average values are mainly located at stops outside the city centre.

Requests for live arrivals of a bus route at a boarding stop. When passengers ask for live arrival of a bus at a specific stop, the information is partial because there are no data on the destination but only on the stop of departure, on the bus route and its direction. For instance Figure 4.6 shows that the user has asked for bus W8 arrivals at Browning stop (47051), the route direction is Herefield Close. In this example even if the destination is unknown, there are only five stops until the end of the line.

Requests of a bus route. Sometimes users request information on a bus line to know the route direction or to have an overview of the list of stops. In this case no real time information on arrivals at stop is requested so that no data are available on the trip. However this information can be used to know how popular a

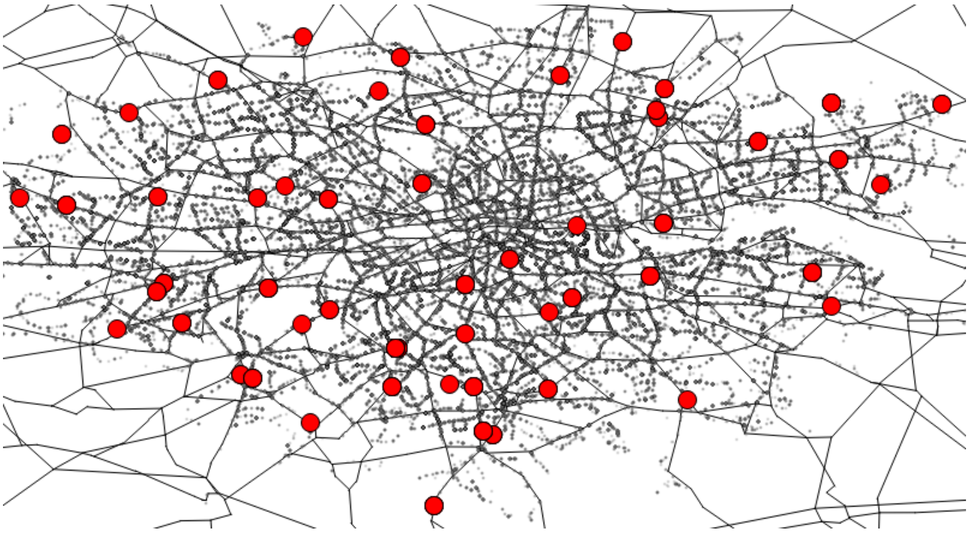
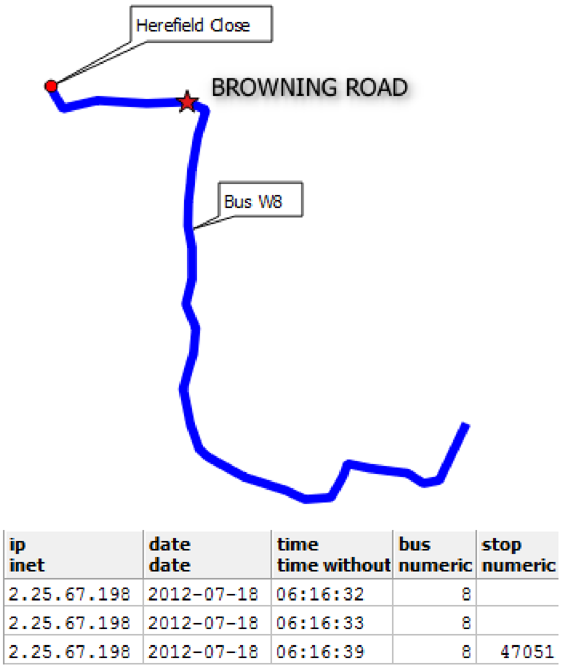


Figure 4.5: Average repeated requests (7-8 a.m) Monday, 16/07/2012 - Friday, 20/07/2012

particular route may be.

Requests at near stops. Furthermore information is partial when a user enquires about the time of arrival at near stops. When a user is interested on arrivals at neighbouring stops, there could be three possible reasons: a. the same bus is passing by the neighbouring stop and the user intends to walk there without missing the bus; b. the user is checking for information on another stop to transfer or board another bus; c. the user wants to use the bus only for very few stops. Figure 4.7 shows a request for information on two stops located near each other. In the example it is not possible to know which of the two bus lines, the 34 and 102, the passenger will board. Thus there is no information on the final destination. This data should be grouped together with the "Repeated requests" and should be handled accordingly to avoid misinterpretation (i.e. by considering the two stops as the origin and the destination).

Requests for same stop but opposite route direction. In some cases users look for Live Bus Arrivals at both directions of the same stop. Therefore it may be that the user cannot easily identify which is correct route direction. Bus route directions are usually identified with the stops at the end of the line. However public transport passengers probably know better about their stop of arrival instead of the end of the line. A new real time functionality could include the possibility to type the stop of arrival and, based on that indication, the system could provide

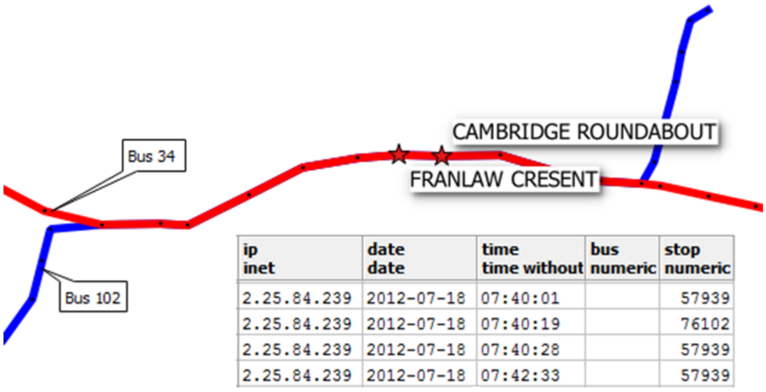


ip	date	time	bus	stop
inet	date	time without	numeric	numeric
2.25.67.198	2012-07-18	06:16:32	8	
2.25.67.198	2012-07-18	06:16:33	8	
2.25.67.198	2012-07-18	06:16:39	8	47051

Figure 4.6: Request for live arrivals of a bus route at a stop

some indication on which is the bus stop the user has to board. Indication of stops direction could consists in colours or letters applied at the stop, near the stop code. So that when a user types his origin and destination, the system could provide the indication on the correct stop to board.

Bus-to-Bus transfer. Passengers often use the Live Bus Arrivals service to get information on arrivals at the bus stop they want to transfer. For instance, Figure 4.8 displays the set of requests of a passenger who has boarded at Kingsbury station (72740). Six bus routes by Kingsbury station but the user has most likely boarded bus route 183 to Pinner Station. This is in fact the only bus route that allows to arrive close to where the next two requested stops (72141 and 59160) are located. The interpretation of user behaviour in the second part of the trip is complex. The two stops for which the user has requested real time arrivals are in reality the same stop (Hunters Grove) where bus routes transit with opposite directions. From these stops several bus routes transit: the 114, H9, H19, H10, H18. All of them have transfer points along route 183. Possible explanations on why the user has asked arrivals information at Hunters Grove are: 1. the user did not know he could transfer along line 183 to one of the buses passing by Hunters Grove so that has caught the bus from another stop; 2. the user knew that there was an easier transfer point but, based on real time of arrivals, has decided to get off one

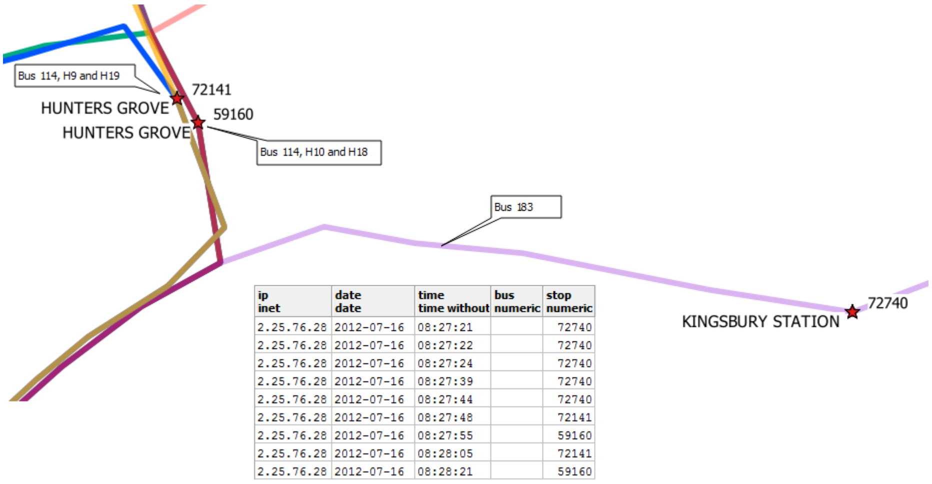


ip inet	date date	time time without	bus numeric	stop numeric
2.25.84.239	2012-07-18	07:40:01		57939
2.25.84.239	2012-07-18	07:40:19		76102
2.25.84.239	2012-07-18	07:40:28		57939
2.25.84.239	2012-07-18	07:42:33		57939

Figure 4.7: Requests for near stops

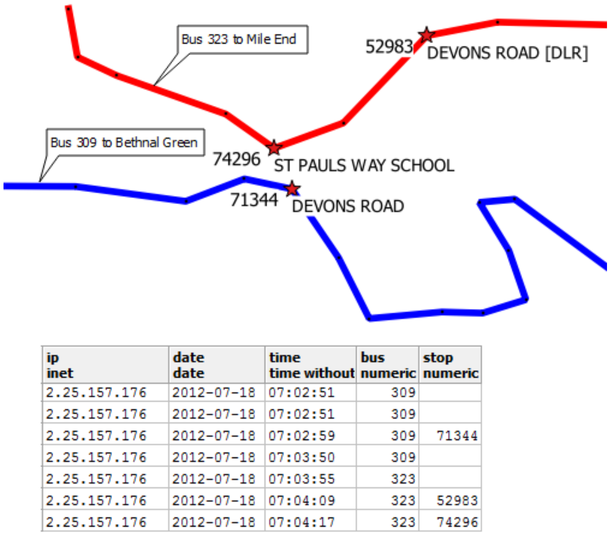
stop earlier and to walk to the transfer stop. The first hypothesis is less probable because the passenger has on-line information. Furthermore this behaviour and other examples that will be described in the next paragraphs suggest that public transport passengers make use of real time information when boarding from one bus line to another. So that a specific functionality aimed to quick and easily provide such type of information could be well accepted by the users.

Another example of bus-to-bus transfer which is easy to interpret is shown in Figure 4.9 where a passenger has asked for real time arrivals of bus route 309 to Bethnal Green at Devons Roads stop. The same user has then looked for departures of bus route number 323 to Mill End firstly from Devons Road (DLR), then to a closer stop, St. Pauls Way School stop.



ip inet	date date	time time without	bus numeric	stop numeric
2.25.76.28	2012-07-16	08:27:21		72740
2.25.76.28	2012-07-16	08:27:22		72740
2.25.76.28	2012-07-16	08:27:24		72740
2.25.76.28	2012-07-16	08:27:39		72740
2.25.76.28	2012-07-16	08:27:44		72740
2.25.76.28	2012-07-16	08:27:48		72141
2.25.76.28	2012-07-16	08:27:55		59160
2.25.76.28	2012-07-16	08:28:05		72141
2.25.76.28	2012-07-16	08:28:21		59160

Figure 4.8: Bus-to-Bus transfer: data interpretation



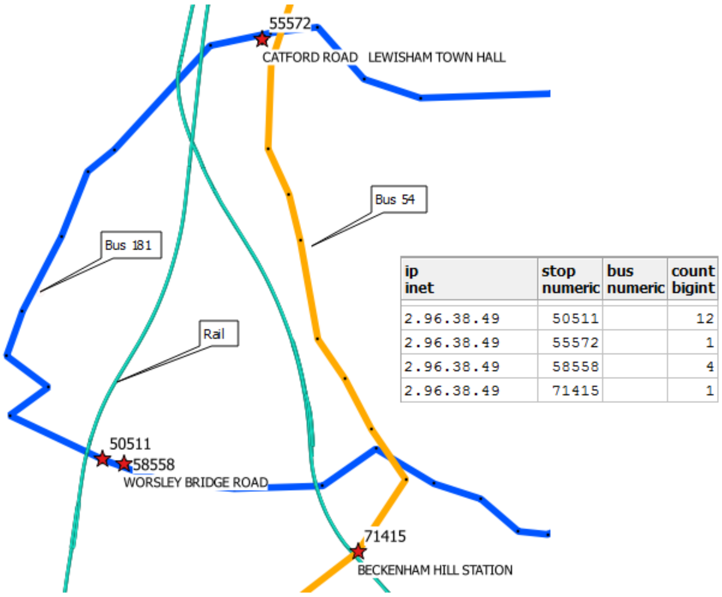
ip	date	time	bus	stop
inet	date	time without	numeric	numeric
2.25.157.176	2012-07-18	07:02:51	309	
2.25.157.176	2012-07-18	07:02:51	309	
2.25.157.176	2012-07-18	07:02:59	309	71344
2.25.157.176	2012-07-18	07:03:50	309	
2.25.157.176	2012-07-18	07:03:55	323	
2.25.157.176	2012-07-18	07:04:09	323	52983
2.25.157.176	2012-07-18	07:04:17	323	74296

Figure 4.9: Bus-to-Bus transfer at near stops

Generally the interpretation of Live Bus Arrivals data becomes complex when users look at two or more travel options. Figure 4.10 shows that twelve repeated requests have been sent to get live arrivals at Worsley Bridge Road to Catford and four requests have been sent for the same stop but for the opposite direction. Worsley Bridge Road is connected to Catford Road by bus route 181, while Beckenham Hill is connected to Catford by bus 54 or by rail. The graphical representation of the requests allows to understand that the user has looked at two possible routes to arrive at destination (Catford Road). The decision on which route to choose has been probably influenced by real time arrivals. So that the user has boarded either the 181 to Catford or the 181 to Beckenham Hill and then bus route 54 to Catford.

Figure 4.11 shows two types of requests for real time arrivals: the first at bus stop Leonard Avenue (51795), the second for bus route 174 at Reinham Road North (56184). In this case there are two optional bus routes that allow to arrive near to Reinham Road North from Leonard Avenue: bus routes 175 and 103. The most probable option is that the user has caught bus 103 to arrive close to Reinham Road North, walked a few hundred meters and transferred to the southbound bus 174.

Figure 4.12 shows an easy interpretation of bus-to-bus transfer. Firstly the user has asked information on arrivals at Whitworth Road to Upper Norwood and then at Springfield Road to West Croydon. The two lines intersect each other even if there is no stop in common. Given the direction of the bus lines which are known



ip inet	stop numeric	bus numeric	count bigint
2.96.38.49	50511		12
2.96.38.49	55572		1
2.96.38.49	58558		4
2.96.38.49	71415		1

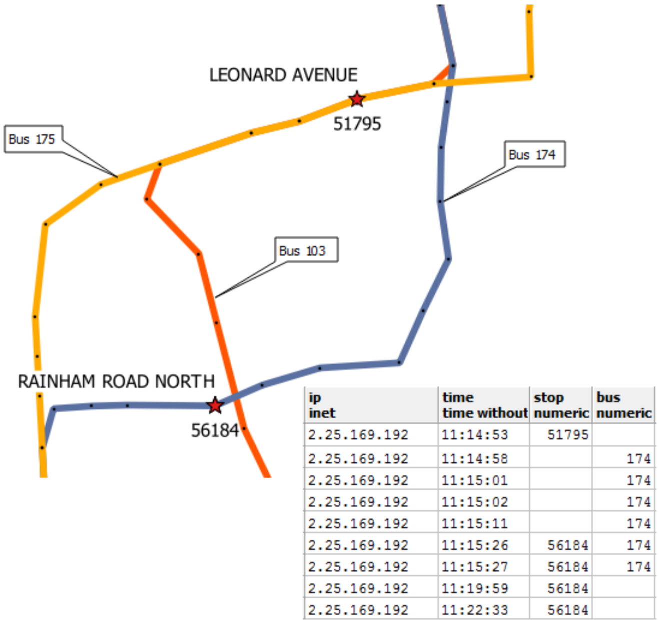
Figure 4.10: Bus-to-Bus transfer: alternative routes

thanks to the stops number, it is possible that the user has transferred from bus 468 to 450.

Finally, Figure 4.13 shows the last example of bus-to-bus transfer. In this case the traveller has firstly asked for information on arrivals at Tottenham Court Road Station (51056). Since from that station is possible to board many bus lines, it is necessary to look at the location of the second request which is Ebury Bridge (47551). The only bus route from Tottenham Court to a stop closed to Ebury Bridge is bus route 73 to Victoria. From Ebury Bridge it is only possible to board bus route C10 to Canada Water. It has to be noticed that 73 to C10 transfer was already possible at Victoria. Therefore this is another example of a user who has boarded from another stop instead of the one located very close to the one of the previous bus line (bus 73). A possible explanation is that, using real time information, users prefer to walk instead of waiting at the stop for the bus.

4.3.3 Outliers

Not all data can be used for the analysis of travel behaviour. For instance, some users ask for arrivals at two opposite direction stops. In this case is it not possible to define from which stop the passenger boards. Furthermore it is necessary to



ip inet	time time without	stop numeric	bus numeric
2.25.169.192	11:14:53	51795	
2.25.169.192	11:14:58		174
2.25.169.192	11:15:01		174
2.25.169.192	11:15:02		174
2.25.169.192	11:15:11		174
2.25.169.192	11:15:26	56184	174
2.25.169.192	11:15:27	56184	174
2.25.169.192	11:19:59	56184	
2.25.169.192	11:22:33	56184	

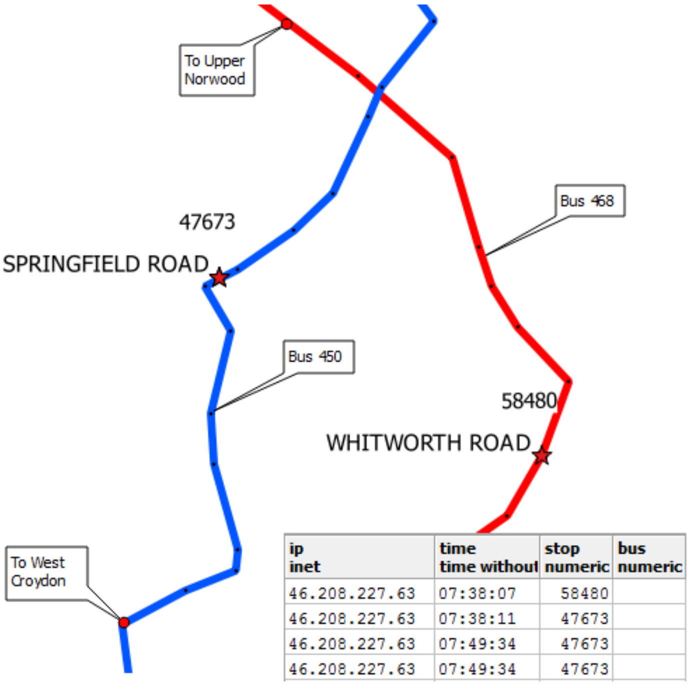
Figure 4.11: Bus-to-Bus transfer: first route identification

exclude users with behaviours that are not clearly defined for instance when many requests are sent during a hour for any stop around the city such as in Figure 4.14.

There are other types of requests which are not possible to interpret. For instance users of the Live Bus Arrivals service who request information on live arrivals at multiple stops within 10 km where several bus routes serve the requested stops. The intent of this type of request is probably to gather information on all bus routes transiting near by. However these data should be eliminated from the analysis because they cannot provide any precise information on user behaviour.

4.4 Methodology

This chapter describes the methodology to classify different behaviours of LiveBus Ariivals users. To this aim a set of classes that characterise users behaviour are identified and the methodology to compute the features that will allow to determine the class in which each observation belongs. Thus a subset of observations is selected for each of the identified classes to train the model and classify the entire dataset.



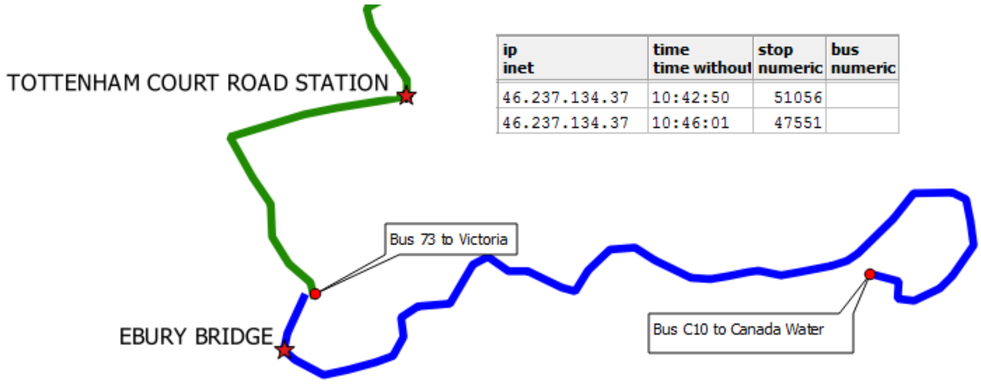
ip inet	time time without	stop numeric	bus numeric
46.208.227.63	07:38:07	58480	
46.208.227.63	07:38:11	47673	
46.208.227.63	07:49:34	47673	
46.208.227.63	07:49:34	47673	

Figure 4.12: Bus-to-Bus transfer

4.4.1 Classes of Live Bus Arrivals users

Following the description of some Live Bus Arrivals user behaviours, four main classes of users are selected with the objective to train a model and classify the entire dataset.

1. The first class relates to the users who send multiple requests for the same stop during a short period of time. These types of observations are characterized by values of the Feature 1 equal to 1 and values of Feature 3 generally low meaning that the time elapsed between consecutive requests is on average low.
2. The second class identifies users that request two or more stops that are located within 500 meters from each other. In this case, the value of Feature 2 is equal or less than 0.5 and the value of Feature 4 is equal or greater than 1.
3. The third class of users are the ones who are looking for interchange stops. In this case if the fraction of requested stops is equal to 0.5, than the number of near stops is 0. While if the user has selected more than 2 stops, the value of the ratio of requested unique stops is less than 0.5. The number of near stops should be also 0.



ip inet	time time without	stop numeric	bus numeric
46.237.134.37	10:42:50	51056	
46.237.134.37	10:46:01	47551	

Figure 4.13: Bus-to-Bus transfer

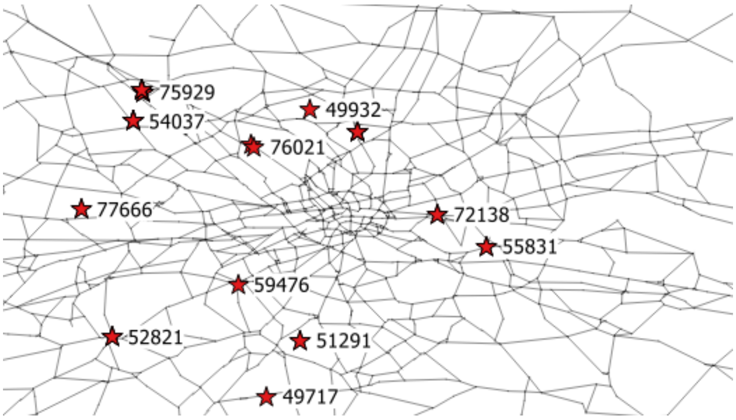


Figure 4.14: Outliers

4. The last class of users is the group of requests that cannot explain any user behaviour so that it is called "outliers". This is the case of multiple requests for live arrivals from any stops located everywhere in the city.

4.4.2 The features

To compute the features of each observation, the entire dataset is imported into Matlab2014a where each array groups the consecutive requests made by the same user. The result is a matrix where each row identifies an observation and the each column the feature. The first two features are called respectively *bus* and *stop*. These variable are computed as the inverse of the unique buses and stops the LiveBus Arrivals user has requested. This choice allows us to obtain a first set of variables that range from zero and one. The third feature is here called *AVG delta time* and it is computed as the average time difference of each request. This variable provides information on the frequency a certain user sends multiple requests for live bus arrivals. Finally the last feature is called *stops within 500m* because it quantifies the number of stops requested that are located within 500 meters from each other. To compute this feature, the unique number of stop are selected and the geographical coordinates are associated to them. A matrix of distances between each couple of coordinates allows to count the number of stops that are located within 500 meters from each other. This feature allows to identify the requests that are related to stops located within 500 meters or stops that are far from each other. As explained in section 4.3.2 with an example, when stops are located far from each other, it is possible that the users are looking for an interchange stop. In case of nearby stops the user is probably searching for different travel solutions or is checking for more precise information on the bus direction. A set of variables are computed to classify different type of requests made by the users. To this aim few main categories of requests are identified and all features are standardised such that their values range between 0 and 1.

4.4.3 Training Set

The training set consists of four elements that characterize each of the four classes. The first four elements of the training set show that only one stop has been requested by the user (Feature 2) with averages of the time elapsed between consecutive requests included between 25 and 723 seconds (Feature 3). The second class of the training set consists of users who select three different stops (Feature 2) that are located within 500 meters from each other (Feature 4). In the third class the users selected two stops (Feature 2) that are far from each other (Feature 4). Finally Class 4 trains the group of requests that are classified as outliers because a large number of stops have been requested (Feature 2).

4.5 Results

The multiclass SVM for the classification of LiveBus Arrivals users is implemented by using LibSVM 3.19 (Chang, 2011 Chang and Lin (2011)) with Matlab2012a interface.

4.5.1 Classification

The algorithm that is implemented for SVM multilevel classification is the Sequential Minimal Optimization (SMO) ?. The method is the one-against-the rest binary

Table 4.2: Training Set

Class	bus	stop	AVG delta time	stops within 500m
1	0.50	1.0000	25.6	0
1	0.00	1.0000	723.2	0
1	0.00	1.0000	38.8	0
1	0.00	1.0000	49.7	0
2	0.00	0.3333	40.0	3
2	1.00	0.3333	31.7	3
2	0.33	0.3333	5.0	3
2	0.50	0.3333	163.6	3
3	1.00	0.5000	48135	0
3	0.00	0.5000	702.0	0
3	0.00	0.5000	3227.0	0
3	0.00	0.5000	3619.0	0
4	0.00	0.0006	4.7	2015
4	0.00	0.0002	4.8	11486
4	1.00	0.0009	42.8	560
4	0.00	0.0002	4.8	11860

classifier proposed by Vapnik (1995). The strategy consists in generating k classifiers for the k classes and then solve the k binary classification problems. Specifically, four quadratic programming problems are solved where each one consists in separating one class from all the other classes. The model is trained to obtain a first prediction of the test set labels. The two LibSVM 3.19 functions, *svmtrain* and *svmpredict*, are implemented for to train and classify the dataset. A sigmoid/poly kernel parameter and a Gaussian radial basis function are selected. This classification has an accuracy of 49.56% and the results are shown in Figure 4.15 where the bar indicates the predicted labels assigned to the each of the four classes.

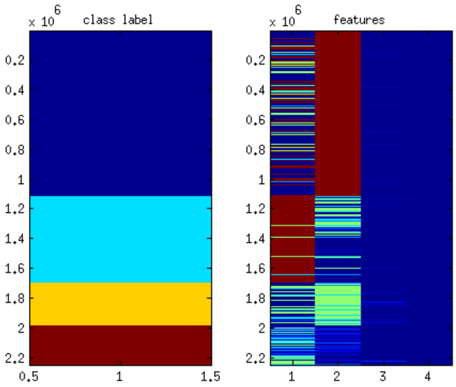


Figure 4.15: First classification results

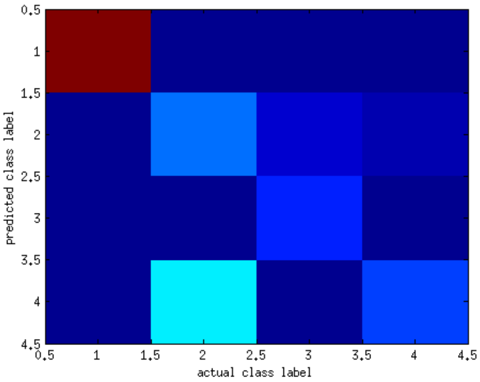


Figure 4.16: Confusion matrix

Next a 3-fold cross-validation is performed to estimate the parameters of the model yielding higher prediction accuracy of the labels. The cross validation method consists in separating the entire dataset in three equal groups of elements and, at each iteration, the 1/3 of the dataset become the validation set while the remaining part is the training set. The accuracy of the classification is defined as the fraction of the number of correctly predicted data over the number of observations of the testing data. The estimated parameters of the model, resulting from the cross-validation are: C=64 and gamma=0.088388. These parameters yield to an accuracy of 79.241%. The confusion matrix (Kohavi and Provost (1998)) is represented in Figure 4.16. The diagonal elements of the matrix represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. As expected the values of the first prediction model are rather different from these resulting from the second classification that uses the estimated parameters using the cross validation.

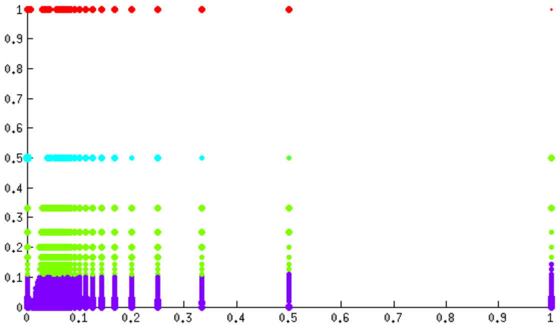


Figure 4.17: Predicted labels

Figure 4.17 shows the decision values in a two dimensional space. The four classes are completely separate from each other meaning that the classification yields to good results. Finally Figure 4.18 shows the support vectors of the four classes. The

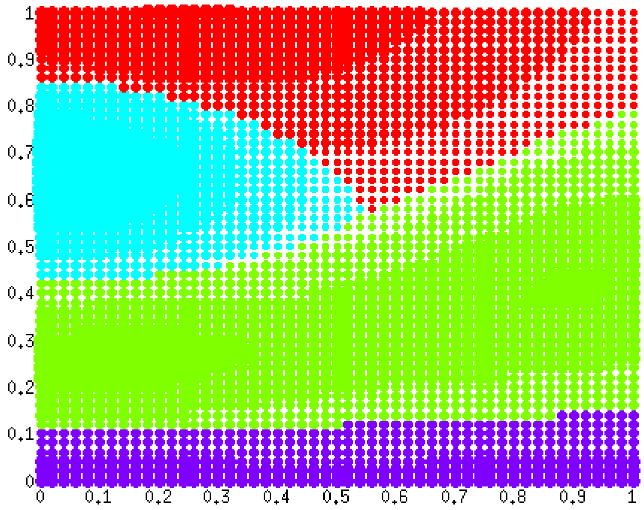


Figure 4.18: SVM support vectors

class on the top of the plot (red) is the set of repeated requests (Class 1). That is users who continuously update the system to get information on the bus arrival at one single stop. This behaviour could be associated with some variables of waiting time at stop. The area included between the values of the y-axes 0.11 - 0.43 (green) represents the class of users who request information on arrivals at more than one bus stop and these stops are located within 500 meters from each others (Class 2). This is the case when the user look for information on nearby stops, for instance when it is not clear the bus direction or when the aim is to find the stops from which the desired bus stops by. This area together with the previous are the largest meaning that most of the requests belongs to these two classes. The smaller area include between the values 0.43 - 0.87 of the y-axes, is the class of the requests for buses that are located at a distance that is greater than 500 meters from each other (Class 3). These users are the ones who are looking for information on buses to interchange. For instance a LiveBus Arrivals user is waiting for a bus and looks for arrivals at another stop to board a different bus. This class can provide important information on the stops where the bus users are most likely to interchange. Finally, the last class on the bottom of the plot (blue) is the one of the outliers (Class 4). So no explanation can be provided on user behaviour.

In this last section the results of the classification are analysed with the objective to validate the City of London Travel to Work CENSUS 2011. This survey provides the number of city workers who travel by bus by place of residence. The survey consists of 22.000 interviews made to commuters who use buses. Inflows and outflows are available for each borough of London.

The first step of the analysis consists in the identification of the peak hours that are the time ranges during which the higher number of requests are sent by the users. The aggregation of the different classes of LiveBus Arrivals requests by London borough is performed by running a spatial query on the set of labeled observations over each borough. The query is run for each class of observation, exluding the class of the outliers. The result of the query is the number of requests aggregated for each borough. By plotting the requests by distributing them for time ranges, it is possible to identify three main peak hours (8-9 AM, 2-3 PM and 6-7 PM). All the three classes have similar peak hours.

Obviously, Class 1 of repeated requests has higher peaks between 8-9 AM and 1-2 PM, meaning that users knows which stop they want to board and impatiently wait for the bus to arrive at the stop. Class 2 of the requests for near by stops has the higher peak in the evening (6-7 PM) when users have more time to walk around the city. Finally, Class 3 that identifies interchange stops has only two peaks in the morning and afternoon and no requests in the evening.

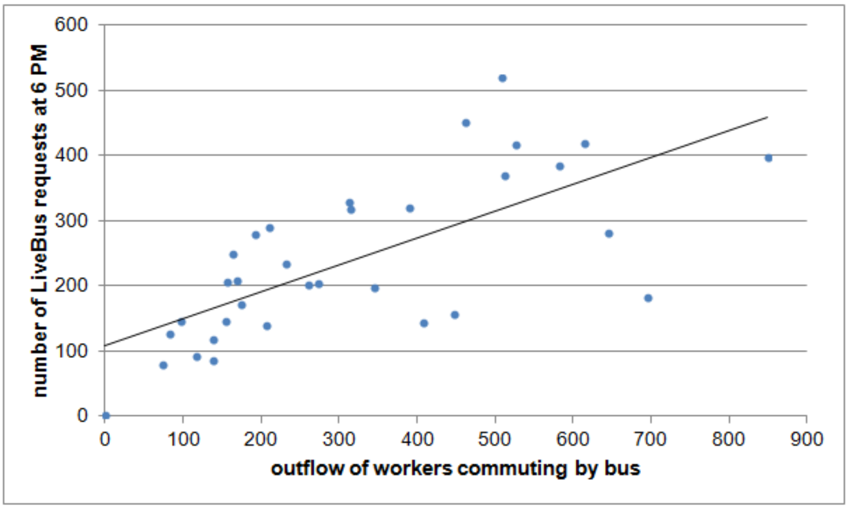


Figure 4.19: Class 1: correlation between outflows and number of requests

To evaluate the possibility to use LiveBus Arrivals to estimate the number of commuters that travel by bus in the city of London, the aggregated number of requests are weighted with the number of residents of each borough. The weight is obtained as the ratio between the residents of the borough and the total population of London. Clearly all borough are considered in the analysis so that the total number of residents corresponds to the population of London.

Figure 4.19 provides an example of what has been done for each of the three classes and peak hours time ranges. The correlation between the weighted number of

Table 4.3: Correlation between weighted number of requests and travels by bus

Time range	Class 1	Class 2	Class 3
8-9 AM	64.7%	62.5%	62.4%
1-2 PM	62.9%	61.6%	62.0%
6-7 PM	69.9%	68.8%	-

requests in each borough and the outflows of workers commuting by bus from each borough have been computed. The graph shows clearly that there is high correlation between the set of variables (70%).

Moreover this correlation is higher for all classes and time ranges as showed in Table 4.3 where the lowest correlation is 62% for Class 2 at 2 PM. As said before, the Class 3 has no requests for real time arrivals at 6 PM.

4.6 Conclusion

This work presented a new deployment of the information on the requests on real time arrivals made by the users of the bus service in the city of London. First of all the raw data have been preprocessed by extracting the number of bus and stop on which the user requested information on real time arrivals. The preprocessed data have been analysed by taking few samples and different classes of users' behaviour have been described and represented in the GIS.

The 10% of users frequently update their request for live arrivals at a bus stop. A real time information system could include the possibility to keep the system updated, for instance with a clock which the user can activate or deactivate. Furthermore the information gathered from repeated requests could allow to develop an indicator of perceived waiting time that depends on the number of requests sent by each user. For instance if the time elapsed between the first and the last request is greater than 10/15 minutes and the user has updated the system every second, the time elapsed between the first and the last request could be used as an estimation of waiting time at stop. The value of the indicators can be then related to weather and traffic conditions or bus frequencies at stops to detect if there are variables that have an influence on perceived waiting time. To this aim two different time periods could be compared: a period of regular traffic and an overcrowding period such as the one during which the Olympic Games took place.

In some cases public transport passengers check for arrivals at both directions of the same stop. This probably means that it is not always clear to them which bus route direction they should board to arrive at destination. A real time public transport information system could include an indication referred to signals applied to the stop that allows to instantly recognise the bus route direction the user is looking for. In order to be able to indicate the route direction, the system could require the user to type the stop of departure and arrival. Therefore information on origin and destination would be provided directly by the user thus allowing to estimate the public transport OD matrix.

Users sometimes request for arrivals at a bus stop and for a bus line. This type of information allows to determine the popularity of a bus route or stop especially with reference to the use of the Live Bus Arrivals service or to determine if users prefer less crowded stops. Moreover the description of some examples of bus-to-bus transfer has allowed to have an idea on how passengers use the Live Bus Arrivals service to gather information on bus arrivals at the transfer stop. They probably adjust their travel plan while they are travelling and choose the route or the alighting stops for transfer basing on real time information. For instance it seems from the examples on bus-to-bus transfer that passengers sometimes walk to another stop instead of boarding at the interchange stop. Hence, following Trozzi et al. (2013) results, future work could concentrate on how live information may influence equally attractive stops and lines upon departures. For instance, given two equally attractive stops, the study could explore how users' route choice change based on LiveBus Arrivals information.

Successively, a set of features that characterize fours different classes of users have been computed to run a multiclass SVM. The result of the classification of the entire dataset yield to a reasonably good accuracy of 79,2%.

Once having classified the LiveBus arrivals users, the last part of this work is focused on the evaluation of the possibility to use these requests as a proxy for the estimation of the number of outflows of workers commuters who travel by bus. Three main peak hours have been identified by plotting the number of requests for each hourly time range. The peak hours were consistent for all three classes of users since they reflected the most probable behaviour of bus users during different time of the day. For instance, in the morning users know the number of stops the intend to board so that they send repeated requests for real time arrivals at the same bus stop (Class 1). Finally, the correlation between the number of requests for live time arrivals in each borough of the city of London and the number of outflows of workers commuters by bus from each borough has shown, in all cases, a high value of the correlation. Therefore it is possible to conclude that requests for real time arrivals may be taken into account to update the census data on commuters by bus. Several possible future analysis could be carried out using the same database. For instance, a more detailed analysis of the three main classes of users could be performed to understand the users' behaviour in bus stop choices.

5 Classification of accidents involving two wheelers vehicles

5.1 Introduction

This chapter tackles the problem of the classification of two wheelers vehicles' accidents. The identification of the type of accident is one of the requirements for the implementation, at European level, of an integrated emergency service, called eCall. This service consists in a trigger system that sends the information to the nearest emergency center that can localize the vehicle and can send the most appropriate rescue service. The European Commission strongly supports the eCall, for instance, by funding several initiatives to help Member States to update their emergency systems or by introducing regulations for vehicle manufacturers to introduce eCall devices.

In 2012 around 28,000 people were killed and more than 1.5 million injured in 1.1 million collisions on the roads in Europe. In addition to the tragedy of loss of life and injury, this also carries an economic burden of around EUR 130 billion in costs to society every year across Europe. It is estimated that 112 eCall can speed up emergency response times by 40% in urban areas and 50% in the countryside and can reduce the number of fatalities by at least 4% and the number of severe injuries by 6%. This equates to an average cost per fatality across Europe of € 225251.00 (Source: The true Cost of Road Crashes International Road Assessment Program).

The implementation of the pan European emergency service (eCall) is the focus of the HeERO (Harmonised eCall European pilot) project. The second phase of the project dealt with a set of tests and analyses aimed at exploring the possibility to extend the system to the two wheelers vehicles.

The activities carried out in the context of HeERO are briefly described to explain how the system should work in Europe. Successively, the main findings of the HeERO phase 2 project with reference to two wheelers vehicles, are presented in order to introduce the main characteristics and differences existing between the implementation of eCall for four and two wheelers vehicles. Among the issues that have to be solved to obtain a reliable system, the problem of the accidents detection for two wheelers vehicles is selected and introduced together with the analysis of the existing literature and past projects results. Finally, based on the described literature, a discussion on the activities to be carried out and the methodologies that could be selected to classify accidents involving two wheelers vehicles, is presented in the conclusion of the chapter.

5.2 eCall

HeERO is a two phases European Union project aimed at facilitating the implementation of the European emergency service (eCall). The project was mainly focused on eCall for four wheelers vehicles that will be mandatory, for all European Member States, by October, 2017. The eCall consists of a system that allows to send a minimum set of information (Minimum Set of Data, MSD) such as the unique identification number of a vehicle, its position, etc., to the nearest emergency service (Public Safety Answering Point, PSAP). The voice communication between the occupant of the vehicle and the call center is also possible because the eCall device has a SIM card. The call is activated by pushing a button installed into the vehicle or it can be activated automatically in case of an accident. In case of automatically activated calls, the triggering system is usually connected to the sensors of the airbag.

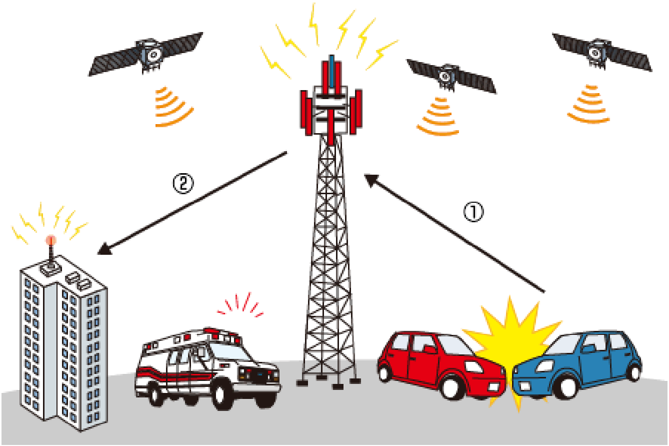


Figure 5.1: eCall system

The eCall chain involves several actors that have to cooperate to make the system to work. The PSAPs have to be updated such that they are able to receive eCalls. This means that the operators have to be trained, the software of the emergency center has to be updated and the architecture of the eCall system has to be decided at Member State level. Mobile Network Operators (MNOSs) have to implement the eCall Flag so that the eCall is identified in the network and Original Equipment Manufacturer (OEMs) have to provide In-Vehicle Systems (IVS) for eCall. There are also additional issues dealing with cross boarder situations where the mobile network is covered by foreign operators (i.e. at the border areas the MNO from Luxembourg could cover the Belgian territory or vice-versa). In this case is not clear where the eCall should be addressed. Low mobile network coverage, especially in rural areas, is a problem because it directly affects the reliability of the eCall system.

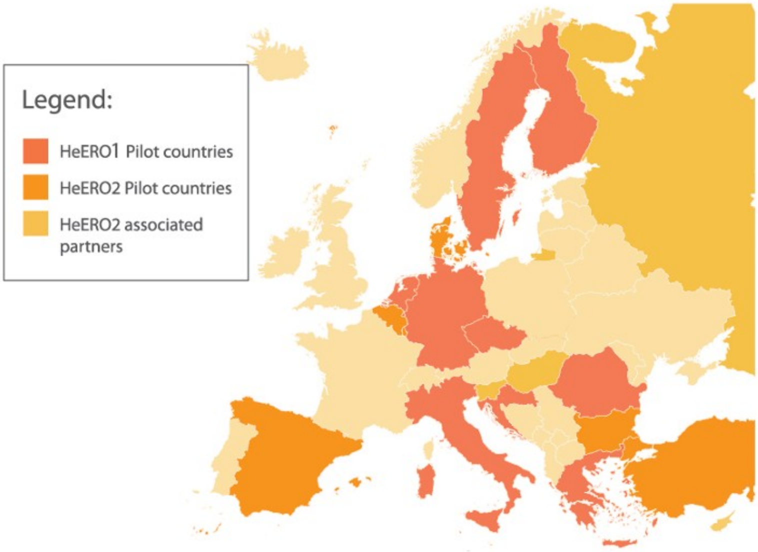


Figure 5.2: HeERO countries

In November 2013 the European Economic and Social Committee (EESC) commented on the eCall proposals to take effect by underlining that *the provision of eCall technology on motorcycles and other powered two-wheel vehicles is not mentioned. As the risk of death and injury to drivers and passengers on these types of vehicles is a significant problem, the EESC urges manufacturers and Member States to extend eCall to powered two wheelers as soon as possible.*

The number of motorcycles in possession for country/region, shown in Table 5.1, gives an idea of the importance of having an emergency system for two wheelers vehicles.

Table 5.1: Global numbers of motorcycles in possession by country/region. Source: Sekine et al. 2014

Country/region	Number
Italy	8,610,000
Spain	4,070,032
France	3,439,417
United Kingdom	1,468,800
Netherlands	1,269,433
Switzerland	833,891
Austria	712,635
Poland	2,102,175
Czech Republic	944,171
Turkey	2,527,190

The possibility to introduce the eCall system for two wheelers vehicles has been analysed in the context of the Spanish pilot of HeERO project phase 2. The Spanish team proposed a wearable solution for two wheelers vehicles and studied and tested the advantages and disadvantages of having in-vehicle and wearable systems. They also carried out a small survey aimed at evaluating the interest of users in an emergency service and their willingness to pay for it. It came out that Spanish users are available to pay up to EUR 100 for an eCall system installed on their motorbikes.

5.2.1 eCall for two wheelers vehicles

The existing technologies for powered two wheelers eCall systems are currently less mature comparing to the devices available for four wheelers. The reason is that the dynamic of motorbikes' crashes is more complex comparing to cars. Secondly, cars can use airbag's sensors to activate the eCall, while new systems have to be introduced for motorbikes thus making the vehicle more expensive. Furthermore, vehicles manufacturers prefer not to allow external systems to interface with the vehicle electronics so that research is mainly performed internally with few resources. External systems are, in most of the cases, not very sophisticated.

A summary of the main differences between cars and two wheelers emergency calls is provided by Matthias Mörbe from Bosch Engineering GmbH (see Table 5.2).

Systems that are installed on the motorbike do not allow to determine the position of the driver if the crash provokes its separation from the vehicle. On the other hand, wearable systems do not allow to detect the type of collision when, for instance, the accident is caused by an obstacle. One of the most important information is the

Table 5.2: Comparison between eCall for two wheelers vehicles and cars

Common	Uncommon
Rescue chain	Separation of vehicle and driver
Billing system	Zero speed in safe area
National and international standards of communication	Sensor types
Forces to vehicle and driver are the same if they are combined	Stability criteria
Power supply requirement	Accident detection
Infrastructure and business case	Location (off road)
	Forces to vehicle and driver after separation
	Ambient noise for voice communication
	Accident recognition
	Crash sensors at all sides
	Population and hours of use

number of occupants which is in general very difficult to determine in two wheelers vehicles, while airbag sensors can provide this type of information. In case of wearable eCall solutions, all passengers should wear a system with eCall capabilities to detect the number of passengers. Since the VIN (Vehicle Identification Number) is embedded into the electronics of the motorbike, in order to link the passengers to the vehicle number, it is necessary to link the vehicle to the wearable eCall system (i.e by introducing additional sensors).

Finally, regulations on environmental, electrical, safety and stress tests, functional safety (ISO 26262 or ISO 61508), reliability of the communication protocol and standards for front end of two wheelers vehicles (i.e. incident detection) need to be defined.

Overall, the pilot carried out in Spain, highlighted the need for a robust accident detection system based on several sensors to be installed on the vehicle and that is combined with a wearable system. Furthermore, several questions arose after the test such as on which type of information (Minimum Set of Data, MSD) has to be sent to the emergency service, on the frequency which these information must be kept updated and on which information on the crash severity are needed by the emergency center (Alfonso Brazalez, CEIT). The following is a list of open questions on eCall for two wheelers vehicles:

- Information on the position and orientation of the driver to be included in the MSD:
    1. Is it useful to differentiate between the position of the driver and the position of the motorbike in the MSD?
    2. Is it useful to communicate the orientation of the driver and the motorbike in the MSD?

    3. Is it useful to communicate the distance between the driver and the motorbike in the MSD?

- Frequency and duration of updates:

    1. How often the positions of the driver have to be updated?
    2. How often the positions of the motorbike have to be updated?
    3. How long should be the duration of the update on the positions of the driver?
    4. How long should be the duration of the update on the positions of the motorbike?

- Information on passengers:

    - Is it useful to know if there is any passenger on the motorbike?

- Information on the severity of an accident:

    - Is it useful to know a crash severity index given by the data registered in the motorbike (i.e. an index based on historical accident data)?

These questions and other issues are the topics of discussion of a technology cluster that has been recently created in the context of the preparation of a new European Union proposal on the implementation of eCall. The team is formed by the representatives of the main motorcycles industries, who meet on a regular basis with the aim to develop an eCall system for two wheelers vehicles.

5.2.2 Existing eCall systems for two wheelers vehicles

The different solutions for two wheelers eCall can be classified in three main types: wearable, apps and in-vehicle solutions. Here one example for each category is described generally since the details on the algorithms deployed are not publicly available.

The system that was tested in the Spanish pilot consisted of a helmet provided by the Spanish manufacturer NZI. The helmet is able to detect an accident if it is dropped on the ground or the bike is flipped in the parking lot. The solution was available only for the test that was performed on two wheels rally racers. However, the test did not provide enough data because of the limited number of accidents and because of the different dynamics of the accidents.

The market offers several apps for motorbikes crash detection. For instance, EatSleepRIDE has been developed in conjunction with the University of Toronto. The technology is called CRASHLIGHT and it consists in a algorithm able to detect an accident. A text message is then sent to the registered contacts. The algorithm uses phone's accelerometer, gyroscope and other sensors to determine if an accident has taken place.

Figure 5.3: Connectivity Control Unit (CCU) for two-wheelers

The Connectivity Control Unit (CCU) produced by Bosch has been presented in November 2014. The CCU is *able to recognize and evaluate critical driving maneuvers on the basis of sensor information from the vehicle and rider as well as with the help of a so-called crash algorithm* (Figure 5.3). Once the system has detected the occurrence of an accident, it transmits the information on the location, time and type of vehicle to the nearest emergency center. Voice connection with the nearest call center is also possible.

5.2.3 Types of two wheelers vehicles accidents

The main difficulty of the classification activity of two wheelers accidents relies on data analysis. Specifically, it is difficult to identify relevant patterns that allow to detect an accident. A current practice consists in the analysis of images without using any automatic procedure. Furthermore crashes involving two wheelers are often the consequence of a chain of interrelated events so that it is not always possible to identify its causes.

One of the most complete research on accidents involving two wheelers is the MAIDS study (Motorcycle Accidents In Depth Study, *www.maids-study.eu*). The three years project was leaded by ACEM, the European Association of Motorcycle Manufacturers and took into account 921 motorbikes crashes, occurred between 1999 and 2000 in 5 OECD countries, and analysed their characteristics. The final report presents a classification of accidents that is based on an in-depth analysis of

Table 5.3: Classes of accidents - MAIDS study

	Classes of accidents involving PTW (two wheelers vehicle)
1	Head-on collision of PTW and other vehicle (OV)
2	OV into PTW impact at intersection - paths perpendicular
3	PTW into OV impact at intersection; paths perpendicular
4	OV turning left in front of PTW, PTW perpendicular to OV path
5	OV turning right in front of PTW, PTW perpendicular to OV path
6	PTW and OV in opposite direction, OV turns in front of PTW, OV impacting PTW
7	PTW and OV travelling in opposite directions, OV turns in front of PTW, PTW impacting OV
8	PTW turning left in front of OV, OV proceeding in either direction perpendicular to PTW path
9	PTW turning right in front of OV, OV proceeding in either direction perpendicular to PTW path
10	PTW overtaking OV while OV turning left
11	PTW overtaking OV while OV turning right
12	OV impacting rear of PTW
13	PTW impacting rear of OV
14	Sideswipe, OV and PTW travelling in opposite directions
15	Sideswipe, OV and PTW travelling in same directions
16	OV making U-turn or Y-turn ahead of PTW
17	Other PTW/OV impacts
18	PTW falling on roadway, no OV involvement
19	PTW running off roadway, no OV involvement
20	PTW falling on roadway in collision avoidance with OV
21	PTW running off roadway in collision avoidance with OV
22	Other PTW accidents with no OV or other involvement
23	PTW impacting pedestrian or animal
24	PTW impacting environmental object

the dynamics of the accident including an encoding process of each event (see Figure 5.3). The classification is based on different types of two wheelers vehicles using the *Common Methodology for on-scene in-depth motorcycle accident investigations* (OECD, 2001).

The topic has been studied by several other European Union projects related to two wheelers safety (SAFERIDER, PISa, MOSAFIM, etc.). For instance, the European project 2BESAFE (2-wheeler behaviour and safety) run from 2009 to 2011 and identified four main types of accidents that involve PTW (Table 5.4).

Table 5.4: Classes of accidents - 2BESAFE

1	A driver accelerates to overtake another vehicle
2	A driver finds a stationary obstacle and activates the breaks in low speed
3	A driver finds a moving obstacle and activates the breaks in high or low speed
4	A driver move to the opposite direction of traffic

Finally, the most common configurations related to fatal and serious motorcycle accidents have been identified between 2006 and 2007 in the context of an European Union project called TRACE (Traffic Accident Causation in Europe):

1. Single accidents due to visibility problems (i.e. due to terrain profile) and inadequate speed
2. Accidents between passenger car and motorcycles:
   - Front-side accidents in rural and urban junctions between motorcycles and passenger cars due to rider parking or getting into a carriageway from another road
   - Side-side accidents in rural and urban non junctions between motorcycles and passenger cars due to driver inexperience and inadequate speed or traffic violation
   - Rear-end accidents in rural and urban non junctions between motorcycles and passenger cars due to i.e. absent-minded, turn incorrectly, overtake illegally, not keeping safe distance

5.3 Literature on accident classification

Most of the literature on crash detection is focused on the classification and identification of cars accidents. For instance, Singh (2010) divides the crash detection algorithms for cars in two main categories: speed dependent and crash dependent. He also underlines how smart airbags algorithms that identify accidents based on airbags sensors, are difficult to implement. So that he proposes the use of Machine Learning techniques such as Hidden Markov Chain (HMM) and Support Vector Machines (SVM). Where HMMs is a method that consists in representing the probability distribution of an observed time series. He found that HMM are more efficient compared to SVM because of the possibility to take into account the time variable.

Machine Learning techniques have been used to detect two wheelers vehicles crashes. A recent study presented by Montella et al. (2012) uses classification trees analysis and rules discovery to classify accidents involving motorcycles. The classification tree analysis aims to find the predictive structure of the problem and to understand which are the predictors that influence most the response variables. Tree methods aim to partition the observations into homogeneous groups of classes. The authors start with the identification of four main variables that are important to describe an accident:

1. severity
2. crash type
3. involved vehicles
4. alignment to identify crash location on the road network

The dataset used for the analysis consisted in 19 categorical variables provided by Italian National Institute of Statistics (ISTAT): area, road type, lighting, weather, pavement, driver PTW gender, driver PTW age, driver PTW outcome, Vehicle B driver gender, Vehicle B driver age, Vehicle B driver outcome, pedestrian gender, pedestrian age, pedestrian outcome, alignment, involved vehicles, PTW type, crash type, and severity. The method consists in the evaluation of the degree of influence of each variable of the dataset on each of the response variables (the class of accident). The results were consistent with previous studies but were suffering from an extreme risk of type I error, which means that there is a high probability of incurring in false positives. As such the method is not suitable for eCall that requires high reliability of the results.

An analysis of Power Two Wheelers risky behaviours has been carried out by Attal et al. (2013) using different machine learning methodologies. They used accelerometers, gyroscopes and vehicle speed sensors to classify five main driving behaviours (turn right, turn left, round about, straight and stop). They implemented five machine learning algorithms: Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs), Support Vector Machines (SVMs), Random Forest (RFs) and k-nearest neighbor algorithm (k-nn). All the classification algorithms led to an accuracy greater than 80% but the most efficient classifier was the continuous HMM. The classifier has provided the best results because it allows to take into account the temporal evolution of the events. Similarly, Vlahogianni et al. (2011) proposed a methodology to distinguish between different driving behaviour using naturalistic data and wheel speed, acceleration, throttle, steering rear breaking, etc. They identified seven classes of accidents (see Table 5.5) and developed a Multi-Layer Perceptron classifier (MLP) to model the interrelation between the measured parameters and the riding behaviour.

Table 5.5: Classes of PTW accidents - Vlahogianni et al. (2011)

1	Moped braking and moving on the right to avoid opposing vehicle
2	Braking due to pedestrians in high grade
3	Moped moving on the left to avoid fixed object
4	Entering sharp turn when interacting with opposite lane's traffic
5	Braking and moving on the right after having overtaken vehicles
6	Overtaking more than one vehicle
7	Moped moving to the left to avoid stationary object

Most importantly, they found that steering, throttle, brake activation and wheel speed are the parameters that allow to identify the probability of an accident. Vlahogianni et al. (2014) propose a methodology based on Bayesian Networks. The methodology consist in the assumption that, in order to detect incidents, it is necessary to relate specific driving behaviour to the accident. The approach is articulated in two steps: first of all a Bayesian classification algorithm is implemented with the aim to group critical actions during driving and after the occurrence of an accident; secondly the identified actions are linked to the type of accident. The result of the analysis is the identification of four main classes of accidents:

1. Interaction with moving obstacle
2. Interaction with stationary obstacle
3. Interaction with opposing traffic
4. Overtaking

Other studies are focused on the development of combined technologies to detect the accidents. For instance, Boubezoula et al. (2013) propose a fall detection algorithm for PTW that allows to activate, in case of fall, an airbag that is installed on a jacket. They identify four fall scenarios (fall in curve, fall on a slippery straight section, fall with leaning of the motorbike, fall in a roundabout) and they collect data thanks to accelerometers and gyroscopes. The first innovation consists in a filtering algorithm for data cleaning and noise reduction. Successively the fall detection algorithm uses the filtered acceleration and angular velocities to check if a threshold is crossed and, based on this information, it decides whether to activate an alarm or not. The main advantage of this approach relies on the possibility to use motorbikes existing instrumentation. The implementation of a Bayesian classification algorithm is also described in Parviainen et al. (2014). The classifier is able to recognise four types of two wheelers vehicle events: Static, Moving, ZeroG and Peak. The main innovation of this work relies on the use of three different measurement units placed on the helmet, on the torso of the driver and on the rear of the motorbike.

5.4 Conclusion and future work

The emergency call for two wheelers vehicles should become one of the priorities of the European Commission and should proceed in parallel with the implementation of eCall for four vehicles. However the process development of eCall for two wheelers needs to consider several aspects such as the dynamics of the crash which is more complex comparing to cars, the difficulty to develop an automatic recognition methodology that allows to categorize accidents and their different levels of severity. For instance, the most frequent situation is when the driver is separated from the vehicle. This scenario implies a decision on where to place the GPS device that is used to locate the accident but it also involves decisions regarding the best solution on where to place a microphone so that a voice connection can be established.

Furthermore, there are many possible types of collisions and consequences that have been partially by several studies. It is necessary to take decisions on the level of details the class of accidents has to be described for the purpose of eCall. Some of the classifications reported in the literature are general and consist of only three / four classes, while other sources describe more detailed in the dynamics of the accidents such as the classes of the MAIDS study. The process of identification of the classes of accidents should begin from the comparison with the existing literature and should be the result of the discussion performed during round tables participated by the technical experts of the vehicles manufacturing industries. The representatives of the emergency services should also be involved in the discussion in order to ensure that the information sent (i.e MSD) is complete enough to be able to identify the type of rescue service that is required.

With reference to the specific issue of accidents' detection, the analysis of the existing literature has shown that Machine Learning techniques are the most appropriated to classify the type of crash. In particular HMM, that is able to consider the evolution of the events on time, has provided good results in a least two different formulations of the classification problem. While Bayesian classification algorithms have been deployed to classify the PTW driving situations using a set of measured parameters that allow to compute the probability of an accident.

Future analyses should define the requirements for the fall detection algorithm. Specifically, additional types of sensors could be installed on the vehicle or on a wearable system to provide new measures that may be important for the detection of an accident (i.e. driver separated from the vehicle, position of the driver after the accident, whether a voice connection has been established, etc.). New types of data could be collected by providing new info mobility services to the driver that could allow to study the his behaviour. Once new types of data will be available to detect the type of accident, factor or component analysis could be used to identify the reduce and create more appropriate classes of accidents or to identify unobserved variables. As a second step, Bayesian classification and Hidden Markov Models could be implemented to evaluate the methodology that is more appropriate in terms of accuracy of the results and of computational efficiency.

6 Conclusion

This work has presented a set of methodologies that allow to classify and analyse Intelligent Transport System data with the aim to extract information on users behaviour. The main problem related to Intelligent Transport System data analysis is related to the design or selection of the most algorithm. The existing clustering algorithms may be not the most appropriate for the analysis of large database.

The selection of the most appropriate algorithm depends on the structure of the dataset and on the objective of the analysis. The described algorithms of the unsupervised type have been proposed about fifty years ago and they are still the most used in the literature. This demonstrates how difficult can be to design a classification algorithm that has a general formulation so that is able to solve several types of problems. Recently new classification algorithms that are able to handle large database have been proposed. However, data preprocessing allows to obtain more information on the underlying structure of the dataset and it also allows to efficiently deploy existing algorithms.

The evaluation of clustering results is an issue, especially for big database and spatial data. For instance, if the analysis consists in the implementation of a hierarchical cluster algorithm, the analysis of the dendrogram become very complex. To improve the solution and the similarity within the clusters, a possibility is to introduce a constraint that creates a stop rule. The two methodologies, hierarchical and partitioning analysis, allows to easily introduce a constraints and to deploy different types of metrics depending on the situation.

The results of the classification of Intelligent Transport System data can bring useful information on drivers and transport users behaviour. For instance, the cluster analysis of observed routes made by drivers and the comparison with modelled routes allows to identify a relation existing between the reliability of a route and the drivers' route choice behaviour. A penalty could be introduced in a traffic assignment model to penalise the alternatives that have lower reliability so that the model would be more realistic.

Bibliography

M. Abdel-Aty, R. Kitamura, and P. Jovanis. Using stated preference data for studying the effect of advanced traffic information on drivers' route choice. *Transportation Research Part C*, 5(1):39–50, 1997.

J. Abonyi and B. Feil. *Cluster Analysis for Data Mining and System Identification.* Birkhauser, 2007.

A. Akgun, E. Erkut, and R. Batta. On finding dissimilar paths. *European Journal of Operation Research*, 121:232–246, 2000.

M. Asif, C. Goh, A. Fathi, M. Xu, M. Dhanya, N. Mitrovic, and P. Jaillet. Spatial and temporal patterns in large-scale traffic speed prediction. *IEEE Intelligent Transportation Systems Conference,*, 2012.

F. Attal, A. Boubezoul, L. Oukhellou, and S. Espié. Riding patterns recognition for powered two-wheelers users' behaviors analysis. In *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013.

K.W. Axhausen, A. Zimmermann, S. Schonfelder, G. Rindsfaser, and T. Haupt. Observing the rhythms of daily life: A six-week travel diary. *Transportation*, 29: 95–124, 2002.

K.W. Axhausen, S. Schönfelder, and J. Wolf. Eighty weeks of gps traces, approaches to enriching trip information. In *Transportation Research Board 83rd Annual Meeting*, volume 178, 2004.

J. Azevedo, M.E.O Santos Costa, J.J.E.R. Silvestre Madeira, and E.Q. Vieira Martins. An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69(1):97 – 106, 1993.

R.A. Becker, R. Caceres, K. Hanson, J.M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. A tale of one city: Using cellular network data for urban planning. *Pervasive Computing, IEEE*, pages 18–26, 2011a.

R.A. Becker, R. Caceres, K. Hanson, J.M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. Route classification using cellular handoff patterns. *UbiComp'11*, pages 123–132, 2011b.

S. Bekhor, M.E. Ben-Akiva, and M.S. Ramming. Evaluation of choice set generation algorithms for route choice models. *Annals of Operations Research*, 144:235–247, 2006.

G. Beni and J. Wang. Swarm intelligence in cellular robotic systems. In Sandini G. & Aebischer P. Dario P., editor, *Robots and Biological Systems: Towards a New Bionics?*, volume 102 of *NATO ASI Series*, pages 703–712. Springer Berlin Heidelberg, 1993.

J.K. Bernard, M.and Hackney and K.W. Axhausen. *Correlation of Link Travel Speeds*. ETH Eidgenössische Technische Hochschule Zürich, Institut für Vekehrsplanung und Transportsysteme, 2006.

A. Boubezoula, S. Espiéb, B. Larnaudiec, and S. Bouazizc. A simple fall detection algorithm for powered two wheelers. *Control Engineering Practice*, 21(3):286–297, 2013.

R.N. Buliung, M.J. Roorda, and T.K. Remmel. Exploring spatial variety in patterns of activity-travel behaviour: initial results from the toronto travel-activity panel survey (ttaps). *Transportation*, 35:697–722, 2000.

C.J.C. Burger. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

Dell'Amico M. & Martello S. Burkard R.E. *Assignment Problems*. SIAM, 2009.

O. Cats, W. Burghout, T. Toledo, and H.N. Koutsopoulos. Effect of real-time transit information on dynamic passenger path choice. *Transportation Research Record*, 2217:46 – 54, 2011.

J. Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370, 1987.

C.C. Chang and C.J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 109–118, 2001.

W. Chun-Hsin, W. Chia-Chen, S. Da-Chun, C. Ming-Hua, and H. Jan-Ming. Travel time prediction with support vector regression. In *Proceedings of IEEE Intelligent Transportation Systems Conference*, 2003.

R.M. Cormack. A review of classification. *Journal of the Royal Statistical Society. Series A*, 134(3):321–367, 1971.

E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

J. Dong and M. Qi. K-means optimization algorithm for solving clustering problem. In *Proceedings of the 2009 Second International Workshop on Knowledge Discovery and Data Mining*, pages 52–55. IEEE Computer Society, 2009.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification, 2nd Edition.* Wiley, 2001.

K. Dziekan and K. Kottenhoff. Dynamic at-stop real-time information displays for public transport: effects on customers. *Transportation Research Part A*, 41: 489–501, 2007.

M. Ester, H.P. Kriegel, S. Jørg, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 448–455, 2003.

R.A Fisher. On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

G. Fitzmaurice, M. Davidian, G. Verbeke, and G. Molenberghs. *Longitudinal Data Analysis.* Taylor & Francis Group, 2008.

A. Fred and A.K. Jain. Data clustering using evidence accumulation. In *International Conference Pattern Recognition (ICPR).*, 2002.

S.D. Grimshaw. Computing maximum likelihood estimates for the generalized pareto distribution. *Technometrics*, 35:185–191, 1993.

P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.

S. Hanson and J.O. Huff. Assessing day-to-day variability in complex travel patterns. *Transportation Research Record*, 891:18–24, 1982.

S. Hanson and J.O. Huff. Classification issues in the analysis of complex travel behavior. *Transportation*, 13:271–293, 1986.

S. Hanson and J.O. Huff. Systematic variability in repetitious travel. *Transportation*, 15:111–135, 1988.

N. Hardy. Provision of bus real time information to all bus stop in london. In *19th ITS World Congress*, 2012.

P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics*, 4:100–107, 1968.

B. Hellinga, H. Izadpanah, P.and Takada, and L. Fu. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. *Transportation Research Part C*, 16:768–782, 2008.

M.D. Hickman. Transit service and path choice models in stochastic and time-dependent networks. *Transportation Science*, 31(2):129 – 146, 1997.

T. Holleczek, L. Yu, J.K. Lee, O. Senn, C. Ratti, and P. Jaillet. Detecting weak public transport connections from cellphone and public transport data. In *ASE International Conference on Big Data Science and Computing*, 2014.

E.R. Hruschka. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 39 (2):133–155, 2009.

J.O. Huff and S. Hanson. Repetition and variability in urban travel. *Geographical Analysis*, 18:97–114, 1986.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Inc., 1988.

A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

O. Jan, A. Horowitz, and Z. Peng. Using global positioning system data to understand variations in path choice. *Transportation Research Record*, 1725: 37–44, 2000.

J. Jariyasunant, D.B. Work, R. Sengupta, B. Kerkez, S. Glaser, and A. Bayen. Mobile transit trip planning with real-time data. *Transportation Research Board*, 2010.

E. Jenelius and N. Koutsopoulos. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B*, 53: 64–81, 2013.

S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.

P. Jones, F. Koppelman, and J.P. Orfeuil. *Activity analysis: State-of-the-art and future directions.* Gower Publishing Co. Aldershot, England, 1990.

I. Kaparias, M.G.H. Bell, and H. Belzner. A new measure of travel time reliability for in-vehicle navigation systems. *Journal of Intelligent Transportation Systems*, 12(4):202–211, 2008.

J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Proc. IEEE Symp. Foundations of Computer Science*, 1999.

J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pair-wise relationships: Metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002.

R. Kohavi and F. Provost. *Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, volume 30, chapter On Applied Research in Machine Learning. Columbia University, New York, 1998.

H. Lee, N. Chowdhury, and J. Chang. A new travel time prediction method for intelligent transportation systems. In I. Lovrek, R.J. Howlett, and L.C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5177 of *Lecture Notes in Computer Science*, chapter A New Travel Time Prediction Method for Intelligent Transportation Systems, pages 473–483. Springer Berlin Heidelberg, 2008.

J. Lee, J. Han, X. Li, and H. Cheng. Mining discriminative patterns for classifying trajectories on road networks. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 25(5):713 – 726, 2011.

H.D. Lin, N.and Liu and C.Q. Gong. Research and simulation on drivers' route choice behavior cognition model. *International Journal of Computer Science Issues*, 9:210–216, 2012.

K.W. Löchl, M.and Axhausen and S. Schönfelder. Analysing swiss longitudinal travel data. In *5th Swiss Transport Research Conference*, 2005.

C. Maag, C.and Mark and H.P. Krüger. Development of a cognitive-emotional model for driver behavior. In J. Piotr, N. NgocThanh, R.J. Howlet, and J.C. Lakhmi, editors, *Agent and Multi-Agent Systems: Technologies and Applications*, volume 6071 of *Lecture Notes in Computer Science*, pages 242–251. Springer Berlin Heidelberg, 2010.

H.S. Mahmassani, C.G. Caplice, and Walton C.M. Characteristics of urban commuter behavior: Switching propensity and use of information. *Transportation Research Record: Journal of the Transportation Research Board*, 1285:57– 69, 1990.

G.W. Milligan. An algorithm for generating artificial test clusters. *Psychometrika*, pages 123–127, 1985.

S.A. Mingoti and J.O. Lima. Comparing som neural network with fuzzy c-means, k-means and traditional hierarchical clustering algorithms. *European Journal of Operational Research*, 174(3):1742–1759, 2006.

T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

T. Miwa, D. Kiuchi, T. Yamamoto, and T. Morikawa. Development of map matching algorithm for low frequency probe data. *Transportation Research Part C*, 22:132–145, 2012.

A. Montella, M. Aria, A. D'Ambrosio, and F. Mauriello. Analysis of powered two-wheeler crashes in italy by classification trees and rules discovery. *Accident Analysis & Prevention*, 49:58–72, 2012. {PTW} + Cognitive impairment and Driving Safety.

A. Mortenson, D. Kostelec, B. Turley, and A. Parast. Evaluating connectivity projects: using point-to-point gis routing to measure the benefits of new transportation connections. In *Transportation Research Board 90th Annual Meeting*, 2011.

M.A. Munizaga and C. Palma. Estimation of a disaggregate multimodal public transport origin-destination matrix from passive smartcard data from santiago. *Transportation Research Part C: Emerging Technologies*, (24):9–18, 2012.

D. Papinski and D.M. Scott. A gis-based toolkit for route choice analysis. *Journal of Transport Geography*, 19:434 – 442, 2011.

D. Papinski, D.M. Scott, and S.T. Doherty. Exploring the route choice decision-making process: A comparison of planned and observed routes obtained using person-based gps. *Transportation Research F*, 12(4):347–358, 2009.

R. Parthasarathy, D. Levinson, and H. Hochmair. Network structure and travel time perception. *Journal of Transportation Research Board*, 2013.

J. Parviainen, J. Collin, J. Pihlstöm, J. Takala, K. Hanski, and A. Lumiaho. Automatic crash detection for motor cycles. Technical report, Tampere University of Technology, 2014.

E.I. Pas. Flexible and integrated methodology for analytical classification of daily travel-activity behavior. *Transportation Science*, 17:405–429, 1983.

E.I. Pas and F.S. Koppelman. An examination of the determinants of day-to-day variability in individuals' urban travel behavior. *Transportation*, 14:3–20, 1987.

E.I. Pas and S. Sundar. Intra-personal variability in daily urban travel behavior: Some additional evidence. *Transportation*, 22:135–150, 1995.

J.C. Platt. *Advances in Kernel Methods - Support Vector Learning*, chapter Fast training of support vector machines using sequential minimal optimization. MA. MIT Press, 1998.

C. Prato and S. Bekhor. Path enumeration by using branch and bound technique. *Transportation Research Record: Journal of the Transportation Research Board*, 1985:19–28, 2006.

PTV. *VISUM 12. 5 User Manual.*, 2012.

C. Ratti, S. Williams, D. Frenchman, and R. M. Pulselli. Mobile landscapes: Using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design*, 33:727–748, 2006.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

R. Schlich and K.W. Axhausen. Habitual travel behaviour: Evidence from a six-week travel diary. *Transportation*, 30:13–36, 2003.

S. Schonfelder and K.W. Axhausen. Activity spaces: measures of social exclusion? *Transport Policy*, 10:273–286, 2003.

N. Schüssler and K.W. Axhausen. Accounting for route overlap in urban and suburban route choice decisions derived from gps observations. In *12th International Conference on Travel Behaviour Research*, 2009.

K. Scott, G. Pabon Jimenez, and D. Bernstein. Finding alternatives to the best path. *Transportation Research Board*, 1997.

R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*, 16(1):30–34, 1973.

G.B. Singh. Comparison of hidden markov models and support vector machines for vehicle crash detection. In *Methods and Models in Computer Science (ICM2CS), 2010 International Conference on*, pages 1–6. IEEE, 2010.

P.H.A. Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17:201 –226, 1957.

D.J. Sun, C. Zhang, L. Zhang, F. Chen, and Z.R. Peng. Urban travel behavior analyses and route prediction based on floating car data. *Transportation Letters*, 6(3):118–125, 2014.

L. Tang and P. Thakuriah. Ridership effects of real-time bus information system: A case study in the city of chicago. *Transportation Research Part C: Emerging Technologies*, 22:146–161, 2012.

V. Trozzi, G. Gentile, M.G.H. Bell, and I. Kaparias. Effects of countdown displays in public transport route choice under severe overcrowding. *Networks and Spatial Economics*, 2013.

S.Y.A. Tsui and A.S. Shalaby. Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1972:38–45, 2006.

N.J. van der Zijpp and S. Fiorenzo Catalano. Path enumeration by finding the constrained k-shortest paths. *Transportation Research Part B*, 39:545–563, 2005.

V.N. Vapnik. *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

E.I. Vlahogianni, G. Yannis, J.C. Golias, N. Eliou, and P. Lemonakis. Identifying riding profiles parameters from high resolution naturalistic riding data. *Proceedings of the 3rd International Conference on Road Safety and Simulation (RSS2011)*, 2011.

E.L. Vlahogianni, G. Yannis, and J.C. Golias. Detecting powered-two-wheeler incidents from high resolution naturalistic data. *Transportation Research Part F: Traffic Psychology and Behaviour*, 22:86–95, 2014.

L. Volker. Route planning in road networks with turn costs. Technical report, Karlsruher Institut fur Technologie, 2008.

J. Vreeswijk, T. Thomas, E. van Berkum, and B. van Arem. Drivers' perception of route alternatives as indicator for the indifference band. *Journal of the Transportation Research Board*, page 10–17, 2013.

J.H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

J.G. Wardrop and J.I. Whitehead. Correspondence. some theoretical aspects of road traffic research. *ICE Proceedings: Engineering Divisions*, 1:767–768, 1952.

K.E. Watkins, B. Ferris, A. Borning, G.S. Rutherford, and D. Layton. Where is my bus? impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A: Policy and Practice*, (8): 839–848, 2011.

P. Winston. Artificial intelligence. Creative Commons BY-NC-SA, 2010. URL http://ocw.mit.edu. Accessed 7 Mar, 2015.

I. H. Witten, E. Frank, and M. A: Hall. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann Publishers Inc., 2000.

J. Wolf, R. Guensler, and W. Bachman. Elimination of the travel diary: experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board*, 36:125–134, 2001.

F.I. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(95):95–110, 1956.

R. Xu. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645 – 678, 2005.

S. Zhu and D. Levinson. Do people use the shortest path? an empirical test of wardrop's first principle. In *TRB 91st Annual Meeting Compendium of Papers DVD*, 2012.