# Modeling Traffic Information using Bayesian Networks

*Master's Thesis*

Paul van den Haak

# Modeling Traffic Information using Bayesian Networks

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Paul van den Haak
born in Tiel, the Netherlands

Computational Intelligence Research Group
Department of Man Machine Interaction
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Department Mobility and Logistics
Team Intelligent Transport Systems
Van Mourik Broekmanweg 6
2628 XE Delft, the Netherlands
www.tno.nl

Cover picture: Modified picture of an empty road in Australia, by Paul van den Haak.

# Modeling Traffic Information using Bayesian Networks

Author: Paul van den Haak
Student id: 1221760
Email: paulvandenhaak@gmail.com
Study: Media and Knowledge Engineering
University: Delft University of Technology
Date: August 2010

**Abstract**

Dutch freeways suffer from severe congestion during rush hours or incidents. Research shows that 64% of congested traffic during rush hour consists of commuter traffic [30]. A traffic congestion increases travel time, resulting in a delay for travelers. Reliable travel time predictions are essential for Dynamic Routing, in which travelers can be rerouted to avoid congestions. Travel times can be calculated from vehicle speed [41] in case of free flowing traffic. In case of congestion, we will make an estimation error regarding the travel time. Therefore, an accurate speed prediction model is necessary.

In this thesis, the predictability of the average vehicle speed by Bayesian Networks is investigated. A case study is conducted where several Bayesian Network models we propose are evaluated for a well known traffic bottleneck in the Netherlands. We show that Bayesian Networks are capable of predicting the start or end of a congestion at the bottleneck reasonably accurate for a prediction horizon until 30 minutes. Further, we propose a prediction model based on historical data, which is able to predict the average vehicle speed at the bottleneck location for longer prediction horizons. In the end, we propose a hybrid model which combines our Bayesian Network and our prediction model based on historical data. This hybrid model is able to predict a traffic congestion with an accuracy of 85% for a prediction horizon of 2.5 hours.

The results of our case study show that modeling traffic using Bayesian Networks is promising. Our models can form the input for a travel time prediction model for Dynamic Routing.

Thesis Committee:

| | |
|---|---|
| Chair: | Prof. Dr. Drs. L.J.M. Rothkrantz, Faculty EEMCS, TU Delft |
| Committee Member: | Dr. Ir. P. Wiggers, Faculty EEMCS, TU Delft |
| Committee Member: | Ir. H.J.A.M. Geers, Faculty EEMCS, TU Delft |
| Company Supervisor: | Drs. B.M.R. Heijligers, TNO |

# Acknowledgments

This Master's thesis is the result of 9 months of research during my graduation project. The work for this thesis was carried out at TNO Build Environment and Geo sciences in Delft and performed for Delft University of Technology. At this moment, I would like to thank some important people who made it possible for me to perform this research.

First, I wish to express my gratitude to my University supervisor, Prof.drs.dr. L.J.M Rothkrantz, for the continued support, devotion, invaluable insights, guidance and constructive criticism during my graduation project. I truly enjoyed our inspiring discussions.

I would also like to acknowledge TNO, and especially my supervisor Bjorn Heijligers. Although he is extremely busy, his guidance was invaluable during my thesis project. Furthermore, a special thank you goes out to Damir Vukovic for showing me the way in the world of Traffic Engineering. I would also like to thank Taoufik Bakri for sharing his Mathematical prowess with me. I truly enjoyed being a member of the Intelligent Transport Systems team at TNO and I would like to thank everybody on this team.

In addition, I would like to thank dr. P. Wiggers and ir. H.J.A.M. Geers for being part of my thesis committee. A special thank you goes out to Marek J. Druzdzel from the Decision Systems Laboratory of the University of Pittsburgh for letting me use the SMILE (Structural Modeling, Inference, and Learning Engine).

Last, but certainly not least, I would like to thank my parents and my girlfriend Kirsten, for supporting me during my graduation project. Especially in the final phase of my work, I had more attention and devotion to my thesis, than to them.

All of these contributions are gratefully acknowledged by me.

Paul van den Haak
Delft, the Netherlands
August 20, 2010

# Contents

# List of Figures

# Chapter 1

# Introduction

This Chapter gives an introduction to this thesis. The motivation of the subject for this thesis is given in section 1.1. Section 1.2 gives a detailed description of the problem formulation and the underlying assignments. This description is divided into 3 parts: a theoretical part, a data and software exploration part and a model and implementation part. Section 1.3 presents the scientific contributions of this thesis and section 1.4 describes the societal relevance of this thesis. An overview of the thesis is given in section 1.6.

## 1.1 Motivation

Many highways in the Netherlands suffer from severe congestion at some times of the day. Regular congestions can be anticipated, but an encountered incidental congestion will change travelers schedules and raise frustration. If there is a way to predict traffic congestions beforehand, people could anticipate by leaving on a different time or by choosing an alternative route. For instance a Personal Travel Assistant (PTA) could advice the traveler.

Congestions are directly related to the dynamic interplay between traffic demand and capacity. If the traffic demand is high and the road capacity is low, it is likely that there will be a congestion. Traffic situations can be measured with variables as speed, intensity or density. If these variables are predicted for freeway sections, traffic situations, for example congestions, can be predicted. When these predictions are done for a large time horizon, travelers can be notified for possible congestions at the earliest. Being notified for possible congestions, e.g. delays, gives travelers the possibility to anticipate. Correct traffic predictions are, of course, essential for the ability to anticipate on congestions. Therefore, there is a need for an algorithm to predict traffic situations accurately.

Travel times can be predicted from accurate predictions of average vehicle speed, if there are no congestions. For example, if it is known how fast a vehicle can travel on the A4, the highway between The Hague and Amsterdam in the Netherlands, over a prediction horizon of 60 minutes, its travel time for a certain distance can be calculated. These calculations form the basis for Dynamic Routing, in which travelers are routed for the shortest or fastest route based on Dijkstra's algorithm [6]. This thesis only deals with the prediction of average vehicle speed for Dutch freeways, as predicting speed is the basic step for Dynamic Routing.

In the research area of traffic prediction algorithms, there is a need for traffic data and characteristics. Traffic information management is usually organized by large organizations and governments. The Netherlands Organization for Applied Scientific Research (TNO) is an independent research organization whose expertise and research make an important contribution to the competitiveness of companies and organizations, to the economy and to the quality of society as a whole. New technologies are now enabling traffic information to dovetail with the requirements of the road user. This is a field in which TNO investigates both the user needs as well as new technologies such as traffic prediction models in the department of Intelligent Transport Systems (ITS). TNO is market leader in traffic information and prediction systems in the Netherlands and has extensive cooperation with governmental organizations such as the Dutch Ministry of Transport, Public Works and Water Management.

The Man-Machine-Interaction (MMI) group of Delft University of Technology is active in the Computational Intelligence field. The MMI section is enhancing user experiences, which requires intelligent, context-aware systems that learn automatically from the user and the environment. The research on computational intelligence within MMI focuses on the development of tools and methods that can deal with real life incomplete, uncertain and ambiguous data, obtained unobtrusively from the user. The group is internationally known for its success in applying these techniques to multi-modal sensing and multi modal fission of information. Research in Intelligent Transport Systems (ITS) with regards to traffic prediction models is therefore in line with the focus of this research group.

The combination of the theoretical knowledge in the computational intelligence field of the MMI group at Delft University of Technology and the knowledge, practical experience, data and connections of TNO is a strong basis for this thesis.

Traffic data is usually acquired from Inductive Loop Detectors (ILDs). Since there is an enormous amount of ILD data available at the Dutch Ministry of Transport, Public Works and Water Management, there is a need for an intelligent approach to cope with the abundance of data. Bayesian Networks give us the opportunity to represent and reason about an uncertain domain [17] like the field of traffic prediction. A Bayesian Network consisting of traffic variables, encodes domain variables and their qualitative and quantitative relationships [24] which is of course essential for prediction purposes.

## 1.2 Assignment

The thesis is part of the requirements for my Master of Science degree at the Man-Machine Interaction Group within the Faculty of EEMCS of Delft University of Technology. The thesis consists of three main parts:

1. A *theoretical part* which consists of an (1) investigation of the current literature on Traffic Modeling and (2) an investigation of the theoretical background of Bayesian Networks.

2. A *data exploration part* which consists of (3) data acquisition and preparation, (4) a data sensibility analysis and (5) a data streaming analysis.

3. A *model and implementation part* which consists of (6) modeling traffic by Bayesian Networks, (7) modeling traffic by historical based models, (8) a combination of historical modeling with Bayesian Networks and (9) a description of the developed software for this thesis.

A detailed description of the subparts of the assignment will be given in the remaining of this section.

### 1.2.1 Theoretical Part

**1. Investigation of the current literature on Traffic Modeling:** Before we can develop prediction models, it is essential to get a general understanding of how traffic can be modeled. Therefore, there is a need for a comprehensive overview of traffic theory and traffic variables such as speed, intensity and density. Traffic variables can be influenced by other factors which is important for this thesis. Further, it is of interest how these variables are related to each other. After having researched the general traffic theory, an overview of different approaches to model traffic would be essential. It is expected that traffic modeling can be approached from different disciplines.

There are numerous traffic prediction systems available in the world, but traffic characteristics differ for different countries because of regulations, environmental situations and social differences between people. Since this thesis focuses on traffic prediction for Dutch freeways, there is a need for a comprehensive overview of current traffic prediction systems for Dutch freeways.

**2. Investigation of the theoretical background of Bayesian Networks:** Since this thesis researches how traffic can be modeled with Bayesian Networks, it is essential that the theoretical background of Bayesian Networks is investigated. A good understanding of the basic Bayesian probability theory is essential for this modeling process. A notation for Bayesian Networks must be formulated which holds for this thesis. Further, the process of using data to model Bayesian Networks, which is called training or learning, should be investigated.

### 1.2.2 Data and Software Exploration Part

**3. Data acquisition and preparation:** Data acquisition is important for every data driven model. Therefore, an investigation in data acquisition methods is necessary.

Traffic data in its raw format is not directly usable for traffic predictions. Traffic measurements such as speed, intensity or density from different locations and for different time intervals are likely to contain missing values or outliers. Therefore, there is a need to prepare the data for further research and analysis.

**4. Data sensibility analysis:** Before we can develop prediction models, it is wise to get an idea of how large the traffic problem actually is. An impression of the most congested traffic area is essential to find possible causes for congestion. Further it is necessary to get an idea of the data sensibility. It should be investigated if there are correlations in the data. If a motorway location has certain characteristics, it is interesting to know whether these characteristics can also be found at other locations. Further, it should be investigated how sensible traffic data is for congestions. Are there typical congestion patterns for certain days? This question should be answered quantitatively by inspecting distributions of average speed for certain situations.

**5. Data streaming analysis:** We expect a huge variance in the total dataset of traffic measurements such as speed, intensity and density. It should be investigated if there are homogeneous subgroups to be found in this data. This can be done by conducting an analysis of the variances in the data to find subgroups of data which have similar patterns and a low variance. This experiment can be conducted for a certain test location, but methods for automatically detection of these homogeneous subgroups is desired. The data should be explored to find similarities or, more specific, traffic patterns.

### 1.2.3 Model and Implementation Part

**6. Modeling traffic by Bayesian Networks:** Gained knowledge about traffic until this point should be modeled in a Bayesian Network. Bayesian Models can be trained on historical data. Different Bayesian Models should be implemented, tested and evaluated in experiments. The most promising Bayesian Network should be evaluated on its robustness.

**7. Modeling traffic by historical based models:** Since a large amount of historical traffic data has already been collected, it should be investigated how this can be used for prediction purposes. A historical based prediction model, based on historical patterns, should be implemented, tested and evaluated in experiments.

**8. Combination of historical modeling with Bayesian Network modeling:** It is expected that Bayesian Networks are able to predict traffic for short horizons. Historical models could be able to predict traffic for longer horizons, but it is expected that a model might be too general and therefore not able to predict certain specific traffic situations accurately. The expert knowledge of traffic is expected to be encapsulated in the Bayesian Network, since this network can be trained on a large historical database which consists of a large amount of traffic examples. It should be investigated whether a hybrid approach between a historical based predictor and a Bayesian Network is beneficial.

**9. Description of the developed software for this thesis:** The software developed for this thesis should be described. It is important that the software is described by the Unified Modeling Language (UML), so that every computer scientist can easily understand how the system is designed. This is beneficial for researchers for future work as well.

## 1.3 Scientific relevance

An overview of the scientific relevance of this thesis is given in the next enumeration:

1. **An overview of traffic data acquisition and preparation.** This thesis presents different methods for traffic data acquisition and a detailed explanation for data preparation. Since ILD data usually contains 12% missing values due to maintenance, incidents, accidents, malfunction or power or communication failures [41], there is a need for filling in these missing values. This thesis applies a well known filter and shows that the spatio and temporal characteristics of the data are kept.

2. **An application of Bayesian Networks in the field of traffic prediction research.** Only a small amount of research has been conducted in the application field of Bayesian Networks in the traffic research area. This thesis presents different Bayesian Models and applies these to

real world traffic problems with real data. The application of Bayesian Networks on realistic, non-simulated data gives a realistic indication of its performance.

3. **A set of Bayesian Network models for traffic Prediction.** This thesis presents four manageable Bayesian Network models which can easily be implemented and tested in different settings. Research in this thesis shows that the model is robust and able to predict traffic, which is an invitation for other researches to continue this path.

4. **A historical based prediction model.** To strengthen the prediction horizon of Bayesian Networks for traffic prediction problems, a historical based traffic predictor has been developed, implemented and tested in this thesis. This prediction algorithm shows how traffic predictions can be done for longer time horizons.

5. **A hybrid model which combines Bayesian Networks with historical knowledge.** This thesis shows how a Bayesian Network model can be combined with a historical based traffic predictor. This gives the research field of Bayesian Networks a new twist from which the foundations have been laid in this thesis.

6. **Research workbench for traffic prediction.** A workbench has been implemented for the work on this thesis to acquire traffic data, to prepare traffic data and to implement and test models. This workbench is developed as general as possible, so that modifications or additions can easily be made by other researchers.

## 1.4 Societal relevance

A reliable and robust traffic network is highly relevant to meet the infra-structural needs of travelers. There are three groups of people who benefit from a reliable travel prediction system: the *travelers*, the *transportation companies* and the *road managers*:

- **travelers** clearly benefit from accurate travel predictions. This holds for leisure traffic, commuter traffic or emergency traffic and especially during rush hours. Traffic during rush hours consists for 64% of commuter traffic [30], which can be seen as the most important group. For commuter traffic, delays are extremely inconventient and could cause financial implications. An accurate traffic prediction system could give commuter traffic the opportunity to anticipate on possible road congestions. Further, if traffic situations can be monitored and predicted accurately, emergency services could use this information to plan their route wisely.

- **transportation companies** need to plan their deliveries and pickups wisely to satisfy their customers. If their truck drivers face congestions on their route, this will have a huge impact on costs such as fuel, vehicle maintenance, time and CO2 impact.

- **road managers** need to have a continuously updated overview of their road network. They must be able to intervene in case of accidents, extreme weather, overcrowded roads or other extreme circumstances. If these extreme circumstances could be predicted, the needed reaction time for road managers increases and better solutions or workarounds can be devised. Further, a well developed research workbench in which traffic data can be analyzed and prediction performance can be evaluated is beneficial for performance evaluations or maintenance of the road network for road managers.

If an accurate travel prediction systems ables travelers to plan their trip more wisely, ables transportation companies to plan their deliveries more wisely and ables road managers to optimize the road network, there will be less traffic congestions. Less traffic congestions will of course has its effect on the environment (CO2 reduction) and road safety.

## 1.5 Traffic information research

This section gives an overview of traffic prediction research and our contribution to the field. Figure 1.1 shows an example of a road network. Lets say that we want to travel from Delft to Amsterdam in the morning rush hour. This road network shows that we can travel over Leiden or over Utrecht. Traveling over Leiden seems to be the most shortest route in this graph.

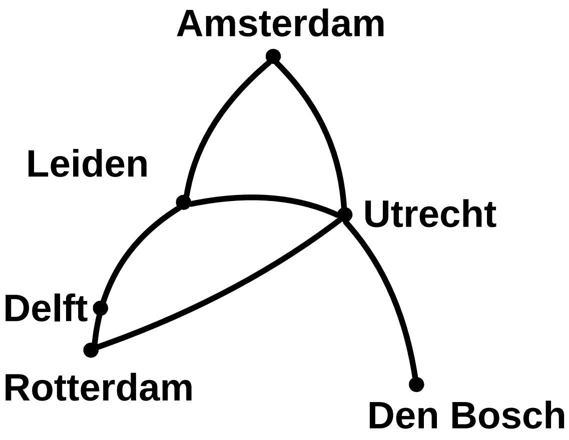


Figure 1.1: An example of a road network

Before we leave Delft, we can calculate the travel time to Amsterdam based on Dijkstra's algorithm [6]. This travel time can be updated when there is new information. If there is new information, we can calculate the travel time again and have up-to-date information about the travel time. In this way, we will always be too late for detecting a congestion.

Another approach is to predict the travel time based on historical data. In this approach, we find the best historical models for this morning rush hour between Delft and Amsterdam. Based on the historical speed measurements in these model, we can calculate the travel time. In our prediction model, which is based on historical data, we continuously find the best fitting historical models. If the current traffic situation changes, we are able to find other historical models and update the predicted speed from which the travel time can be calculated later on. Traffic congestions can be predicted beforehand based on historical data.

Another approach to predict traffic information, is by developing a prediction model. This can be done by, for example, statistical models. In this thesis, we predict the vehicle speed by using Bayesian Networks. As an addition, our Bayesian Network is combined with our prediction model based on historical data.

The different approach for traffic prediction is to track individual vehicles real time from their navigation system. In this way, we have knowledge about the routes of individual vehicles which we can use to predict traffic flow.

When accurate prediction models are able to predict a congestion, travelers can be rerouted. But if we reroute travelers, the historical models might not be representative anymore. And what should we do if travelers do not follow the advice of being rerouted?

The field of traffic research is challenging and will not be solved completely in the next few years. Nevertheless, our research in this thesis investigates whether Bayesian Networks can be used to predict traffic speed. These predictions can later on be used for travel time predictions and to reroute travelers.

## 1.6 Thesis overview

This thesis is divided into four main parts which are set up in the following way:

- **Part I:** contains the theoretical part of this thesis. Here, an overview of traffic modeling theory and the current state of the art is given in Chapter 2. The theory about Bayesian Networks is described in Chapter 3.

- **Part II:** contains the data exploration part. For the experiments which are conducted in this part, an explanation of the data preparation is given first in Chapter 4. Then, a sensibility analysis of the data is conducted in Chapter 5. After this analysis, an experiment is conducted to find traffic patterns in a traffic measurements database. This experiment and its results are described in Chapter 6.

- **Part III:** contains the model and implementation part. First, the Bayesian Models are proposed, implemented and evaluated in Chapter 7. The historical based predictor is proposed, implemented and evaluated in Chapter 8. Then, the most promising Bayesian Model is combined with the historical based predictor to form a Hybrid model, which is described in Chapter 9. At the end of this part, the developed software for this thesis is described in Chapter 10.

- **Part IV** contains the results and final remarks of this thesis. The conclusions and discussion are given in Chapter 11. In the end, some ideas are given for future work in Chapter 12.

# Part I

# Theoretical Part

# Chapter 2

# Traffic Modeling

Before we are able to model traffic, it is necessary to understand the fundamentals of traffic theory first. Section 2.1 describes traffic measurements (data) such as speed, intensity or density and its underlying relations. These underlying relations are important for modeling traffic. There are different modeling approaches. The most simple approach for traffic modeling is the *instantaneous* approach which is described in section 2.2. Civil Engineers often take the model approach, which is described in section 2.3. Another approach is the *data driven* approach, which could be seen as a top-down modeling approach. The data driven approach is described in section 2.4. In the end, related work is described in section 2.5.

## 2.1 Traffic theory

Traffic can be seen as a dynamic interplay between traffic demand and traffic supply. Traffic demand can be seen as the number of vehicles using the road network and the supply can be seen as its capacity [41]. In fact, all analytical techniques of traffic systems are structured in a demand-supply framework [11]. This framework can focus on microscopic interactions, in which individual traffic units are studied, or macroscopic interactions, where attention is given to groups of traffic units in aggregated form. These traffic units can have certain movements in a road network which can be called *traffic flow*. It this stage, it is essential to provide the readers some theoretical basic principles of traffic flow and to present some formal descriptions which are used for the rest of this thesis.

Traffic typically flows through a network with roads (*links*) and intersections (*nodes*). Traffic on freeways has different characteristics opposed to traffic in urban areas, since traffic on freeways usually has less intersections and different traffic regulations. This thesis only discusses traffic flow on freeways.

The most important variables in traffic analysis are: *trajectories*, *intensity*, *density* and *speed*, which will all be explained in this section respectively.

Trajectories are paths which a traffic unit or a group of traffic units follow. Traffic flow can be seen as a movement in two dimensions: *space* and *time*. If a certain traffic unit, lets say a car, travels on a road it will travel a certain distance in a certain amount of time. An example trajectory is illustrated in Figure 2.1. In this figure, the variable $t$ represents time, and $x$ represents space (a location on the road). Lets say that the distance $x$ is measured in kilometer, and time $t$ is measured in hours, than the

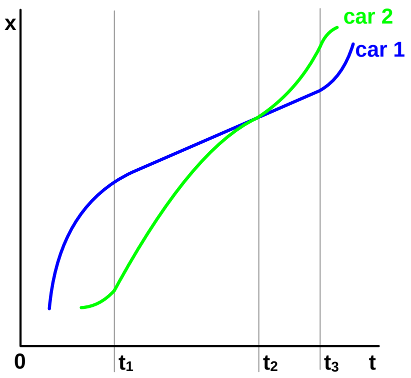


Figure 2.1: Vehicle trajectory example

angle of a trajectory actually represents the speed of the car in km/hour. Speed is actually a continuous variable during a trajectory which can change in value over time. Speed $V_i$ at time $t_i$ can be expressed in formula form as follows:

$$V_i(t) = \frac{d}{dt}x_i(t). \tag{2.1}$$

If an angle in a trajectory changes, i.e. the travel speed of a car changes, then the car could be accelerating or braking depending whether the angle increases or decreases respectively. The acceleration can be expressed as follows:

$$a_i(t) = \frac{d}{dt}V_i(t) = \frac{d^2}{dt^2}x_i(t). \tag{2.2}$$

Keeping this in mind, we can see in Figure 2.1 that car 2 travels from the same starting location as car 1, but car 1 starts earlier. Since the average angle in the graph for car 2 is greater then the angle of car 1, i.e. car 2 travels with a higher speed compared to car 1, car 2 is able to overtake car 1 at $t_2$. Both lines in this figure are called a *trajectory*. In this example, the trajectories represent single cars, but for macroscopic models these trajectories could of course also represent groups of cars or other traffic units.

The variable *intensity* expresses the number of vehicles passing a certain location of a road in a unit of time [11]. An intensity can be measured for a section of a road, or more specific: for a separate lane. If only measurements for separate lanes are available, these values could be aggregated to indicate section intensities. In general, the intensity is expressed as follows:

$$q = \frac{n}{T}, \tag{2.3}$$

where $n$ is the number of vehicles which pass the cross-section of a road and $T$ is the unit of time. Normally, intensity is expressed in vehicles/hour, but for some situations it might be useful to have 1, 5 or 15 minute aggregates for intensity.

The variable *density* of a traffic flow is the number of vehicles present on a unit of road length at a certain time. The formal definition for density is:

$$k = \frac{m}{X}, \tag{2.4}$$

where $m$ denotes the number of vehicles which are present on the road section, and $X$ denotes the unit length of the road section [11]. Usually, the density is given in vehicles/km and it can refer to a road or a specific lane.

Traffic researchers are often interested in *mean speeds* of vehicles on roads. The mean speed can be expressed in two ways [11]:

1. For cars *which are passing a cross-section of a road during a certain period*. This can be expressed as a formula as follows:
$$u_L = \frac{1}{n}\sum_{i=1}^{n} V_i, \tag{2.5}$$
where $L$ refers to a certain location.

2. For cars *which are present on a road section at a certain time* . This can be expressed in formula form as follows:
$$u_M = \frac{1}{m}\sum_{i=1}^{m} V_i, \tag{2.6}$$
where $M$ refers to a certain moment.

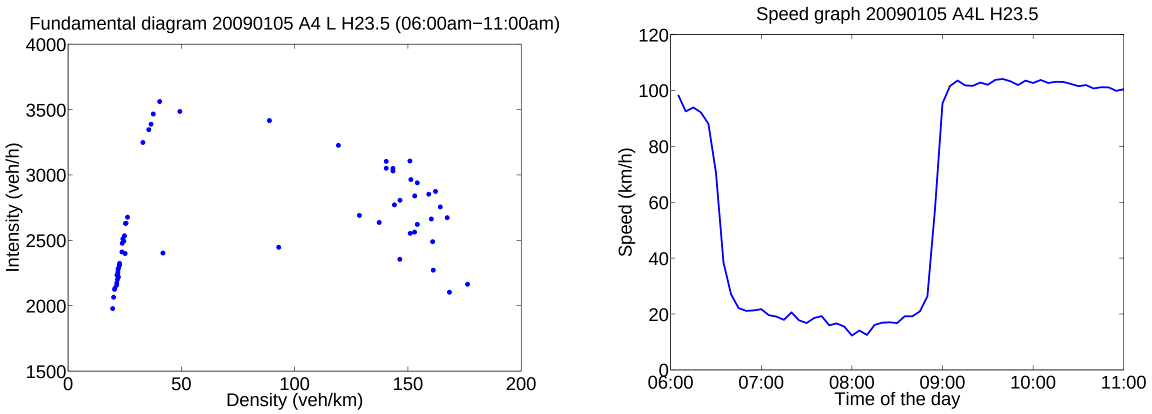
A detailed explanation on dealing with mean speed values can be found in section 4.

If a stationary traffic situation is considered, it is reasonable to assume that there exists a relation between mean speed, intensity and density. Speed is measured in kilometer/hour, intensity in vehicles/hour and density in vehicles/kilometer. If, from the set of speed, intensity and density, two are known the other can be calculated by doing some simple mathematics. To be more specific: density = intensity/speed, speed = intensity/density and intensity = density · speed.

After having explained the definition, mathematical expression and relations of speed, intensity and density, it should be noted that there is something special about their macroscopic relation which is called: a *fundamental diagram*. More details of this diagram can be found in [11], but for now it is important to inform that this fundamental diagram is not a physical law, but depends on the characteristics of:

- road type
- traffic composition (trucks, cars, etc)
- traffic type (commuter traffic, holiday traffic, etc)
- measures such as speed limits
- lighting conditions (daylight, dark, road lighting)
- weather conditions (dry, rain, fog).

The fundamental diagram consist of the relation between intensity-density, speed-density an speed-intensity. For illustration purposes, the relation between intensity and density is illustrated in Figure 2.2(a). The figure consists of all data points for a certain location at the A4 in the Netherlands measured between 06:00 am and 11:00 am on Monday the 5th of January 2009. By looking close to this figure, an important aspect becomes visible. If the flow (intensity) and density are increasing seen from the origin, there is a certain flipping point where after the flow immediately drops down and the density increases. Generally, a low intensity and a high density indicate a traffic jam. Before this flipping point there exists a 'instable' traffic situation where the flow is likely to drop suddenly. By looking at the corresponding speed graph in Figure 2.2(b) it becomes clear that there indeed was a traffic jam between 06:00 am and 09:00 am, which is typical for a morning rush hour. This knowledge should be kept in mind for the rest of this thesis.



(a) Fundamental diagram between intensity and density

(b) Typical speed graph of a congested morning rush hour

Figure 2.2: Fundamental traffic relation between intensity and density with a corresponding speed graph

After having explained these important traffic relations and fundamentals of traffic characteristics, it is important to explain how this information is gained and used in general. As explained before, traffic can be seen as a dynamic interplay between traffic demand and traffic supply. If this dynamic interplay is played well, travelers can travel from origin to destination in time. Travel time is important information for travelers as well as for road management, since it could be seen as the performance of a road network. Travel times are bound to change if the traffic conditions change. Accurate travel times are most relevant before a traffic jam, because this is the moment drivers can still change their plans. There are different approaches for traffic information prediction:

- instantaneous approach
- model approach
- data driven approach.

These approaches are explained in more detail in the remaining of this Chapter.

## 2.2 Instantaneous approach

The instantaneous approach only takes historical or current data into account and travel times are often predicted with an instantaneous approach, or the *online* approach as it is sometimes called [41].

The instantaneous approach searches historical or current data and should be able to yield immediate results if the number of necessary computations are optimized. Van Lint [41] states that travel time estimators based on historical data can only be used as predictors if and only if the current traffic conditions remain stationary for the time the prediction holds. Of course, the fact that traffic conditions remain stationary is debatable.

There are many different travel time estimators, but the most common approach are speed-based travel time estimators based on inductive loop detector data (ILD) because of their simplicity [39]. Details on these inductive loop detectors are given in Chapter 4. Speed data is available at locations where ILDs are situated, but the speed between the measuring locations is unknown. Different methods exists in literature for estimating travel times from speed values of ILDs. A convenient overview is given by Li et al. in 2006 in [19].

**Instantaneous model**

The most simple model is the instantaneous model, which is also the basic of the well known VicRoads Drive Time system in Australia [16]. In this method, the average speed on a link is calculated by averaging the speed of the ILD at the start end at the end of the link according to the following formula:

$$\mu(i,t) = \frac{2l_i}{v(i_a,t)+v(i_b,t)}, \tag{2.7}$$

where $\mu(i,k)$ denotes the mean speed for link $i$ at time $t$, $v(i_a,t)$, $v(i_b,t)$ are the speeds on both ILD's at time $t$ and $l_i$ denotes the distance of the link. As becomes clear from this formula, the variable $t$ does not change in this formula. This indicates that the speed information from the ILD is taken at exactly the same moment on both ILD's, but the law of physics imply that a vehicle cannot be at both ILD's at the same time. Therefore, this formula can be seen as an approximation for the mean speed. The advantage of this method is that it can be applied online since it only relies on current traffic measurements and this makes it computationally inexpensive.

**Time Slice model**

Another model found in literature and reviewed by Li et al. in [19] is the *Time Slice model*. This model incorporates the fact that speed is a function of the time. Therefore the speed value from the ILD is taken on the time that a vehicle passes that particular ILD. The travel time of the first link must be calculated in the same way as in the instantaneous model, because here the time is $t_0$. To calculate the travel time of the consecutive links, we need to incorporate the fact that time has already passed for crossing the previous link(s). Lets denote the time to travel a certain link as $t(i,t_i)$, where $i$ is the link number and $t_i$ is the the time when the link is traveled. This is not correct for the first step, as this should be the result of the instantaneous model: $t(1,k)$ as explained previously. The complete model is defined by Li et al. [19] as follows:

$$t(n,t_n) = \frac{2l_n}{v(n_a,t_n)+v(n_b,t_n)}, \tag{2.8}$$

where $t_n = k + t(1,k) + \sum_{i=2}^{n-1} t(i,t_i)$, $l_n$ denotes the distance between link $a$ and link $b$. The total travel time can be calculated by adding all these individual link travel times. For each time slice, a new travel time is calculated and this makes this method more computationally demanding than the instantaneous model. Therefore, it is more suited for off-line computations [19].

**Dynamic Time Slice Model**

Another model presented in the overview by Li et al. [19] is the *Dynamic Time Slice* model, developed by Cortes et al. [4]. This model is not very different from the time slice model, but it calculates the travel time at link $i$ given time $t_i$ instantaneous by a recursive formula. Since it is computationally heavy as well, this model is used for off-line computations.

**Piece-wise Constant Speed Based model (PCSB)**

Another model, which is mostly used nowadays for travel time estimations in the Netherlands, is the piece-wise constant speed based (PCSB) trajectory method developed by van Lint [41]. Although this method is not presented in the overview by Li et al. [19], there is a extensive description in the dissertation written by van Lint in 2004 [41]. The most common approach taken nowadays is this PCSB algorithm and here speed is assumed to be constant between two ILD's [41]. To illustrate how this works, Figure 2.3 gives a graphical representation of a vehicle trajectory [41]. Details of vehicle trajectories are described in section 2.1. This representation is called a space-time graph, in which the horizontal axis represents time and the vertical axis represents space (distance). With simple calculations one could interpret the angle of the trajectory as the vehicle speed.

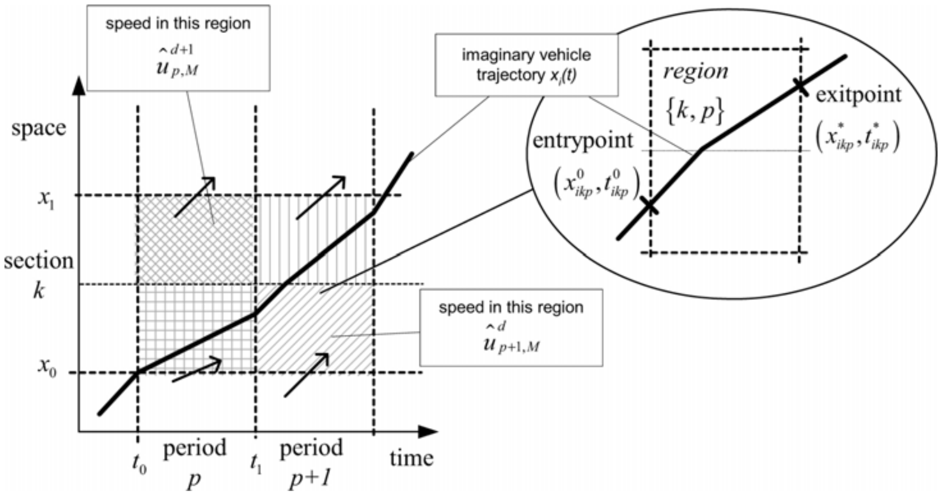


Figure 2.3: Example of a vehicle trajectory in a space time graph [41]

In this figure, $x_0$ and $x_1$ denote ILD's along the freeway. The period is the window between $t_0$ and $t_1$ in which the speed values from the ILD's are sampled. A specific algorithm is necessary to construct travel times based on this ILD data. To construct a travel time, based on the ILD data, we can construct a trajectory by starting at $t_0$ at $x_0$ and draw a trajectory with the given speed (angle) until we cross the new time point $t_1$ or the new ILD $x_1$. If the trajectory is about to cross a new time point, the next part of the trajectory will be drawn with a angle based on the speed value of the current ILD ($x_0$) at the new time $t_1$. Otherwise if the trajectory is about to cross the next ILD, the line will be drawn with an angle based on the speed value of the next ILD $x_1$ at the current time $t_0$. In this way, the algorithm makes use of information which is most recent based on the location (space) of the vehicle and the current time.

**Piece-wise Linear Speed Based model (PLSB)**
An extension of the PCSCB algorithm, the *Linear* model, is presented as the fourth model by Li et al. [19]. This model was earlier presented by van Lint in 2004 where it is called the *piece-wise linear speed based model* (PLSB). This model incorporates the fact that vehicle speeds cannot be constant, but speed is a constant changing value and it changes smoothly. Therefore, Van Lint proposed a method which extrapolates the speed values linearly so that a smooth transition of speed values is secured [41]. The formulas in this method tend to become more difficult but can be solved analytically. The question is, will this work better? Van Lint explains that it depends on the actual traffic conditions in between ILD's wether the PCSB or PLSB method yields better results.

As explained previously, to get accurate predictions from instantaneous travel time estimators, the traffic conditions need to remain stationary in the time domain in which the prediction holds. The assumption of stationarity of traffic conditions is of course doubtful. Van Lint [41] claims that stationarity of traffic conditions cannot be assumed in congested traffic conditions, based on research conducted by Lindveld and Thijs in 1999 [20]. Van Lint [41] further claims that the performance of instantaneous predictors generally decreases when traffic is in a congested state. This is where accurate travel time predictions are most valuable. Van Lint [41] does not deliver actual proof for this statement, but Li et al. [19] propose an evaluation method for speed-based travel time estimation models.

Since traffic conditions in congested traffic cannot be assumed to be stationary and since travel time estimation tends to yield poor results in situations of increasing flow, the instantaneous approach might not me the best approach for predicting travel time. Although these algorithms could be used as an estimation for the initial values for calibration of prediction algorithms, the low performance in congested traffic conditions must not be underestimated. Errors made in these estimation could easily propagate in the prediction to yield even greater errors when used as calibration values.

## 2.3 Model approach

The model approach is often taken by Civil Engineers as it implies that the developer has extensive knowledge about the traffic domain. Van Lint [41] introduces *microscopic* and *macroscopic* traffic flow models. Hoogendoorn [12] adds the notions of *submicrosopic* and *mesoscopic* models in his State-of-the-art review in 2001 about traffic flow models, and clearly defines the models from submicroscopic to macroscopic ordered to their level of modeled detail. Submicroscopic traffic flow models give a high detailed description of the functioning of vehicle subunits and the interaction with their surroundings. Submicroscopic traffic flow models are defined by Hoogendoorn [12] as follows:

**Definition 1.** *A submicroscopic model describes the characteristics of individual vehicles in the traffic stream. However, apart from a detailed description of driving behavior, also vehicle control behavior (e.g. changing gears, AICC operation, etc.) in correspondence to prevailing surrounding conditions is modeled in detail. Moreover, the functioning of specific parts (sub-units) of the vehicle is described.*

Microscopic models are aimed to model individual vehicle trajectories and are defined by Hoogendoorn [12] as:

**Definition 2.** *A microscopic simulation model describes both the space-time behaviour of the systems entities (i.e. vehicles and drivers) as well as their interactions at a high level of detail (individually). For instance, for each vehicle in the stream a lane-change is described as a detailed chain of drivers decisions [12].*

Mesoscopic models can be put between the microscopic and macroscopic models with respect to their level of detail in model formulation en are defined by Hoogendoorn [12] as:

**Definition 3.** *A mesoscopic model does not distinguish nor trace individual vehicles, but specifies the behaviour of individuals, for instance in probabilistic terms. To this end, traffic is represented by (small) groups of traffic entities, the activities and interactions of which are described at a low detail level. For in- stance, a lane-change manoeuvre might be represented for an individual vehicle as an instantaneous event, where the decision to perform a lane-change is based on e.g. relative lane densities, and speed differentials. Some mesoscopic models are derived in analogy to gas-kinetic theory. These so-called gas-kinetic models describe the dynamics of velocity distributions [12].*

Macroscopic models tend to model complete streams of traffic [41] and therefore it decreases the level of detail in which the model is formulated. Macroscopic models are defined by Hoogendoorn [12] as:

**Definition 4.** *Macroscopic flow models describe traffic at a high level of aggregation as a flow without distinguishing its constituent parts. For instance, the traffic stream is represented in an aggregate manner using characteristics as flow-rate, density, and velocity. Individual vehicle manoeuvres, such as a lane- change, are usually not explicitly represented. A macroscopic model may assume that the traffic stream is properly allocated to the roadway lanes, and employ an approximation to this end. Macroscopic flow models can be classified according the number of partial differential equations that frequently underlie the model on the one hand, and their order on the other hand [12].*

Research in Traffic Flow modeling has its roots in the 1950's and since then a wide range of traffic flow theories and models have been developed. To explain them all here would be out of scope for this thesis. These models and theories can be categorized to their level of detail from submicroscopic to macroscopic as explained previously. Hoogendoorn [12] presents a comprehensive overview of the most important steps in traffic flow modeling since the 1950's. This overview is presented in appendix B for convenience.

The main question at this moment is, which model is the best? This question is impossible to answer since each model has its own advantages and disadvantages. But in the higher level of abstraction, the general applicability and performance of these models can be discussed and categorized to their level of detail in variables.

Traffic flow modeling is based on the assumption that the behavior of each variable of the model is a function of traffic conditions and its environment [12]. However, characteristics like human behavior are hard to observe and to measure. The level of detail in which the variables can be measured, is largely important for the choice of detail in the model.

Submicroscopic and microscopic models describe each vehicle by its own equations, which yields a burden in computational power and time. Therefore, these models are most suitable for off line computations for detailed characteristics as car design, drive support systems, etc. [12].

Hoogendoorn claims that Macroscopic models are suitable for analyzing and reproducing macroscopic characteristics such as queue lengths and shock waves. Shock waves are defined as:

**Definition 5.** *Shock waves in traffic flow propagates from one vehicle to the next, while being amplified until at some point a vehicle comes to a complete stop. [12].*

Nevertheless, Traffic Flow modeling has the disadvantage that the computational complexity grows proportionally with the number of vehicles, and might therefore not be the ideal approach for predicting traffic. Engineers need extensive knowledge about traffic flow when taking the model approach which could also be a burden for choosing this approach. In the end, the model needs traffic demand and supply as input which are hard to predict [41].

## 2.4 Data driven approach

The data driven approach differs itself from the model approach in the way it considers traffic processes. Data driven models predict travel times by inductive techniques, while the model approach throws in extensive knowledge of the traffic flow. In this interpretation, data driven methods can be seen as a *top-down* approach while the model approach is a *bottom up* approach.

Data driven methods try to correlate observed traffic measurements to current and past traffic data to predict traffic [41]. When correlations are found between historic traffic data and the corresponding travel times, pattern matching is done with the current traffic data on historic traffic data to find known travel times in history. The best suitable travel time given the best match of traffic data is usually given as an answer.

There is an enormous amount of data driven approaches for prediction of travel times to be found in literature and to list them all here would be too much. An extensive overview is presented in van Lint [41] and can be categorized as follows:

- ARIMA models
- linear time series models
- non-linear time series models
- support vector regression models
- feed-forward neural networks
- recurrent neural networks.

A rather new and promising method is predicting traffic by using Bayesian Networks. The focus for this thesis is on modeling traffic with Bayesian Networks. Different approaches have already been investigated in this field in [33], [34], [35], [46] and [25].

Sun et al. [34] proposed a Bayesian Network (BN) model in 2004 to forecast traffic flow with incomplete data. These method is concerned with the prediction of traffic intensity. First, the structure of the Bayesian Network proposed by Sun et al. [34] is explained with the help of Figure 2.4.

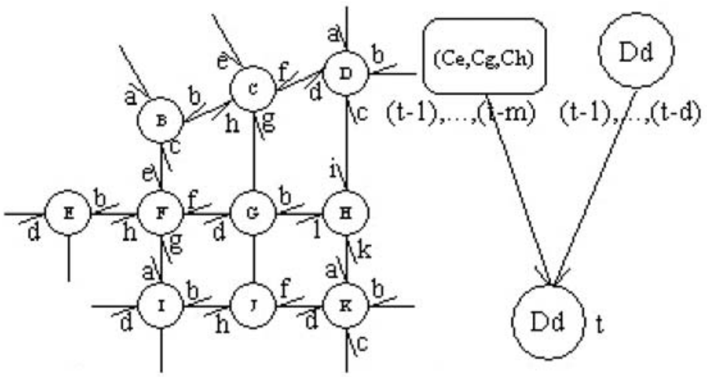


Figure 2.4: The Bayesian Network as proposed by Sun et al. [34]

This figure represents a subsection of a road network where travel directions are denoted as arrows on the links. Each circle is a node where travelers can change directions. Sun et al. [34] do not explain the length of a road link. Suppose that we would like to predict the intensity at node $D_d$ at time $t$. Sun et al. [34] claim that the vehicle flows of $C_e$, $C_g$ and $C_h$ should have causal relations with the vehicle flow of $D_d$. Of course, we could question ourselves whether this relation is truly causal. Sun et al. [34] also take intensity values from road links at different time instants in the past. The Bayesian network grows in number of nodes by taking the history into account. This method is proposed for short-term traffic flow forecasting, which is in the range of five minutes to half an hour according to Sun et al. [34].

After creating this Bayesian network, Sun et al. [34] performed a Principal Component Analysis (PCA) to reduce the number of dimensions in which the data is present. PCA reduces the dimension of the data by a mathematical procedure which transforms variables into principal components. A principal component is a linear combination of variables chosen such that data originating from different classes is separated as far as possible. With this procedure, the data becomes more discriminable so that classes can be defined. However, it becomes complicated to understand these principal components. Therefore, we could question ourselves whether PCA is useful in this situation. To understand traffic flows and to predict travel speeds or intensities, it is certainly for traffic managers interesting to know the bottlenecks of a traffic network and to understand the variables.

After reducing the dimension of the data, a Gaussian Mixture Model (GMM) is used to approximate the joint probability distribution between input and output of the Bayesian Network. The parameter estimation for the GMM is done by the well known maximum likelihood estimation (MLE) method which is carried out iteratively by the well known expectation maximization (EM) algorithm. Both the MLE and the EM algorithm are described in detail in [10].

The results of the work of Sun et al. [34] show that certain peaks in the traffic flow intensity in the real data, are hard to predict. The prediction method is not able to recognize all the peaks. The peaks are of course most valuable for travelers and road managers, since there can be acted upon if they are known beforehand.

In 2005, Sun et al. [35] proposed a spatio temporal Bayesian network which is actually an extension of the previously proposed approach. The term *spatio temporal* here means that the Bayesian network incorporates, besides traffic flow intensities, also common-sense background information. Sun et al. [35] introduce the fact that people's activities around shopping centers, car parks, home communities, etc. usually obey some consistent laws on the long term. Shops have opening hours, people have working hours, etc. Sun et al. [35] propose a method on how to use this spatio temporal information of a complete transportation network to construct feasible Bayesian networks to forecast traffic flows.

The method proposed by Sun et al. in 2005 [35] is a procedure to find road sections (ILD's) which have a causal relationship with other road sections. The strength of this correlation is calculated by the Pearson correlation coefficient. A complete transportation network contains a lot of nodes which endangers the computational feasibility and incorporation of irrelevant data of Bayesian Networks. Sun et al. [35] rank the variables based on their Pearson correlation and form an order of variables by using the best-first search strategy. After this ordering, the best $n$ variables are taken into account for further computation.

An advantage of this approach compared to the PCA approach is that now it is clear which characteristic is represented by each variable. This is important for road managers and analysts [41]. Sun et al. [35] do not explain exactly which variables are used in their Bayesian Network, but eventually the forecast (output) is a traffic flow intensity value. The input data for their Bayesian Network originates from UTC/SCOOT, which is a system of the Traffic Management Bureau of Beijing. Traffic flow intensities are measured in vehicles/hour. No details are described about a train or test set separation, and about how the performance is measured.

As explained previously, the first $n$ variables (road links) are taken as prerequisites for the causal relations. It is not straightforward to chose a value for $n$. In 2006, Sun et al. [33] extended their Bayesian Method with a detailed investigation on dimensionality of their data. Instead of ranking the variables, Sun et al. [33] returned to the well known PCA technique to reduce dimensionality, but now they took into account the amount of variance which could be explained by the principal components as well as the number of training objects. Using a higher dimension implies that more training data is needed for accurate training according to the curse of dimensionality [21].

After ranking the variables the joint probability distribution is predicted in the same way as in the basic approach as is explained earlier. Also the used inference technique here is again the EM algorithm. There is no detailed information on the results of the predictions done with this method in Sun et al. [35].

As a reaction on Sun's work [34], [35], [33], Yu et al. [46] proposed another method which is slightly more sophisticated. The aim of the method proposed by Yu et al. [46] is to predict the short term traffic speed. The average vehicle speed is taken from the ILDs as input data and vehicle speeds at a certain link in the future can be predicted. Yu's method differs in two ways:

Firstly, instead of just taking downstream links, Yu et al. [46] also include upstream links in the Bayesian Network. Downstream links are links along the driving direction and upstream links are links against the driving direction. Basically, the model proposed by Yu et al. [46] 'looks' back and ahead. Figure 2.5 represents a part of a road network. $BC$ represents the road link from $B$ to $C$. The upstream links in this example are $AB$,$EB$ and $FB$ and the downstream links are $CD$, $CG$ and $CH$.

Secondly, after computing the prediction of traffic speed with the Bayesian Network, the current flow data is used to correct the prediction when necessary.

To predict traffic information, Yu et al. [46] take around 4 or 5 downstream links, 4 or 5 upstream

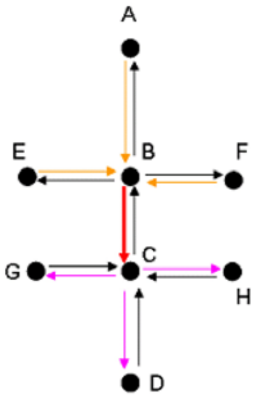


Figure 2.5: An example of a road network [46]

links and the current link as causal nodes in their Bayesian Network. The information from each link is taken at different time instants with steps of 5 minutes: $t_{-5}$, $t_{-10}$ and $t_{-15}$.

A Gaussian Mixture Model (GMM) with 3 components is used to approximate the joint probability distribution of the Bayesian Network. The parameter estimation for the GMM is done by the well known maximum likelihood estimation (MLE) method which is carried out iteratively by the well known expectation maximization (EM) algorithm. Both the MLE and the EM algorithm are described in detail in [10].

Yu et al. [46] do not incorporate real time information in the Bayesian Network. The complete PDF is computed beforehand and therefore cannot embed real time data. Since the traffic prediction accuracy of Bayesian Networks become unreliable in case accidents or incidents happen, the real time information is used in the last step to correct for unusual situations. The traffic flow at a certain moment is matched with a database for a similar day and a standard error (difference) can be calculated. Yu et al. [46] connect weights to these errors to compensate for incidents and accidents.

## 2.5 Related work

Currently, a state of the art system in the data driven traffic prediction field is developed by Inrix. Inrix is a traffic service company in the USA with partners as Microsoft, TNO, Tele Atlas and the Texas Transportation Institute. Inrix developed a Smart Driver Network which aggregates traffic-related information from GPS-enabled vehicles, mobile devices, traditional road sensors and other sources [1]. The service provided by Inrix is a real-time, historical and predictive traffic advice for highways and secondary roadways. Unfortunately, there are no technical details given as this is a commercial product.

The current state of the art system for travel time prediction on Dutch freeways is ASTRIVAL. This prediction algorithm only incorporates historical traffic measurements into account and can be seen as an historical model. Different algorithms are incorporated in ASTRIVAL to calibrate the

[1] More details can be found at www.inrix.com

prediction model off line [5]. The ASTRIVAL prediction system is compared to another state of the art prediction system developed by van Lint [41]. This system is based on a state space neural network (SSNN) and it performs better then ASTRIVAL [42]. Details of the SSNN prediction model can be found in [41].

# Chapter 3

# Bayesian Network Theory

This Chapter describes the theory about Bayesian Networks. A good understanding of the basic principles and available techniques will help us to develop an accurate traffic prediction model. Section 3.1 introduces the concept of Bayesian Probability, which is important to understand if dealing with Bayesian Networks and expert knowledge. In section 3.2, the definition of Bayesian Networks is given, based on literature. Then, section 3.3 gives the notation for Bayesian Networks which is used for the rest of this thesis. This notation is based on the standards found in literature, and illustrated with a traffic congestion based example. In the end, we give a brief description about how the conditional probability tables (CPTs) of Bayesian Networks can be learned from data in section 3.4.

## 3.1 Bayesian probability and statistics

Before we introduce the term Bayesian Networks and its learning methods, it is essential to first introduce the concept of Bayesian Probability since it differs from the classical probability. Bayesian Probability of an event $x$ is described by Heckerman [10] as a person's *degree of belief* in that event. An important difference between the classical probability and the Bayesian probability is that for Bayesian probabilities there is no need for repeated trails to measure the probability. As an example, consider the question: Will the Netherlands win the next World Championship in soccer? The classical statistician will remain silent but the Bayesian statistician can express his/her degree of belief in the Netherlands winning the Championship.

There is some criticism on the Bayesian approach to probability. Finkelstein and Fairley [7] published a paper in 1970 in which Bayesian probability was introduced in the court room for the first time. The most common critic which was given was that Bayesian probabilities seem arbitrary. It is not clear on which scale Bayesian probabilities are measured. The mean question is, how can you express a degree of belief on a scale from 0 to 1, since probabilities have values in this range. It is not very difficult to express a degree of belief at the extremes of the probability scale 0 or 1, but trying to express a belief between 0 and 1 could result in 'guessing'.

To express a person's degree of belief in an event X, we can use the well known example of the Wheel of Fortune in Figure 3.1 based on the work by Heckerman [10].

The wheel is symmetric except for the colored region, therefore it is equally likely for the wheel to stop in any position. A person's degree of belief can be expressed in terms of the area of the

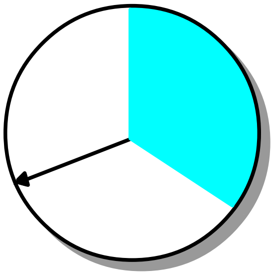


Figure 3.1: The probability wheel: a tool for assessing probabilities

colored region. The larger the colored region is, the more likely it is that the wheel will stop pointing somewhere in the colored region. For example, what is your probability that Barack Obama will be re-elected for president during the 2012 elections in the United States? First you can question yourself what the public opinion about Barack is at the moment, and how it will evolve in the coming 2 years. Then you enlarge or reduce the colored area until you think that the percentage of the colored area relative to the total area of the wheel expresses the probability that Barack Obama will be re-elected. The process of measuring a degree of belief is commonly referred to as *probability assessment*, which is often done in Management Science, Operations Research and Psychology literature [10].

The actual probability that Barack Obama will be re-elected can be 0.60 or 0.62 and we can question ourselves whether somebody is able to express a degree of belief in such level of accuracy. In most cases, this is not possible but nevertheless these probabilities can be used to make decisions.

To illustrate Bayesian Probability more formally, consider the well known coin tossing example[1]. Since there is a lot of literature to be found on Bayesian Networks, there seems to be a general notation for Bayesian Probabilities. The formal notation is based on the work by Heckerman [10]. His work and notation is highly accepted and often cited in literature.

If we toss a coin, it can hit the ground facing *heads* or *tails*. Suppose that we toss this coin $N + 1$ times, making sure the physical conditions remain the same during this experiment, and record the outcomes heads and tails. This example will be called the coin-toss example in this thesis from now on. Suppose that we are interested in the probability of the $N + 1$th toss being heads. In classical probability theory, there is a physical probability of a toss resulting in heads. By observing N tosses of the coin, a classical probability theorist will estimate the probability of a result being heads using criteria such as low bias and low variance. This estimate can be used to estimate the $N + 1$th toss.

In Bayesian Probability theory, we also acknowledge a certain physical probability of a coin toss resulting in heads, but this probability is expressed as a Bayesian Probability. To explain this formally, there is a need for a notation:

A variable in Bayesian Probability is denoted with a upper-case letter (e.g., $X, Y, X_i, \Theta$) and its value in lower case (e.g., $x, y, x_i, \theta$). The corresponding set of variables is denoted in bold-face upper-case (e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{X_i}$) and the corresponding lower-case (i.e.,$\mathbf{x}, \mathbf{y}, \mathbf{x_i}$) to denote an assignment (or configuration) of value to each variable in a given set. Now we can define Bayesian probability as

[1] This example is taken from Howard (1970)

equation 3.1:

$$p(X = x|\xi) = p(x|\xi), \tag{3.1}$$

in which the probability $p$ is calculated when $X = x$ of a person with state of information $\xi$. $p(x|\xi)$ can also be used to denote a Probability Density Function (PDF) or Probability Mass Function (PMF).

If we go back to the coin-toss example, we can describe the problem as follows. Lets denote the set of observations as: $D = \{X_1 = x_1, ..., X_N = x_N\}$ with $N$ as the number of observations. The coin toss problem can then be solved by computing $p(x_{N+1}|D, \xi)$ from $p(\theta|\xi)$ in which $\theta$ represents the physical probability of the coin resulting in heads after tossing it. The PDF for $\Theta$, given $D$ and background knowledge $\xi$, can be expressed by use of the Bayes rule as follows:

$$p(\theta|D, \xi) = \frac{p(\theta|\xi)p(D|\theta, \xi)}{p(D|\xi)}, \tag{3.2}$$

where

$$p(D|\xi) = \int p(D|\theta, \xi)p(\theta|\xi)\, d\theta, \tag{3.3}$$

since by the law of total probabilities:

$$\begin{aligned} p(D|\xi) &= \frac{p(D \cap \xi)}{p(\xi)} \\ &= \int \frac{p(D \cap \xi \cap \theta)}{p(\xi)} d\theta, \end{aligned}$$

by doing the mathematical reformulations:

$$p(\theta|\xi) = \frac{p(\theta \cap \xi)}{p(\xi)},$$

$$p(D|\xi) = \int \frac{p(D \cap \xi \cap \theta)}{p(\theta \cap \xi)} p(\theta|\xi) d\theta.$$

The term $p(D|\theta, \xi)$ can be expressed as the likelihood function for binomial sampling, since the observations in D are mutually independent and the probability of heads is always $\theta(1 - \theta)$. Equation 3.2 becomes:

$$p(\theta|D, \xi) = \frac{p(\theta|\xi)\theta^h(1 - \theta)^t}{p(D|\xi)}, \tag{3.4}$$

where $h$ and $t$ are the number of observed heads and tails respectively. In Bayesian Probability Theory, $p(\theta|\xi)$ is the *prior* and $p(\theta|D, \xi)$ is the *posterior* for $\Theta$; the probability that the coins faces heads. The probability that the $N + 1$th toss will result in heads can then be computed as follows:

$$\begin{aligned} p(X_{N+1} = heads|D, \xi) &= \int p(X_{N+1} = heads|\theta, \xi)p(\theta|D, \xi)d\theta \\ &= \int \theta p(\theta|D, \xi)d\theta \equiv E_{p(\theta|D,\xi)}(\theta), \end{aligned}$$

where $E_{p(\theta|D,\xi)}(\theta)$ denotes the expectation of $\theta$ with respect to the distribution $p(\theta|D,\xi)$ as explained in Heckerman [10]. Before we can assign probabilities to the prior, we need to choose or calculate a distribution for $\Theta$. A common approach in binomial sampling is to assume that the prior distribution is a *beta* distribution out of convenience [10]:

$$P(\theta|\xi) = Beta(\theta|\alpha_h, \alpha_t) \equiv \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)}\theta^{\alpha_h-1}(1-\theta)^{\alpha_t-1}, \tag{3.5}$$

where $\alpha_h$ and $\alpha_t$ both are greater then 0, $\alpha = \alpha_h + \alpha_t$, $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. The beta distribution is a family of continuous probability distributions defined on the interval $(0,1)$ with parameters $\alpha, \beta$ which define its shape as can be seen in Figure 3.2.

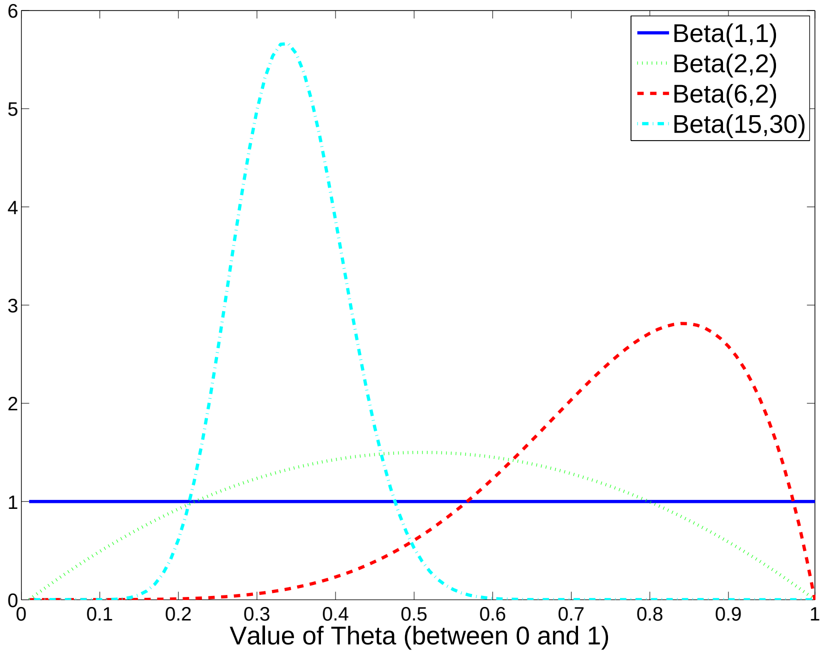


Figure 3.2: An example set of beta distributions

The posterior distribution $p(\theta|D,\xi)$ is also a beta distribution:

$$Beta(\theta|\alpha_h + h, \alpha_t + t). \tag{3.6}$$

Since the beta distribution is assumed for the binomial sampling problem, we can make use of the simple expression for $E_{p(\theta|D,\xi)}(\theta)$ as follows:

$$\int \theta Beta(\theta|\alpha_h,\alpha_t)d\theta = \frac{\alpha_h}{\alpha_h+\alpha_t} = \frac{\alpha_h}{\alpha}.$$

Now we can turn back to our coin-tossing example. Given that we have a beta prior, the probability of the $N+1$th toss being heads can be written as:

$$p(X_{N+1} = heads|D,\xi) = \frac{\alpha_h+h}{\alpha+N}. \tag{3.7}$$

From now on, there are numerous ways to assign probabilities for a toss resulting in heads. We can estimate the probability of a toss being heads based on the the previous tosses. After each toss, the probability can be reassessed. If these probabilities are estimated with, for example, the help of the wheel of fortune as explained previously, the values for $\alpha_h$ and $\alpha_t$ can be calculated. This is the method of *imagined future data*. More details on methods for probability assignment can be found in [10],[45] and [3].

The beta distribution is not only useful for binomial sampling problems such as this coin tossing example, but can also be used for more complex problems. The probability densities of these more complex problems can then be expressed by mixtures of beta distributions. The distribution for multinomial sampling is called the Dirichlet distribution and the beta distribution can actually be seen as a special case of the Dirichlet distribution. Lets say that we have a dataset $D = \{X_1 = x_1, ..., X_N = x_N\}$ with the set of statistics $N = \{N_1, ..., N_r\}$ where $N_i$ is the number of times $X = x^k$ in $D$. The prior used with multinomial sampling is the Dirichlet distribution:

$$p(\theta|\xi) = Dir(\theta|\alpha_1,...,\alpha_r) \equiv \frac{\Gamma(\alpha)}{\prod_{k=1}^{r}\Gamma(\alpha_k)} \prod_{k=1}^{r} \theta_k^{\alpha_k-1}, \tag{3.8}$$

where $\alpha = \sum_{i=1}^{r}\alpha_k$ and $\alpha_k > 0$, $k = 1,...,r$. The posterior distribution used for multinomial sampling can also be expressed as a Dirichlet distribution:

$$p(\theta|D,\xi) = Dir(\theta|\alpha_1+N_1,...,\alpha_r+N_r). \tag{3.9}$$

The same probability assignment methods as explained earlier in this Chapter can be used which yield Bayesian Probabilities that differ from the classical statistic approach. The classical statistician believes that $\theta$ is fixed and unknown [10]. It can be estimated with the help of a dataset $D$. To evaluate this estimator, the expectation and variance can be computed as follows given that $E(X) = \sum_i x_i p(x_i)$ and $Var(X) = E[(X-\mu)^2]$:

$$E_{p(D|\theta)}(\theta^*) = \sum_D p(D|\theta)\theta^*(D) \tag{3.10}$$

and

$$Var_{p(D|\theta)}(\theta^*) = \sum_D p(D|\theta)(\theta^*(D) - E_{p(D|\theta)}(\theta^*))^2). \tag{3.11}$$

A good estimator is usually an estimator which is consistent, robust and has a low bias and variance [10]. A commonly used estimator is the Maximum-Likelihood (ML) estimator:

$$\theta^*_{ML}(D) = \frac{N_k}{\sum_{k=1}^{r} N_k}, \tag{3.12}$$

since the ML estimator is known to be unbiased.

In the Bayesian Approach, the dataset $D$ is fixed and we try to answer how this dataset could have been generated. The variable $\theta$ is still unknown, and our expectation of $\theta$ with respect to the dataset D (our posterior beliefs) can be calculated as follows:

$$E_{p(\theta|D,\xi)}(\theta) = \int \theta p(\theta|D,\xi)d\theta, \tag{3.13}$$

which clearly differs from equation 3.10.

## 3.2 Bayesian Networks

At this stage, it is necessary to explain the term: *Bayesian Networks*. What are Bayesian networks and what can we do with it? In literature there are different formulations for the term *Bayesian Networks*, but in essence, they represent the same. The term 'Bayesian Networks' was first coined in 1988 by Pearl [24] as follows:

> *"A probabilistic network (also referred to as a belief network, Bayesian network, or, somewhat imprecisely, causal network) consists of a graphical structure, encoding a domain's variables and the qualitative relationships between them, and a quantitative part, encoding probabilities over the variables"*[24].

This could be seen as the most important publication in Bayesian Networks as stated by Russel and Norvig [31] in 2002. Since there are multiple definitions present in literature for Bayesian Networks, from this point on the term Bayesian Network in this thesis stands for the union of belief networks, causal networks, and all other terms which represent the same.

A somewhat shorter and more recent definition for the term Bayesian Networks is proposed by Heckerman [10] in 1995 as follows:

> *"A Bayesian Network is a graphical model for probabilistic relationships among a set of variables"*[10].

A somewhat broader definition is proposed in 2004 by K. Korb and A. Nicholson as follows:

> *"A Bayesian network is a graphical structure that allows us to represent and reason about an uncertain domain"* [17].

In this formulation of a Bayesian Network the term *reasoning* is introduced, which widens our view on Bayesian Networks. By continuously updating the knowledge and beliefs of a Bayesian network, we can use Bayesian Networks for reasoning about uncertainties in terms of probabilities, which is of course interesting if we want to use it for predictions.

Heckerman [10] explains four main advantages for using Bayesian Networks:

1. BN can readily handle incomplete datasets
2. BN allow one to learn about causal relationships
3. BN facilitate the combination of domain knowledge and data
4. BN offer an efficient and principled approach for avoiding the over fitting of data.

In the area of Traffic predictions, we are typically interested in looking for relationships among a large number of variables. These are variables like speed and density measurements at different locations at different times. Missing values could be present in traffic data and there is also a lot of domain knowledge in the traffic research area. A Bayesian Network can represent this knowledge in variables in a graphical model that efficiently encodes the joint probability distribution.

## 3.3 Notation and example

In this stage it is essential to define a notation for a Bayesian Network which will be referred to in the rest of this thesis. The formal definition given in this thesis is the general formulation which is also given in [10] or [13] and can be seen as the general formulation in literature. Lets say that we have a set of variables which are situated in a network structure $S$ such that they form a Bayesian Network.

A Bayesian network consists of:

- A set of variables $X = X_1, ..., X_n$.
- A set $P$ of local probability distributions associated with each variable in $X$. $P_i$ can be calculated with equation 3.14.
- A network structure $S$ which is an acyclic directed graph. $S$ encodes a set of conditional independence assertions about variables in $X$ and the nodes in S are a one-to-one correspondence with the variables in $X$.

Together the network structure $S$ and the set $P$ define the joint probability distributions for set $X$. If we define $Pa_i$ to denote the parents of the corresponding variables of node $X_i$ in $S$, then the joint probability distribution (JPD) for $X$ is given by:

$$p(x) = \prod_{i=1}^{n} p(x_i|pa_i). \tag{3.14}$$

The probabilities are Bayesian probabilities if they origin from prior knowledge alone. If they are learned from the data they are physical probabilities and then they may contain uncertainties.

To create a Bayesian Network, Heckerman [10] proposes the following steps:

1. Identify the goals of the modeling.
2. Identify relevant observations.
3. Select most important observations.
4. Organize the observations into mutually exclusive variables with exhaustive states.

To explain how Bayesian Networks can be used in traffic research, an example will be given. Lets say that a driver steps into his car and questions himself, would there be a traffic congestion? Then he would first think of what time it is to estimate whether there are more people traveling at this time. Further, he would recall whether it is a holiday to estimate how much commuter traffic will be on the road. It is also important to think of possible announced road works and weather forecasts to estimate the probability of slow traffic. Lets say that we have the next set of variables: ***T**ime*, ***H**oliday*, ***V**isibility*, ***R**oad work* and *traffic **C**ongestion*. The *traffic congestion* variable is the effect variable, whereas *time*, *holiday*, *visibility* and *road work* are cause nodes. These variables can be structured in a Bayesian Network for which an example is given in Figure: 3.3. The local probability density functions, usually structured in Conditional Probability Tables (CPTs), can be learned from historical data. This process is called: *Bayesian Network Learning*, and is explained in section 3.4.

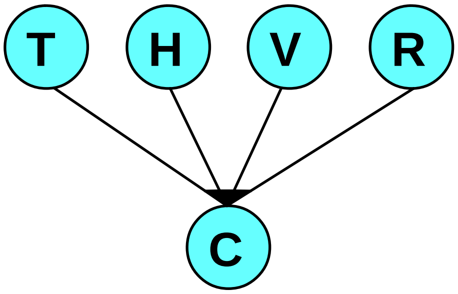


Figure 3.3: Bayesian Network example for traffic congestions

If these probabilities are learned, the network could be used by setting evidence for some nodes, and calculate the probabilities for the unknown nodes. This process is called *inference*. *Probabilistic inference* is the process of computing a probability of interest from a Bayesian Network model. Because a Bayesian Network for $X$ determines a joint probability density (JPD) for $X$, such that a probability of interest can be calculated. For example, from the Bayesian Network in Figure 3.3 the probability of a traffic congestion could be calculated as follows:

$$p(c|t,h,v,r) = \frac{p(c,t,h,v,r)}{p(t,h,v,r)}. \tag{3.15}$$

This method is not very practical for networks with many variables [10], but the conditional independencies encoded in the Bayesian Network can be exploited to make this computation more efficient [10]. There are numerous probabilistic algorithms for Bayesian Networks and more details can be found in [10].

## 3.4 Learning Bayesian Networks

There are numerous ways to learn Bayesian Networks and to find the conditional probability tables (CPTs). Generally, methods for learning CPTs combine prior knowledge with data to produce improved knowledge [10]. Since explaining all computational details here for learning CPTs would be too much, the readers are referred to read the work of Heckerman [10] for more details. A brief description of learning Bayesian Networks is given in the remaining of this section.

If there are no missing values in the data, there are multiply ways to learn the probabilities from data. But, of course, usually there are unobserved cases and thus missing values. In the traffic example above, it is likely that not all combinations of the variables *time*, *holiday*, *visibility*, *road work* and *traffic congestion*, have been seen in history.

In general, the exact computation of CPTs when there is missing data is intractable [10]. Therefore, usually approximation algorithms are used for incomplete data.

Monte-Carlo methods are often used for this approximation and they are likely to yield accurate results, provided one is willing to wait long enough [10]. Another approach is the Gaussian Approximation, where variables are often approximated as a multivariate-Gaussian distribution.

If the number of samples in the dataset is large enough, the most often used technique for learning probabilities is the Expectation Maximization (EM) algorithm. The EM algorithm is an iterative algorithm which consists of an expectation step and a maximization step [10]. This algorithm estimates probabilities for missing combinations in the CPTs as well and is often used in literature.

After having seen the fundamental theory of traffic and the basics of Bayesian Networks, it is time to analyze the traffic data we have. The next Chapter analysis the data, which is used for our proposed Bayesian Network later on in this thesis.

# Part II

# Data Exploration Part

# Chapter 4

# Data Acquisition and Preparation

This Chapter describes how traffic data for this thesis is acquired and prepared. There are several data acquisition methods available. The most important and most used traffic data is acquired by inductive loop detectors (ILDs) which are situated on the roads itself. A description of traffic data acquisition, and especially about ILDs is given in section 4.1. Raw data is often not directly useful for analysis purposes, since it often contains missing or corrupted values. Therefore, data preparation is an important process during the development of data driven models. Section 4.2 describes how missing values can be handled in traffic data. In the end, section 4.3 describes how traffic data can be averaged from low detailed data into more meaningful high level data.

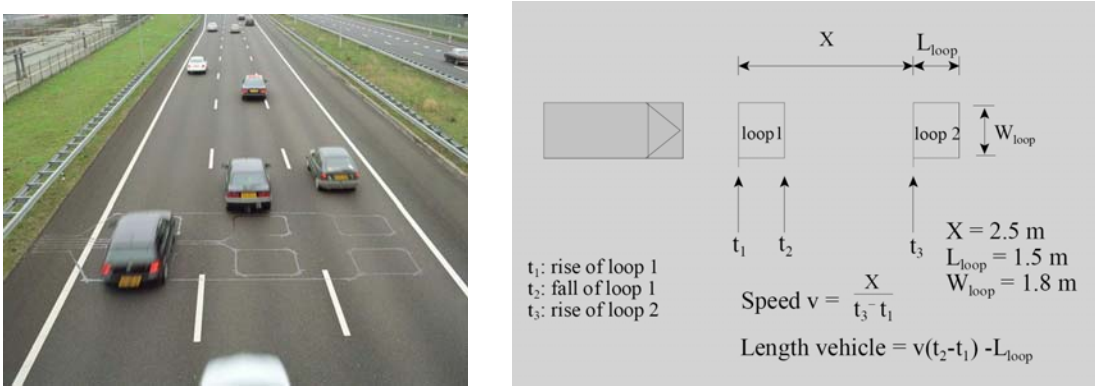
## 4.1 Traffic data acquisition

Real traffic data is one of the most important elements in analyzing and improving traffic systems. With real data there is a better representation of the real world compared to simulated data. There are many ways to acquire traffic data such as speed, intensity or density measures. Only two of these three variables should be known and the other can be calculated, as explained previously in Chapter 2.

A convenient overview of data acquisition methods can be found in [11] and some important aspects are briefly discussed here. The oldest technique to acquire traffic data is the *pneumatic tube*. This is a hollow rubber tube that sends a pressure wave when a vehicle runs over it. Although this method delivers quite accurate results, it is not very reliable as the tubes easily get damaged [11].

Measuring traffic flow is mostly done with induction loop detectors (ILD's) because of their wide spread availability [43]. There are two different kinds of ILDs: *single* and *dual* ILD's, which both measure changes in induction if a vehicle passes. Single loop detectors are standardly used in the USA because they are inexpensive. It is not possible to measure speed directly, since single loop detectors can only do a single measurement on a passing vehicle. To measure speed, it must be measured how many distance is traveled between two measuring locations in a certain amount of time. With advanced single ILD's it is possible to obtain disaggregated information for individual vehicles by using advanced high speed scanning loop detector cards that capture more detailed inductance changes over the loop [23]: the *vehicle signature*. Although speed cannot be measured directly by single ILDs, there is a way to estimate the speed using this signature. Details about measuring traffic characteristics with single ILD's can be found in [23].

Dual ILDs are capable of doing two measurements, from which the speed can easily be calculated. The Dutch freeways are equipped with dual ILDs. The data collection for the road network in the Netherlands is managed by the Dutch Ministry of Transport, Public Works and Water Management with the **MONI**toring **CA**sco (MONICA) system [41]. An example of a dual ILD is visualized in Figure 4.1(a) and its loop configuration as applied in the Dutch Freeways is visualized in Figure 4.1(b).



(a) Photo of a dual ILD setting for 3 lanes [36]

(b) ILD setting on Dutch freeways [11]

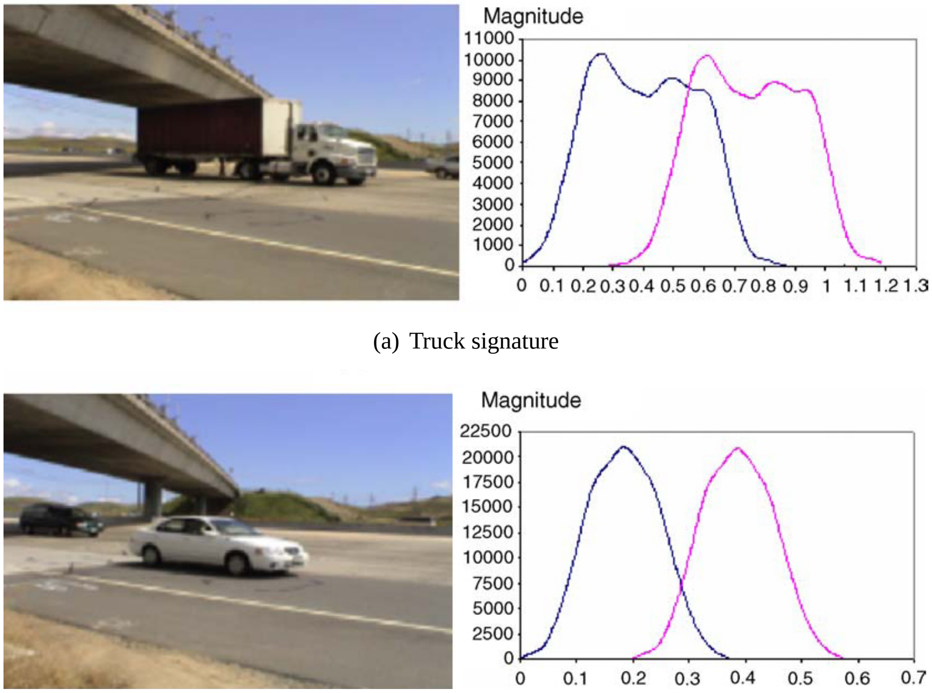
Figure 4.1: ILD example on Dutch freeways

From now on in this thesis, the term ILD denotes these dual ILDs at Dutch Freeways. An example of a vehicle signature from a dual ILD is illustrated in Figure 4.2. This figure represents a signature of a truck and a car. This figure shows that the type of the vehicle can be detected by analyzing the patterns of their signatures. There are some ILDs in the Netherlands which are capable of doing this, but they are not widely spread and therefore this dataset is sparse.

The MONICA system collects traffic data from widely spread dual ILDs at every freeway in the Netherlands. In general, there is a ILD every 500 meter at the Dutch freeways [41]. The MONICA system consists of dual inductive loop detectors which measure point mean speeds (kilometer/hour), point mean intensities (vehicles/hour) and contains the messages on **D**ynamic **R**oute **I**nformation **P**anels (DRIPs). Messages on DRIPs contain possible lane closures, temporal speed limits, etc. An example of a DRIP is presented in Figure 4.3.

MONICA data is delivered in ASCI data files which are in the ADY-format. A tutorial and description of this format can be found in [40]. MONICA data is the most important data source for this thesis. Details of how this data is prepared and used for this research are described in the remaining of this Chapter.

Other measure instruments are infrared detectors or instrumented vehicles. Infrared detectors detects passing vehicles when a light beam is interrupted. In this way, individual passing times, headways and intensity can be measured. Instrumented vehicles are sometimes used as moving observers to measure traffic characteristics. This is a rather expensive approach and is of course only useful if



(a) Truck signature

(b) Car signature

Figure 4.2: Vehicle signatures from ILDs [22]

the location of the instrumented vehicle is representative enough to represent the traffic situation in general.

There are also some prospective methods for traffic data acquisition. For example, floating car data (FCD) data acquisition methods to get individual vehicle data are already in development. These methods use GPS positioning to acquire traffic measurements of single cars. There are also some data acquisition methods based on wireless networks [14], but these methods are not widely available yet [44].

## 4.2 Handling missing values in traffic data

On average, the MONICA system has a missing value percentage of 12% for a particular time instant due to maintenance, power or communication failure, or indcidents and accidents. In a significant number of cases there are even occasions in which over 20% of the measurements are either missing or corrupt [41]. Especially at times of congested traffic it is extremely hard to acquire reliable data, since the traffic flow could be low and only few cars are passing the detectors.

The most simple approach for filling in missing data is to replace the missing values by the mean. Traffic propagates with a harmonic behavior through a network. Taking just the mean value would disturb this harmonic process. Lets say that a collection of cars are braking and the speed is decreasing. A missing value here cannot be the mean speed, because this will result in a spike in the speed graph.

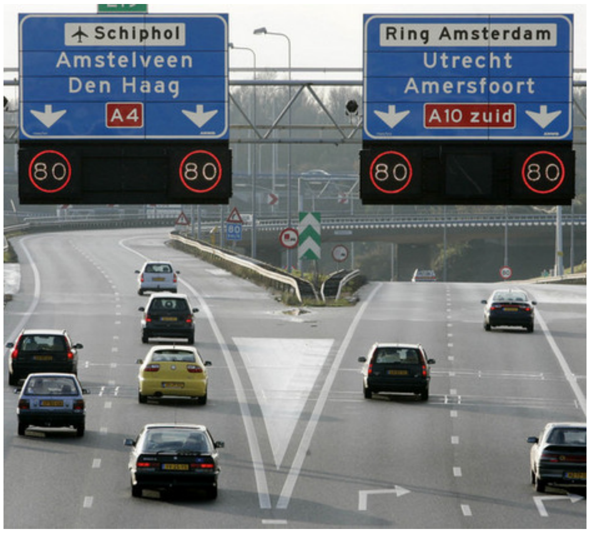


Figure 4.3: Dynamic Route Information Panel (DRIP)

A car cannot brake and accelerate at the same time, so the speed graph becomes physically impossible.

Another simple approach would be to use linear interpolation with, for example, a nearest neighbor technique. The traffic measurements around a missing value, i.e. the ’neighbor’ values, should be related to the traffic measurement in between which is missing. The term ’neighbor’ here denotes measurements close by in time or location. Therefore, linear interpolation by making use of the nearest neighbor measurements is not a bad idea. Although, when the missing data gap becomes to large, it would be difficult to get good results.

A state-of-the-art algorithm for filling in these missing values is the Treiber and Helbing filter [38]. Treiber and Helbing proposed a method to obtain spatio temporal information from aggregated data of stationary traffic detectors, for example inductive loop detectors (ILDs).

The filter, developed by Treiber and Helbing, filters out small-scale fluctuations and adaptively takes the main propagation direction of information flow into account. The filter is nonlinear and adapts itself to the traffic situation. The three important properties of the filter [38] are:

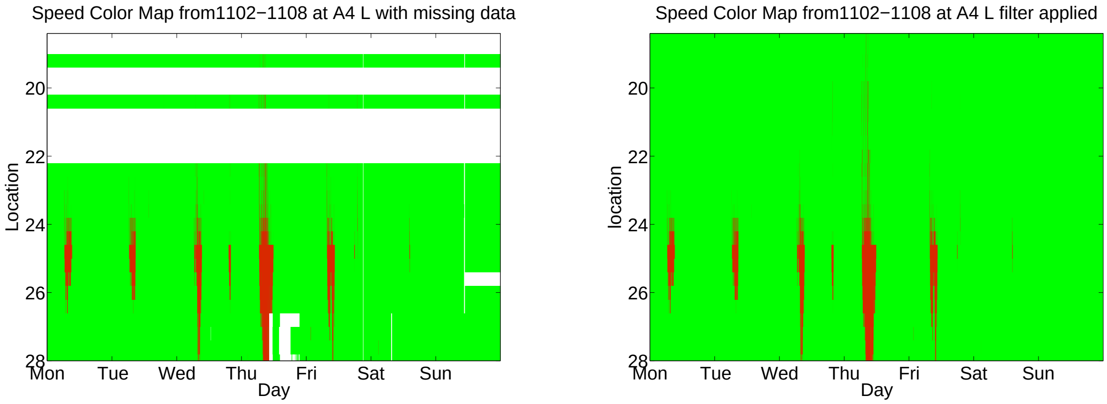
1. In case of free traffic flow: perturbations in speed or intensity move essentially along the direction of traffic flow. Research shows that these perturbations travel at 80% of the desired velocity on empty roads [38].

2. In case of congested traffic, traffic disturbances in speed or intensity move against the direction of traffic flow with a constant speed of 15 km/h.

3. The filter smooths out fluctuations.

The Treiber and Helbing filter has been implemented by the faculty of Civil Engineering at Delft University of Technology and optimized by the Netherlands Organization for Applied Science (TNO) and used to prepare the data for this thesis. The performance of this filter will be illustrated with the next example.

Lets inspect the first week of November in 2009. Figure 4.4(a) shows a space time color map of speed measurements between hectometer location 18 and 28 on the A4 left side. The vehicles travel from location 28 in the direction of 18. Seen from the graph, the vehicles travel from the bottom to the top. The white colored areas are the missing values, the green color means speed values above 60 km/h and the read color means speed values under 60 km/h. It becomes clearly visible that from Monday until Friday there are morning traffic congestions and on Wednesday there is an evening traffic congestion as well. Saturday and Sunday seem to have no congestion at all, which should not come as a surprise.



(a) Unfiltered speed plot

(b) Filtered speed plot

Figure 4.4: Application of the Treiber and Helbing filter

Figure 4.4(b) shows the same information, but now after the Treiber and Helbing filter is applied. The traffic congestions are nicely smoothed over the missing values areas and the spatio temporal information seems to be kept. Another interesting point, is that the red areas (the traffic congestions) seem to propagate against the direction of the traffic flow in time. This corresponds with the properties of traffic, as explained earlier.

## 4.3 Lane aggregation

The MONICA system measures speed and intensity per lane. The road network in the Netherlands typically consists of roads with 2, 3 or 4 lanes. If every lane would be investigated separately, there would be 21.354 ILDs to inspect in the Netherlands. Not every lane is occupied by a vehicle all the time, so there will be a lot of missing values in this case.

Therefore, it would be a better idea to aggregate the lanes to road measurements. This decreases the number of measurements from 21.354 to 10.183 and there are less missing values since the probability for a vehicle being present on a road is higher than for being present on a certain lane. The main question now is: how can these measurements be aggregated?

There are three methods to describe the central tendency of a dataset: *arithmetic mean*, *harmonic mean* and *geometric mean*. These methods in general all calculate the expected value or the *mean* of the dataset, but each with its own characteristics. The most common method is the arithmetic mean and can be calculated as follows:

$$AM = \frac{1}{N}\sum_{i=1}^{N} a_i, \tag{4.1}$$

in which $a_i$ are the data points in dataset $D = a_i, ..., a_n$. For example lets say we have an apple tree, which gives us 100, 150, 170 and 210 apples in the years 2007, 2008, 2009 and 2010 respectively. The annual growth percentage on the number of apples is 50%, 13.3% and 23.5% between 2007-2008, 2008-2009 and 2009-2010 respectively. This example is called the *apple example* for the rest of this thesis. The average annual growth in this example would be $(1.50\% + 1.133\% + 1.235\%)/3 = 1.2893\%$. If we would use this mean to calculate to total number of apples after 3 years starting with 100 apples, we would get: $100 \cdot 1,29^3 = 215$ apples, which is 5 apples more then we really have in 2010. At this stage it might look over simplistic to explain how to average speeds, but it will become clear why this is important later on in this section.

The geometric mean differs from the arithmetic mean and is described as follows:

$$GM = \left(\prod_{i=1}^{N} a_i\right)^{\frac{1}{N}}. \tag{4.2}$$

If we would use this method to calculate the average annual growth percentage based on the *apple* example explained previously, we would get: $\sqrt[3]{1.50 \cdot 1.133 \cdot 1.235} = 1.28$. If we take this average to calculate the total number of apples after 3 years starting with 100 apples, we would get: $100 \cdot 1.28^3 = 210$ apples, which is exactly the number of apples we will have in 2010. This example indicates that the arithmetic mean has a tendency to over estimate the proportional growth, whereas the geometric mean takes this grow into account and is therefore in general more appropriate for describing proportional growth (for example interest computations in the business field). It becomes clear by this example that the choice for an average method matters!

The harmonic mean is typically used in situations where an average of rates or rations needs to be calculated and is denoted as follows:

$$HM = \frac{N}{\sum_{i=1}^{N} \frac{1}{a_i}}, \tag{4.3}$$

where $a_i > 0$ for all $i$. Speed can be seen as a ratio when it is used as to calculate travel time. A travel time $\tau$ in general can be calculated as follows:

$$\tau = \frac{d}{V}, \tag{4.4}$$

where $d$ denotes a certain distance and $V$ denotes a certain speed. The speed $V$ is the denominator in equation 4.4 so it can be seen as a ratio. This can be illustrated with the next example. Lets say that a car $C$ is traveling on a road for 5 km with a speed of 30 km/hour. After this trip, the car immediately travels another 5 km with a speed of 90 km/hour. Calculating the travel time for the whole trip is simple: $\frac{5}{30} + \frac{5}{90} \cdot 60 = 13$ minutes. The arithmetic mean of the speed values is $\frac{30+90}{2} = 60$ which yields a travel time of $\frac{5}{60} + \frac{5}{60} \cdot 60 = 10$ minutes, which is convincingly lower then the real travel time. The harmonic mean, which is typically used for averaging rates, is: $\frac{2}{\frac{1}{30}+\frac{1}{90}} = 45$. Using the harmonic mean to calculate the travel time yields in: $\frac{5}{45} + \frac{5}{45} \cdot 60 = 13$ minutes, which is the same as the real travel time as calculated earlier. At this stage, it seams logically to choose the harmonic mean method for averaging the variable speed. For this example, the harmonic mean seams to calculate the correct average speed, if speed is used to calculate travel times later on. But this example consists of equidistant travel distances, so how does this work for non-equidistant travel distances?

Lets investigate the travel time problem a little more mathematical, to see what the underlying principles are. Lets consider a trip from location A to B as illustrated in Figure 4.5. The travel distances of the underlying travels and its corresponding speed values are denoted as $d_i$ and $V_i$ respectively where $1 \leq i \leq N$. If a car travels from location A to B, the total travel time $\tau$ can be expressed as follows:

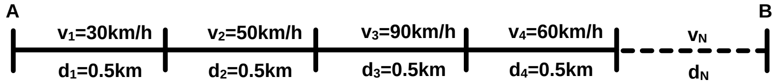


Figure 4.5: Trip information

$$\tau = \sum_{i=1}^{N} \frac{d_i}{V_i} = \frac{D}{\bar{V}} = \frac{d_1}{V_1} + \cdots + \frac{d_N}{V_N}, \tag{4.5}$$

where $\bar{V}$ denotes the mean speed and $D$ denotes the total distance $D = d_1 +, \cdots, +d_N$. From equation 4.5 it follows that the mean speed $\bar{V}$ can be expressed as follows:

$$\bar{V} = \frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} \frac{d_i}{V_i}} = \frac{D}{\sum_{i=1}^{N} \frac{d_i}{V_i}}. \tag{4.6}$$

If distance $d_i$ is a certain constant for all $i$, i.e. the travels are equidistant, then $\frac{D}{N} = d$ and so $N = \frac{D}{d}$ which results in:

$$\bar{V} = \frac{D}{d} \cdot \frac{1}{\sum_{i=1}^{N} \frac{1}{V_i}} = \frac{N}{\sum_{i=1}^{N} \frac{1}{V_i}}, \tag{4.7}$$

which is actually the harmonic mean as given by equation 4.3. Therefore, the harmonic mean method gives the correct speed average values when its values are used for travel time calculations. Since the eventual variable of interest in most traffic research is travel time, it is decided to take the harmonic average of speed values. When speed is measured at different ILD's at different locations, an harmonic average can be calculated under the assumption that the locations are equidistant. Since the ILD's are roughly evenly spread at the Dutch Roadway system with distances of around 500 meters, the harmonic mean method is used to calculate speed average for the rest of this thesis. More details about averaging speed can be found in [41].

# Chapter 5

# Sensibility Analysis

In this Chapter we perform an explorative analysis on traffic data. At a first look, congestion seem to be a random process. Traffic queues are coming and going at different locations at different times. Our goal is to predict traffic congestions. To get more grip and understanding on the data, we take a closer look at some congestions. Section 5.1 introduces the most congested areas of the Dutch road network and gives an impression of the traffic situation in the Netherlands. To get an impression of the dependency between traffic measurements, a correlation analysis has been conducted in section 5.2. To find out which parameters play a role, it has been investigated which parameters influence the probability of a congestion in section 5.3.

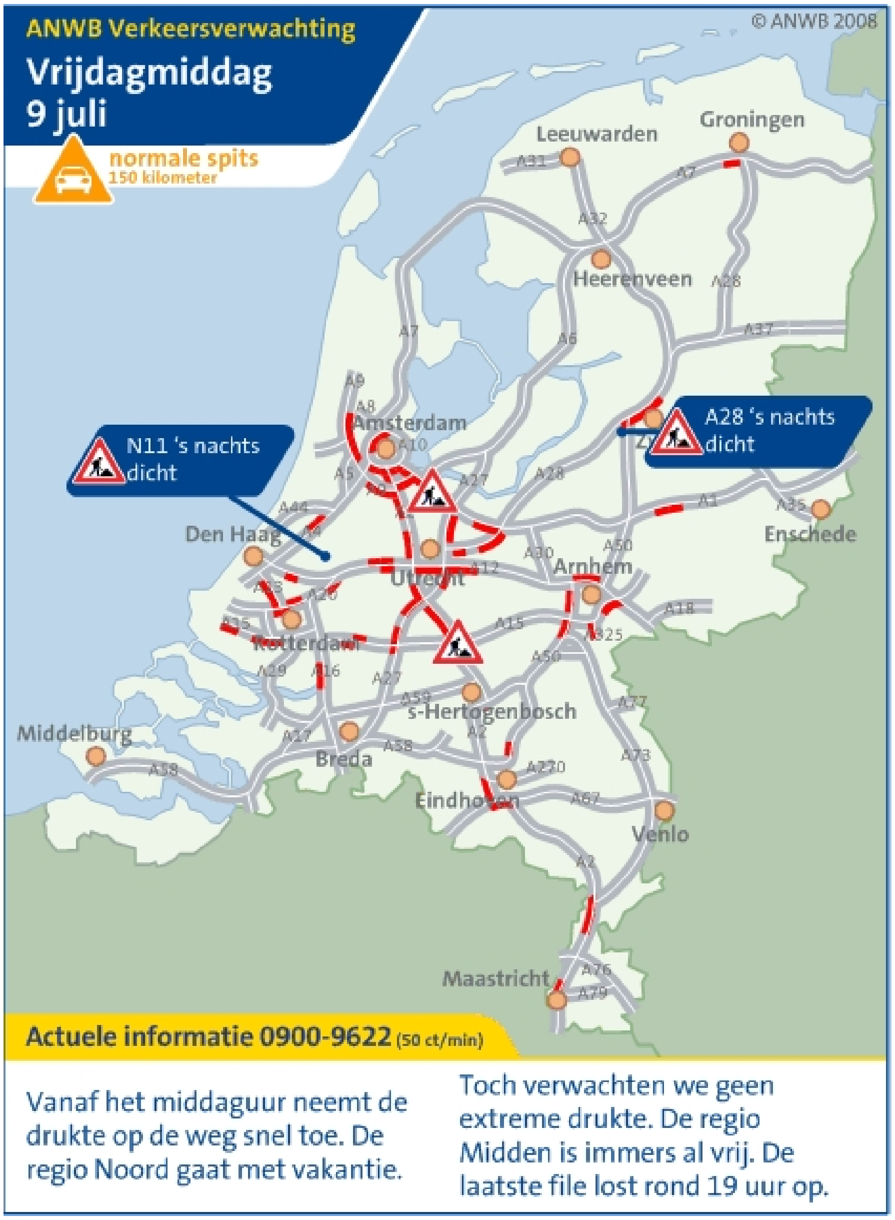
## 5.1 Traffic problem areas

To get an impression of the size and impact of traffic congestions in the Netherlands, let us look to an overview of the Dutch road network on an average Friday afternoon. Figure 5.1 shows an overview of the expected congested traffic areas on the Dutch road network for Friday the 9th of July 2009. Overview maps like this are produced by the ANWB (The Dutch Organization for Drivers). The overview shows a normal rush hour situation, with around 150 kilometers of congested traffic in total. There are some roadwork locations and a lot of red tubes. A red tube denotes a traffic congestion on a particular side of the road.

The figure shows that the highest concentration of traffic congestions is in the central-west part of the Netherlands. This part is also called the *Randstad*, which is an agglomeration of the four largest Dutch cities: Amsterdam, Rotterdam, the Hague and Utrecht, and the surrounding areas. Almost half of the population of the Netherlands (7.1 million people) lives in this part.

The Randstad contains various motorways, most of them starting in Amsterdam and Rotterdam. The port of Rotterdam and Schiphol airport are large transport hubs and invite a lot of traffic on the freeways. The most important motorways in the Randstad are the A1, A2, A4, A7, A12, A15, A16 and the A20 and most of them suffer from severe congestion.

To get a better overview of these congestions, a ’bottleneck’ matrix can be calculated which indicates how many bottlenecks will be encountered on routes between 16 important cities in the Netherlands. This matrix is presented in Table 5.1. The number of encountered bottlenecks can deviate 1 or 2 from the real value, since it depends which road entry is taken. The routes are optimized for the



Vanaf het middaguur neemt de drukte op de weg snel toe. De regio Noord gaat met vakantie.

Toch verwachten we geen extreme drukte. De regio Midden is immers al vrij. De laatste file lost rond 19 uur op.

Figure 5.1: Bottleneck map

shortest travel time with Google Maps and the number of encountered bottlenecks is computed from the bottleneck map in Figure 5.1.

Table 5.1: Bottleneck matrix for major routes in the Netherlands

| From | Amsterdam | Arnhem | Breda | Den Haag | Eindhoven | Enschede | Groningen | Heerenveen | Leeuwarden | Maastricht | Middelburg | Rotterdam | ’s-Hertogenbosch | Utrecht | Venlo | Zwolle |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Amsterdam | 0 | 3 | 5 | 2 | 6 | 4 | 2 | 1 | 1 | 7 | 3 | 4 | 5 | 4 | 7 | 4 |
| Arnhem | 1 | 0 | 3 | 2 | 3 | 1 | 1 | 1 | 1 | 3 | 2 | 2 | 2 | 1 | 2 | 1 |
| Breda | 2 | 3 | 0 | 1 | 1 | 4 | 3 | 2 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 4 |
| Den Haag | 2 | 4 | 3 | 0 | 5 | 5 | 3 | 2 | 2 | 8 | 3 | 2 | 4 | 3 | 6 | 6 |
| Eindhoven | 3 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 2 | 1 |
| Enschede | 0 | 1 | 4 | 2 | 5 | 0 | 1 | 1 | 1 | 4 | 2 | 3 | 3 | 0 | 1 | 1 |
| Groningen | 0 | 3 | 3 | 1 | 6 | 3 | 0 | 0 | 0 | 5 | 3 | 3 | 5 | 0 | 3 | 0 |
| Heerenveen | 0 | 2 | 3 | 1 | 5 | 2 | 1 | 0 | 0 | 5 | 3 | 3 | 4 | 0 | 3 | 1 |
| Leeuwarden | 0 | 2 | 3 | 1 | 5 | 2 | 1 | 0 | 0 | 5 | 3 | 3 | 4 | 0 | 3 | 1 |
| Maastricht | 4 | 2 | 1 | 2 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| Middelburg | 2 | 0 | 0 | 1 | 1 | 3 | 3 | 2 | 2 | 4 | 0 | 0 | 0 | 1 | 2 | 4 |
| Rotterdam | 2 | 2 | 1 | 2 | 2 | 7 | 6 | 5 | 5 | 5 | 1 | 0 | 2 | 4 | 7 | 7 |
| ’s-Hertogenbosch | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 2 | 1 |
| Utrecht | 3 | 2 | 3 | 1 | 3 | 3 | 3 | 1 | 1 | 7 | 4 | 3 | 3 | 0 | 5 | 3 |
| Venlo | 3 | 0 | 0 | 2 | 0 | 3 | 3 | 3 | 3 | 2 | 0 | 3 | 0 | 1 | 0 | 3 |
| Zwolle | 3 | 1 | 4 | 2 | 6 | 2 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 1 | 2 | 0 |

If there are 6 or more bottlenecks on a route, the entry is colored in red. As becomes clear from this table, the largest number of encountered bottlenecks can be found on routes which have to pass the Randstad. From the outer regions in the Netherlands, it takes around 1.5-2 hours to reach the Randstad. Therefore, a prediction model which is only able to predict traffic for, lets say, the next 30 minutes is not sufficient. Travelers which are leaving from Groningen or Maastricht need to predict the traffic situation for their route, which is around 2 hours ahead. Therefore, a useful traffic prediction system should be able to predict at least horizons of 2 hours, given that travelers are leaving immediately. If travelers need traffic prediction information a few hours earlier, or maybe even one day before, there is a need for reliable longer term traffic predictions. Therefore, we need to explore the traffic data to investigate the sensibility of the data. We start with an investigation of correlations between traffic measurements in the next section.

## 5.2 Correlation in traffic data

An accurate prediction systems demands extensive knowledge of the data. In case of traffic predictions, there should be some consensus about how traffic data is related. For example, if a vehicle hits the brakes, it is likely that following cars will have to brake as well to avoid collision. For now, it is important to define locations and directions on the road.

**Definition 6.** ***downstream traffic*** *is traffic at a location further along the driving direction.*

**Definition 7.** ***upstream traffic*** *is traffic at a location against the driving direction.*

It is likely that upstream traffic measurements are correlated with downstream traffic measurements. In case of congestions, as explained in Chapter 4, downstream measurements could be correlated with upstream measurements.

At this early stage, correlations between traffic measurements at different locations are investigated to get an impression of the sensibility of the data. For this, MONICA data of Tuesday the 3rd of November 2009 has been selected on the road A13 between hectometer location 13.2 and 16 on the right side of the road. The Treiber and Helbing filter, as explained in Chapter 4, has been applied on the data to fill in the missing values.

Since the correlation between subsequent ILDs in situations of free flowing traffic is expected to be maximal, this is not interesting at this moment. Correlation research during rush hours is therefore more interesting. For this section, a morning rush hour is defined between 06:00 am and 10:00 am and the evening rush hour between 15:00 pm and 19:00 pm.

A situation overview of the A13 is presented in Figure 5.2. This location is chosen because there are no merging or exit lanes at this part of the A13. Therefore, it is expected that this data consists of less disturbances, compared to data of roads with merging or exiting traffic. The yellow push pins in this figure indicate the locations of the ILDs. It has been decided to investigate the ILDs of the second lane.

Table 5.2 presents more information on 8 ILDs which were selected for further research. Each ILD has a certain ID: a BPS code. There are different types of ILDs. These ILDs are TSW/TLP detectors, which measure speed and intensity for a certain location. All ILDs are located at the A13 and their specific location is given in the table.

Traffic measurements could be seen as collection of subsequent measurements ordered in time: a *time series*. Speed measurements are conducted by the MONICA system every minute. If a part of these measurements is selected for location A, it is likely that this signal is correlated with measurements of another location. These measurements as a function of time can be seen as a signal. The signals at different locations could be similar if they are corrected for a certain (travel) time shift. Measuring similarity of two (time-shifted) signals can be called: *cross-correlation* which is also known as the *sliding dot product* and is usually expressed in formula form as follows for a 2 signals between time $a$ and $b$:

$$(f * g)[n] = \sum_{m=a}^{b} f^*[m]g[n+m], \tag{5.1}$$

where $f^*$ denotes the complex conjugate of $f$ and $n$ denotes the time shift.

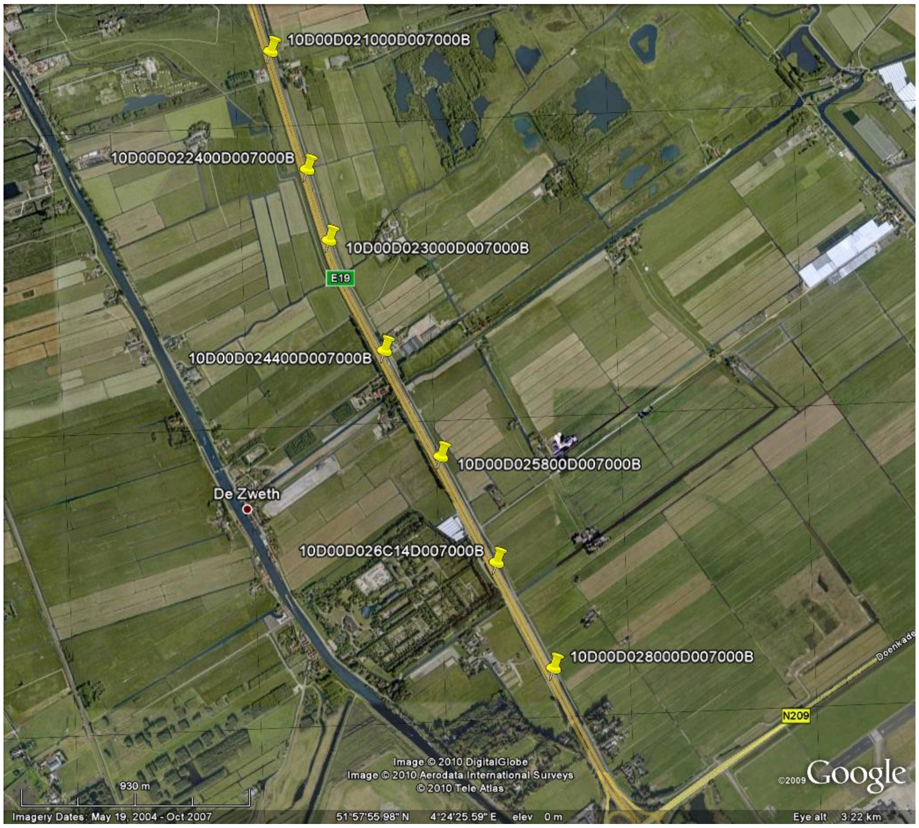


Figure 5.2: Situation straight road A13 with ILD's on lane 2 in direction R

Table 5.2: Inductive Loops on the A13 in direction R on lane 2

| Location | BPS code | BPS type | Road type | Road nr | Hectometer | Direction |
|---|---|---|---|---|---|---|
| A | 10D00D021000D007000B | [TSW]/[TLP] | 'RW' | 13 | 13.2000 | R |
| B | 10D00D022400D007000B | [TSW]/[TLP] | 'RW' | 13 | 13.7000 | R |
| C | 10D00D023000D007000B | [TSW]/[TLP] | 'RW' | 13 | 14 | R |
| D | 10D00D024400D007000B | [TSW]/[TLP] | 'RW' | 13 | 14.5000 | R |
| E | 10D00D025800D007000B | [TSW]/[TLP] | 'RW' | 13 | 15 | R |
| F | 10D00D026C14D007000B | [TSW]/[TLP] | 'RW' | 13 | 15.5200 | R |
| G | 10D00D028000D007000B | [TSW]/[TLP] | 'RW' | 13 | 16 | R |

Table 5.3 shows the cross correlation values between the different ILDs for the variable speed. As becomes clear, this variance is strong, but decreases if the distance between the ILDs increases. Especially in the evening the ILDs become less correlated for longer distances. This could be due to the fact that this part of the A13 has usually strong evening congestions, in which traffic shows strong harmonic behavior.

Table 5.3: Cross correlation for Speed between Inductive Loops on the A13 in direction R on lane 2

| ILDs | Distance(m) | xcorr day | xcorr morning rush hour | xcorr evening rush hour |
|---|---|---|---|---|
| A-B | 500 | 0.9936 | 0.9944 | 0.9723 |
| A-C | 800 | 0.9911 | 0.9917 | 0.9577 |
| A-D | 1300 | 0.9905 | 0.9874 | 0.9373 |
| A-E | 1800 | 0.9888 | 0.9861 | 0.9226 |
| A-F | 2320 | 0.9874 | 0.9848 | 0.9246 |
| A-G | 2800 | 0.9856 | 0.9833 | 0.9267 |

The results of the cross correlation computations for the variable intensity can be found in Table 5.4. For the intensity, we find the same effect as for speed, since the cross correlation decreases if the distance between the ILDs increases.

Table 5.4: Cross correlation for Intensity between Inductive Loops on the A13 in direction R on lane 2

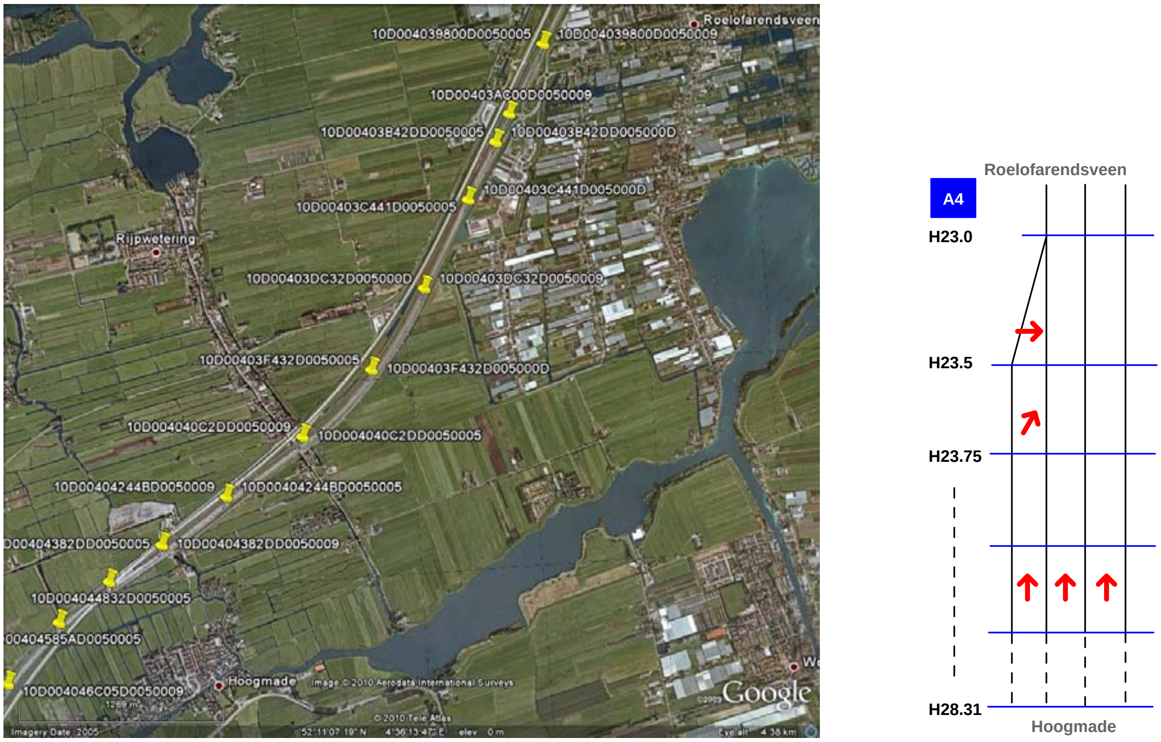
| ILDs | Distance(m) | xcorr day | xcorr morning rush hour | xcorr evening rush hour |
|---|---|---|---|---|
| A-B | 500 | 0.9841 | 0.9886 | 0.9803 |
| A-C | 800 | 0.9811 | 0.9848 | 0.9746 |
| A-D | 1300 | 0.9757 | 0.9801 | 0.9520 |
| A-E | 1800 | 0.9794 | 0.9792 | 0.9542 |
| A-F | 2320 | 0.9774 | 0.9778 | 0.9544 |
| A-G | 2800 | 0.9783 | 0.9770 | 0.9566 |

## 5.3 Sensibility of congestion parameters

Traffic propagates through a road network from origin to destination. It should be clear that the capacity of a road becomes an important aspect when the number of vehicles on that road increases. With a decreasing capacity and an increasing number of vehicles, traffic could be seen as a full bottle of water turned upside down. The water has to flow through the bottleneck where the capacity is usually lower then the rest of the bottle.

In the Netherlands, there is such a traffic bottleneck at road A4 between Hoogmade and Roelofarendsveen on the left side of the road. Of course, the definition of left or right side depends on the orientation. Therefore, the ILDs in the MONICA system have unique IDs (BPS codes) in which the side of the road is encoded. For the rest of this thesis, the directions which are encoded in these

IDs are chosen as notation. At this bottleneck location, heavy morning rush hours can be seen on working days because the road capacity decreases from 3 to 2 lanes around hectometer 23.5. Figure 5.3 illustrates the situation.



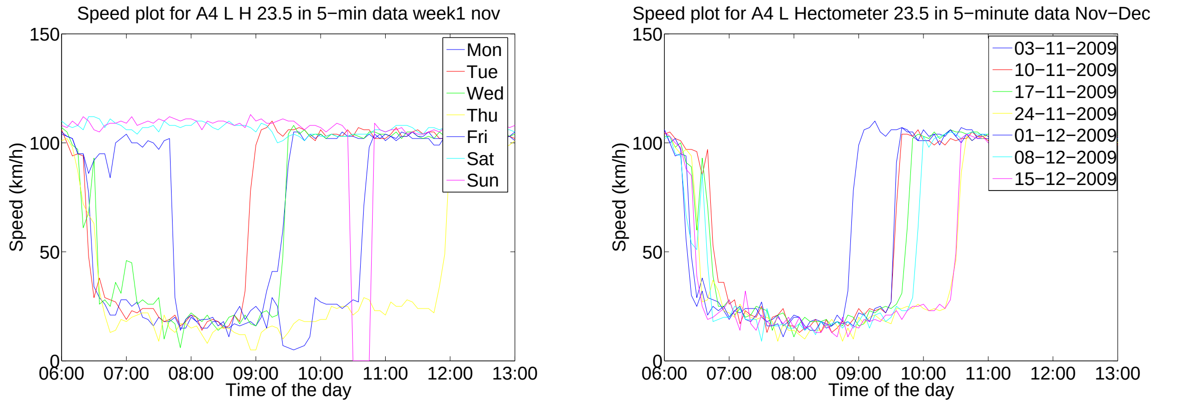
(a) Google Earth Bird's eye view

(b) Schematic overview A4 bottleneck

Figure 5.3: Situation overview A4

The question is: how predictable are congestions during rush hours and how 'sensible' is the information available for these situations? Figure 5.4(a) illustrates speed graphs for morning rush hours from Monday to Sunday of the first week in November 2009. Figure 5.4(b) represents speed graphs for morning rush hours for seven subsequent Tuesdays in November and December 2009. It can clearly be seen that there are different congestions for different days of the week. Saturday and Sunday tend to have no congestion at all, and there was a long lasting congestion on Thursday (yellow line). This extreme congestion on Thursday could be a specific pattern for a Thursday, or an outlier since there could have been an accident, road work, heavy rainfall, special offer at Ikea, or an unknown cause that day. If we look, for example, at congestions on Tuesdays, there is no specific pattern visible. Although Tuesdays tend to be similar, the congestions start or end at different times.

The next question is, how much different is a Tuesday morning rush hour congestion from a Wednesday, or a Monday from a Saturday, etc. A quantitative view on these questions would be profitable, since inspecting 365 speed graphs would be ineffective. Therefore, histograms are calculated for morning rush hours in the year 2009 for this specific location for each day of the week.



(a) Speed graph for weekdays in november 2009

(b) Speed graph for Tuesdays in november and december 2009

Figure 5.4: Speed plots A4 Leiden for different days

For comparison purposes, the average histogram computed over all days is also given. The x-axis of the histogram represents measured speeds and the y-axis represents the normalized probability that a certain speed is measured on that location during morning rush hour. Figure 5.5(a) shows these histograms measured on the A4 Leiden left side on hectometer 23.5, which is exactly the location where the 3 lanes are merged to 2 lanes.

The traffic on the location with hectometer 23.5 can of course influence the traffic conditions further upstreams during congestions. If a traffic queue grows in the upstream direction, traffic at hectometer location 25.95 or 28.31 could also become congested. Therefore, Figure 5.5(b) illustrates the histograms on hectometer location 25.95 and Figure 5.5(c) represents the histograms on hectometer location 28.31 to get an impression of this congestion propagation.

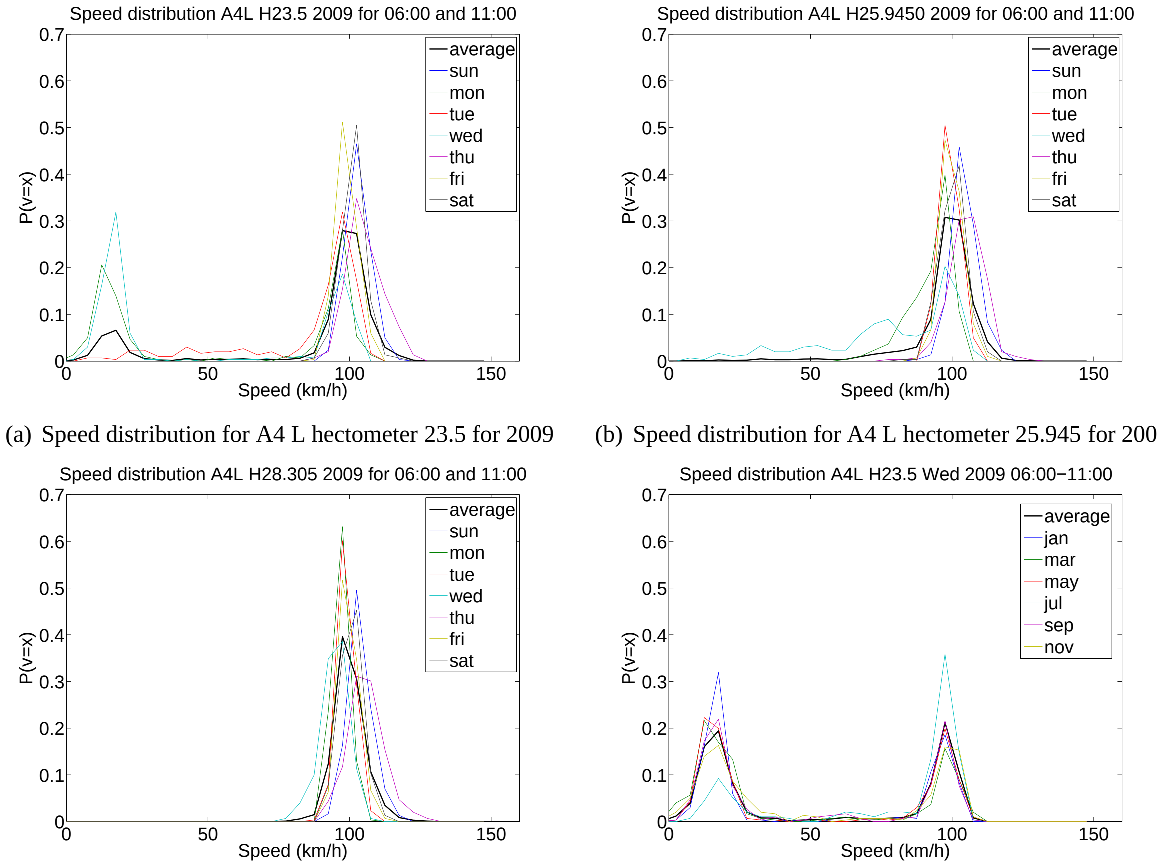
It can clearly be seen, if averaged of the whole year 2009, that at hectometer location 23.5 a Monday and Wednesday clearly have a bimodal distribution which indicates that there are more congestions during morning rush hours than the other days of the week. Tuesdays tend to have a wider distribution and Thursdays, Fridays, Saturdays and Sundays tend to have less congestions. But if we look at hectometer location 25.95, the probability for a congestion seems to decrease and at hectometer location 28.31 there seems to be hardly any congestion measured in 2009 on average.

But how exactly can we interpret the difference between these histograms? At this time it is necessary to introduce the term *Entropy* or *Shannon Entropy* as what it is usually referred to. The Shannon Entropy is a measure of uncertainty associated with a random variable such as speed in this case. It is actually a description of the amount of information that the signal contains. The Shannon entropy [32] (or self-information) H of a random variable X can be expressed as:

$$H(X) = \sum_{i=1}^{N} p(x_i) I(x_i) = -\sum_{i=1}^{N} p(x_i) log_b p(x_i), \tag{5.2}$$

where $b = 2$ bits in this thesis and $p$ denotes the probability mass function (histogram) of X.

The Shannon Entropy of a speed histogram can be calculated so that we have a measure of information content. If we can calculate entropies of speed histograms, how can we then compare



(a) Speed distribution for A4 L hectometer 23.5 for 2009

(b) Speed distribution for A4 L hectometer 25.945 for 2009

(c) Speed distribution for A4 L hectometer 28.305 for 2009

(d) Speed distribution for A4 L on Wednesdays in different months

Figure 5.5: Speed distributions

them? Therefore it is necessary to introduce the: *Kullback Leibler divergence*, or KL-divergence. The KL-divergence is a measure to calculate the difference between two probability distributions. The KL-divergence can be used to measure information gain during Bayesian updating when moving from a prior distribution to a posterior distribution. This is where we can use the differences between speed distributions of different days for prediction purposes. For example: if a new fact $Y = y$ is discovered, this fact can be used to update the probability distribution for $X$ from $p(x|I)$, where $I$ denotes the state of information, to a new posterior probability distribution $p(x|y, I)$ using the Bayes' theorem:

$$p(x|y, I) = \frac{(p(y|x)p(x|I))}{p(y|I)}. \tag{5.3}$$

This distribution has a new entropy:

$$H(p(x|y, I)) = \sum_x p(x|y, I) logp(x|y, I), \tag{5.4}$$

which may be less than or greater than the original entropy $H(p(x|I))$. The KL-divergence can then be calculated as follows:

$$D_{KL}(p(x|y,I)|p(x|I)) = \sum_x p(x|y,I) log \frac{p(x|y,I)}{p(x|I)}. \tag{5.5}$$

The KL-distance between a speed distribution for a certain day and an average day is then a value for information gain when adding more knowledge. The added knowledge in this case would be knowing which day it is. The KL-divergences for different locations on the road are placed in Table 5.5. It can easily be seen that knowledge about the day is most profitable in situations where congestions are present, for example at hectometer location 23.5. Further it can be seen that knowledge that it is a Thursday, Friday, Saturday or Sunday would give a gain in information seen over the year 2009 for that particular location, since there is a lower probability for congestions on these days. The histogram for a Wednesday clearly differs from the average histogram in that there is a much higher probability for a congestion. The information gain decreases for locations further situated from the bottleneck.

Table 5.5: Kullback Leibler divergence for different locations and days compared with an average over a 7-day week

| Hectometer | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
|---|---|---|---|---|---|---|---|
| 23.5 | 6.98 | 2.06 | 1.89 | 5.21 | 6.50 | 7.66 | 6.36 |
| 25.95 | 3.40 | 6.76 | 5.59 | 2.02 | 2.07 | 3.34 | 2.64 |
| 28.31 | 1.12 | 2.55 | 1.81 | 1.96 | 1.09 | 1.10 | 1.07 |

Traffic patterns on work days are likely to differ from traffic patterns during weekends. Therefore, it could be interesting to calculate KL-divergences between workdays and average workdays. Table 5.6 represents these KL-divergences for working days.

Table 5.6: Kullback Leibler divergence for different locations and days compared with an average over a 5-day working week

| Hectometer | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|
| 23.5 | 2.00 | 2.02 | 4.12 | 9.20 | 10.33 |
| 25.95 | 5.81 | 6.95 | 1.72 | 2.92 | 4.53 |
| 28.31 | 2.63 | 1.84 | 1.77 | 1.51 | 1.46 |

It can easily be seen that the KL-divergence on hectometer location 23.5 between speed distributions of a Friday or a Thursday in 2009 differs from speed distributions from an average working day. Speed distributions of Wednesdays also differ quite a bit compared to average working days, but knowing that its a Monday or a Tuesday does not seem to gain information for the traffic prediction problem. It should also be noticed that knowledge that it is a Monday or a Tuesday is profitable when measured at hectometer location 25.95 or 28.31. This indicates that differences between speed distributions of different days are dependable on their location.

Can we now claim that if we know the day, place and time that we can just take a historical average for a predicted traffic condition? I.e.: Does a certain day has the same characteristics seen over a year? By looking at the data, it is interesting to see whether the heavy congested days (for example Wednesdays) are similar to each other seen over a year. Figure 5.5(d) shows the speed distributions for Wednesdays in 2009 in the months: January, March, May, July, September and November. As can be seen in this figure, knowing that it is a Wednesday is not sufficient, since there are differences between the Months, and maybe also between weeks. If we divide our data more and more into subsets by months, weeks, etc, then our data becomes sparse en we end up with only a few days left to, for example, get an averaged value of speed.

The results in this Chapter indicate that it might be profitable to create subsets or clusters of traffic data which contain homogeneous data. This process can be called: *data streaming*, and is investigated in the next Chapter.

# Chapter 6

# Data Streaming

This Chapter investigates the effect of data streaming to find basic traffic patterns by an explorative analysis. The question is whether there are homogeneous subsets in the traffic data, so that maybe a rainy or a sunny day of the week has a characteristic traffic pattern. In Figure 6.1 we display the average vehicle *speed* $V(t, d)$ on a fixed location as a function of the time $t$ and for day $d$ of the year. As we can see, there is a variation in the start/end of the congestion and the depth of the valley. We expect the same situation for the variables *intensity* and *density*.

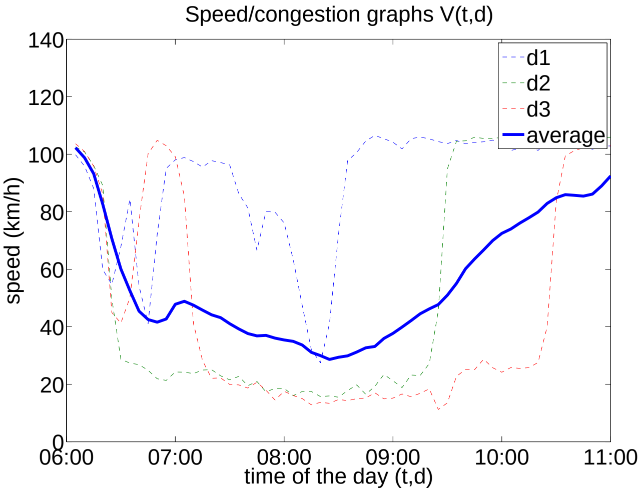


Figure 6.1: Vehicle speed graphs

We want to represent all days by the average speed graph $\bar{V}(t)$ for all days. Later, we want to use this average to compute the travel time. But if the vehicle speed shows huge variation, then the average is not a good representation for the data.

Our expectation is that the $\bar{V}(t)$ shows a great variance for all days of the year, but if we take a subset or cluster $S = (1 \leq d \leq M)$, then maybe the variation decreases and the dataset $S$ becomes more homogeneous.

Section 6.1 describes how these subsets could be seen as basic traffic patterns for the purpose of traffic prediction. The main question is if there are such patterns to be found. A dataset to analyze these patterns is prepared and described in section 6.2. The analysis of the variances of the clusters is described in section 6.3. In section 6.4 the clusters are inspected by face validity. After analyzing the data by inspecting the standard deviations and the homogeneity of the traffic patterns, a method is proposed in section 6.5 to automate the process of finding structure in the data. An experiment for this method is explained in detail in section 6.6. The results and interpretation of this experiment are described in section 6.7.

## 6.1 Using basic patterns for traffic prediction

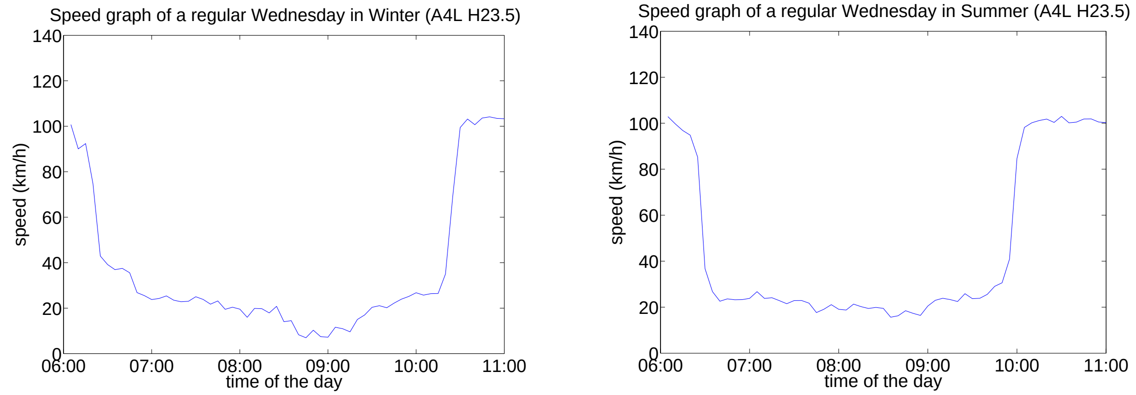
After having seen different congestion graphs in Chapter 5, we observed that there is a huge variance in the graphs. There are days which have no congestion, an extreme congestion, or everything in between. The question is: is it possible to find homogeneous subsets in the total collection of speed, intensity and density graphs? Every speed, intensity or density graph can be classified belonging to a certain day of the week, a certain time, a season, a type of weather, etc.

For example, lets say there is a congestion $C_1$ and we know that was is a day in winter. From another traffic congestion $C_2$ we know that it was a day in summer. The question is, how much different is congestion $C_1$ from congestion $C_2$? If these congestions are different, this could indicate that congestions during the winter are different from congestions during the summer at the same location. Generally speaking, if there are certain characteristics for which there are typical traffic or congestion patterns, this knowledge can be used for prediction purposes.

For clarity purposes, example congestion graphs for congestion $C_1$ and $C_2$ are presented in Figure 6.2, where $C_1$ is a randomly drawn morning rush hour of a day in winter and $C_2$ is a randomly drawn morning rush hour of a day in summer. An abrupt drop in speed is clearly visible in both graphs, so both days contain congested traffic. But do these graphs really differ enough so that it can be concluded that they have different patterns? The winter congestion seems to last a little longer and to start a little later then the summer congestion seen from these two congestions. This could be due to the fact that people are starting to work at a later time in the winter maybe because of the morning light. It could be a coincidence as well.

Do clusters (subsets) of congestion graphs in general differ enough, so that it might be useful to discriminate between possible congestions if we know, for example, what kind of day it is? The difference between these certain selection criteria needs to be quantified, in order to be able to use this information. If, for example, speed data is visualized for one year (365 days), it is expected that the variance in this dataset is large. By clustering the dataset into subsets for which certain characteristics hold in the corresponding subset, it is expected that the variance decreases and the congestion graphs will be become more homogeneous. In the end of this clustering process, the last split fraction should be able to yield a representative congestion graph, which could then be used as a general congestion model for that branch. For this, an experiment is conducted in this thesis. This experiment was conducted for the variables: *speed*, *intensity* and *density* in the year 2009. Possible subsets (clusters or split fractions) could be:

- weekend: yes - no
- day of the week: Mon - Tue - Wed - Thu - Fri - Sat - Sun



(a) $C_1$ : Congestion Graph for a day in Winter

(b) $C_2$ : Congestion Graph for a day in Summer

Figure 6.2: Two example congestion graphs

- holiday: yes - no
- public holiday: yes - no
- summer: yes - no
- rain: no - little - medium - heavy - extreme
- temperature: low, medium, high.

In this Chapter, an experiment is conducted to quantify the differences of these subgroups. It is investigated whether such homogeneous subgroups can be found in the data.

## 6.2 Data preparation

In this experiment, we choose to take the well known traffic bottleneck in the Netherlands: the A4 left side around hectometer 23.0. Around this location, the road capacity decreases from 3 to 2 lanes, and commuter traffic utilizes this road to travel between The Hague and Amsterdam during working days. Details of this location can be found in Chapter 5.

Typically, morning and evening rush hours can be detected on this road and it is decided to investigate subsets for the morning rush hour in 2009. The morning rush hour is defined between 06.00 am and 11.00 am and there are 365 days in 2009. The same location and time instants are used through the remaining of this Chapter.

A dataset was prepared for this experiment. This dataset consisted of speed, intensity and density values of morning rush hours in 2009. The data is conveniently enclosed in the structure as presented in Table 6.1. Each column contains data for a day $d$, where ($1 \leq d \leq M$) and each row contains data for a certain time $t$, where ($1 \leq t \leq N$), corresponding with that day. In this table, the $V$ can be replaced by a $I$ or a $D$ for the variables *intensity* or *density* respectively.

Table 6.1: Data Matrix for split fraction analysis

| time | Day 1 | Day 2 | ... | Day m |
|---|---|---|---|---|
| $t_1$ | $V_{11}$ | $V_{12}$ | ... | $V_{1M}$ |
| $t_2$ | $V_{21}$ | $V_{22}$ | ... | $V_{2M}$ |
| ... | ... | ... | ... | ... |
| $t_N$ | $V_{N1}$ | $V_{N2}$ | ... | $V_{NM}$ |

Since this particular ILD at hectometer 23.5 contains missing data, it was decided to filter the data with the Treiber and Helbing filter to fill in the missing values. Details of this Treiber and Helbing filter can be found in Chapter 4.

## 6.3 Data analysis

In this section, we compute the homogeneity of traffic data set $S$. Let us assume that $V(t, d)$ is the average vehicle speed as function of the time $t = t_i$, where $(i \leq t \leq N)$, and for a specific day $d$, where $(1 \leq d \leq M)$. We may assume that the $V(t_i, d)$ on a specific time for day $d$ has a normal distribution, as is shown by the distribution graphs in Chapter 5.

The standard deviation $\sigma$ is a measure to express the homogeneity of a dataset. Generally, the standard deviation $\sigma$ for a variable $X$ is:

$$\sigma = \sqrt{E[(X - \mu)^2}. \tag{6.1}$$

In our case, the average vehicle speed graph of the dataset$\bar{V}(t = t_i)$ is

$$\bar{V}(t = t_i) = \frac{1}{M} \sum_{d=1}^{M} V(t_i, d), \tag{6.2}$$

where $M$ denotes the number of days in set $S$. To calculate the standard deviation $\sigma_i$ at time $t_i$, we compute

$$\sigma_{t_i} = \sqrt{\frac{1}{M-1} \sum_{d=1}^{M} (V(t_i, d) - \bar{V}(t_i))^2}. \tag{6.3}$$

Since the data is sampled from the total data set we use $M - 1$.

The average standard deviation $\sigma_{total}$ averaged over $i$ becomes:

$$\sigma_{total} = \frac{1}{N} \sum_{i=1}^{N} (\sigma_{t_i}), \tag{6.4}$$

where $N$ denotes the number of time instants $t_i$.

The starting point is a dataset with 365 morning rush hours in 2009. We take this total set as the first level in clustering. Table 6.2 presents the standard deviations $\sigma$ for different subsets at the first level, in which $\sigma_V$, $\sigma_I$, $\sigma_D$ denotes the average standard deviation for the cluster for the variable

Table 6.2: Standard deviation for cluster level 1

| Cluster | $\sigma_V$ | $\sigma_I$ | $\sigma_D$ | #samples |
|---|---:|---:|---:|---:|
| **Total set** | **35.17** | **14.82** | **52.36** | **365** |
| Weekend days | 6.14 | 5.46 | 5.93 | 261 |
| Week days | 32.97 | 9.90 | 48.43 | 104 |
| Mondays | 28.34 | 9.02 | 42.04 | 52 |
| Tuesdays | 31.23 | 7.61 | 46.66 | 52 |
| Wednesdays | 30.75 | 8.74 | 45.67 | 52 |
| Thursdays | 34.37 | 11.82 | 51.50 | 53 |
| Fridays | 29.28 | 10.84 | 38.41 | 52 |
| Saturdays | 4.68 | 4.64 | 5.87 | 52 |
| Sundays | 6.27 | 3.81 | 3.44 | 52 |
| Holiday | 28.59 | 16.03 | 40.82 | 97 |
| No holiday | 35.83 | 14.16 | 53.66 | 268 |
| Public holiday | 21.62 | 13.47 | 24.46 | 19 |
| No public holiday | 35.14 | 14.25 | 52.26 | 346 |
| Summer | 34.79 | 14.53 | 51.44 | 211 |
| Winter | 35.42 | 15.03 | 53.40 | 154 |
| No rain | 34.80 | 15.13 | 51.71 | 149 |
| Little rain | 35.36 | 14.59 | 52.67 | 215 |
| Medium rain | 38.02 | 14.22 | 55.82 | 14 |
| Heavy rain | 0 | 0 | 0 | 1 |
| Extreme rain | NaN | NaN | NaN | 0 |

*speed*, *intensity* and *density* respectively. This table shows that the total set (cluster) contains a huge standard deviation. The standard deviation is around 35 km/h for the variable speed. This means that the average, $\bar{V}(t,d)$, deviates 35 km/h on average from a certain day $d$ in set $S$, if the average speed graph is taken as a representation for every day $d$ in set $S$. There is a high standard deviation in the intensity and density as well for the total set.

If we select a cluster $S$ in which only the weekend days are included, the variance in speed decreases enormously. If the average speed graph for this cluster is taken as a representative for this set $S$, the days $d$ in $S$ on average deviate only around 6 km/h. On the other hand, the standard deviation for the cluster consisting of week days is still high. Therefore, we are not interested in finding more subgroups for the weekends at this point, but we are interested in finding homogeneous subgroups for week days.

As can be seen in Table 6.2, the standard deviation for the days of the week cluster is lower then for the total set. Clustering the data for rain or season seem to have little effect on the standard deviation. The cluster of holidays or public holidays seem to be more homogeneous as well. This does not come as a surprise, since the commuter traffic is not likely to travel on these days. Traffic during congestion consists of 64% of commuter traffic in the Netherlands [30].

Another interesting result is that the cluster for Mondays, Tuesdays en Wednesdays have comparable standard deviations for speed, whereas Thursdays have a higher standard deviation. Further, the results show that the standard deviation for density for Fridays is lower then the standard deviation for the other working days of the week. This indicates that there could be clusters in the week days cluster for certain combinations of week days.

Now, we are interested in a cluster of the next level. Let us take the cluster of Mondays, and cluster this set again for holidays, season and rain. The results are presented in Table 6.3. At a first glance, it becomes clear that the data becomes more sparse. This means that the results become less reliable and unstable since there are less samples. Especially for the higher level of rain clusters, there is not enough data. It is interesting to see that the standard deviation for the little rain cluster is lower then for the total cluster of Mondays, although this is computed over 27 days.

Table 6.3: Standard deviation for cluster level 2 (Mondays)

| Cluster | $\sigma_V$ | $\sigma_I$ | $\sigma_D$ | #samples |
|---|---:|---:|---:|---:|
| **Mondays** | **28.34** | **9.02** | **42.04** | **52** |
| Mondays - holiday | 27.23 | 9.00 | 36.62 | 13 |
| Mondays - no holiday | 27.01 | 8.85 | 40.89 | 39 |
| Mondays - public holiday | 1.15 | 2.62 | 1.43 | 2 |
| Mondays - no public holiday | 27.17 | 7.29 | 40.18 | 50 |
| Mondays - summer | 29.27 | 10.01 | 42.96 | 30 |
| Mondays - winter | 25.51 | 7.28 | 39.19 | 22 |
| Mondays - no rain | 29.86 | 9.79 | 43.96 | 25 |
| Mondays - little rain | 25.83 | 8.27 | 39.03 | 27 |
| Mondays - medium rain | NaN | NaN | NaN | 0 |
| Mondays - heavy rain | NaN | NaN | NaN | 0 |
| Mondays - extreme rain | NaN | NaN | NaN | 0 |

Further, the table shows that the holiday clusters have a different standard deviation then the total Mondays cluster. Which seems logical, since the commuter traffic is likely to travel less during holidays. We remind the reader that 64% of the traffic during congestions is commuter traffic in the Netherlands [30]. In the end, the season does seem to have influence for the homogeneity of the cluster, but we should keep in mind that the data becomes sparse.

For comparison purposes, the same clustering for the second level is done for the Tuesdays cluster. The results are places in Table 6.4. Here, the results show a difference between the holiday clusters for Tuesdays as well. For the rest, the data becomes sparse here as well and this makes it hard to draw any strong conclusions. Therefore, it is necessary to do a face validity on the clusters and see whether the data indeed becomes more homogeneous for different subsets.

## 6.4 Face validity of homogeneous subgroups

After having seen these tables of standard deviations, the main question still remains: Do these congestion graphs become more homogeneous after an appropriate number of clustering levels? This question is investigated by visual inspection of a couple of congestion subgroups. Figure 6.3 presents the congestion graphs on speed, intensity and density for the total set, and for Tuesdays. Figure 6.4

Table 6.4: Standard deviation for cluster level 2 (Tuesdays)

| Cluster | $\sigma_V$ | $\sigma_I$ | $\sigma_D$ | #samples |
|---|---|---|---|---|
| **Tuesdays** | **31.23** | **7.61** | **46.66** | **52** |
| Tuesdays - holiday | 32.14 | 7.67 | 45.64 | 13 |
| Tuesdays - no holiday | 28.11 | 7.05 | 42.85 | 39 |
| Tuesdays - public holiday | 0 | 0 | 0 | 1 |
| Tuesdays - no public holiday | 30.78 | 7.63 | 46.22 | 51 |
| Tuesdays - summer | 32.24 | 7.51 | 48.01 | 30 |
| Tuesdays - winter | 29.50 | 7.08 | 44.70 | 22 |
| Tuesdays - no rain | 33.13 | 8.46 | 49.80 | 20 |
| Tuesdays - little rain | 30.03 | 6.66 | 44.60 | 32 |
| Tuesdays - medium rain | 28.70 | 4.90 | 49.17 | 4 |
| Tuesdays - heavy rain | 0 | 0 | 0 | 1 |
| Tuesdays - extreme reain | NaN | NaN | NaN | 0 |

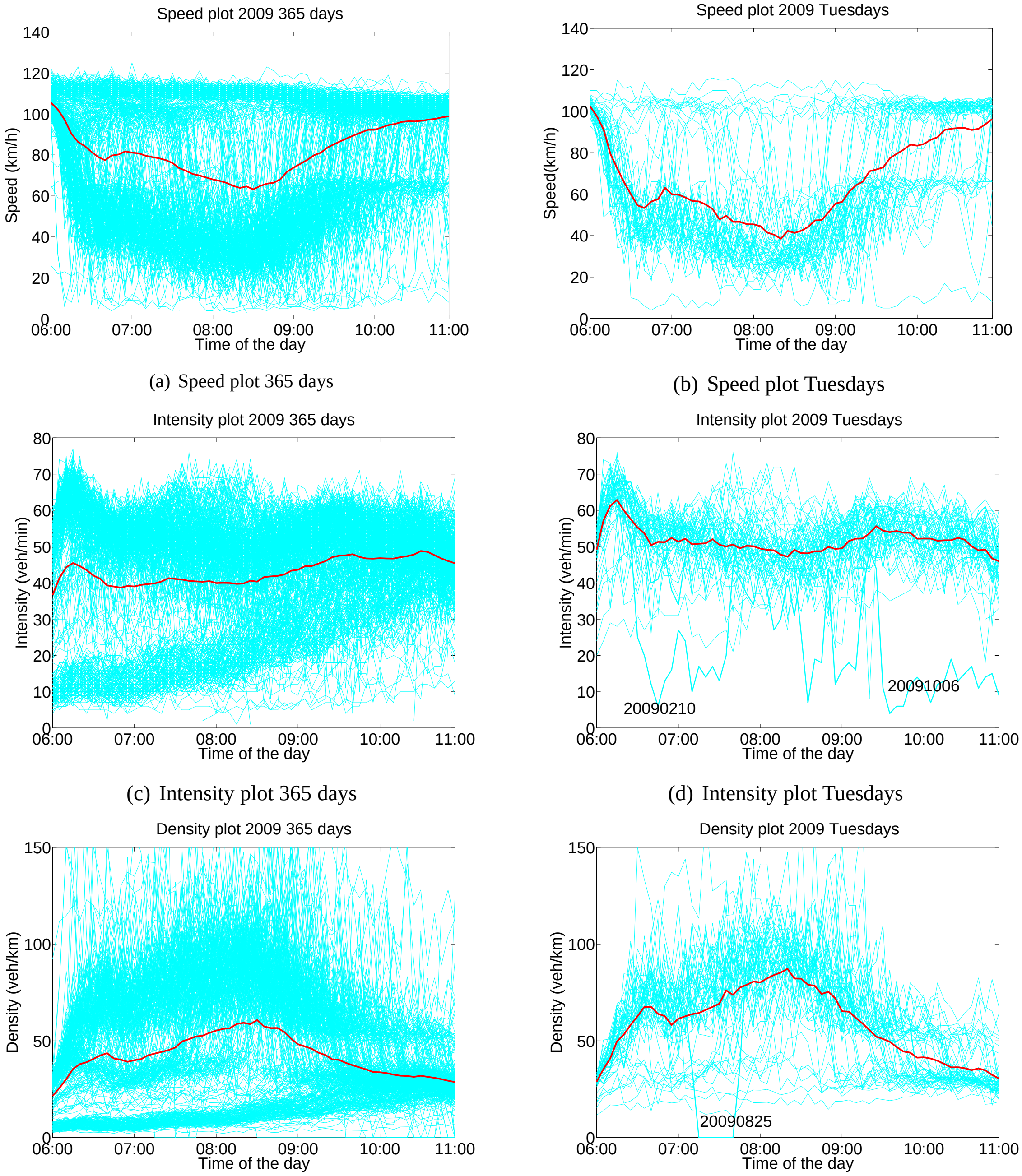
goes even a few steps further and represents congestion graphs on speed, intensity and density for Tuesdays on which there is no holiday, it is a winter day, and there is a little rain.

It becomes clear that in the Tuesdays cluster, the data becomes more homogeneous compared to the cluster of the total dataset. In Figure 6.4 it is less visible that the data becomes more homogeneous. After a few cluster levels further, the data becomes more sparse and therefore unstable. Note that the outliers are annotated with their date in the figures. It is difficult to compare the different means in these figures, therefore an interesting selection of means is plotted in Figures 6.5 and 6.6 to compare the means at a somewhat higher level. These plots can be seen as typical *day patterns*. From Figure 6.5 it becomes clear that there are typically 3 clusters in the data. Cluster 1 consists of weekend days, cluster 2 of Mondays, Tuesdays, Wednesdays and Thursdays, and cluster 3 consists of Fridays. These groups can be seen as clusters, since the mean plots of the days within the clusters have a low distance to each other and are therefore similar.

Another interesting and more traffic theory related result is a change in distance headway. It can be seen that the mean speed on Fridays is higher then from Monday to Thursday. It can also be seen that the densities on Fridays are lower then from Monday to Thursday. The intensities from Monday to Friday are more or less in the same range. This implies that the distance headway is different on Fridays. Since the drivers tend to drive faster on Fridays, it is likely that their distance headway is higher then on Monday, Tuesday, Wednesday or Thursday. Details of the fundamental traffic theory and formulas behind this can be found in Chapter 2.

From Figure 6.6 it becomes clear that the graphs become unstable. After clustering the data for several levels, the data becomes more and more sparse. Therefore it becomes more difficult to see specific traffic patterns.

It should be clear that the results of this experiment are only based on the morning rush hours in 2009 at the A4 left side Leiden at hectometer location 23.5. This experiment should be conducted at different locations, for example, at different bottle necks. It would also be interesting to inspect how the variances of the congestion graphs behave during evening rush hours, or maybe even in weekend rush hours. Section 6.5 describes how this process could be conducted automatically.



(a) Speed plot 365 days

(b) Speed plot Tuesdays

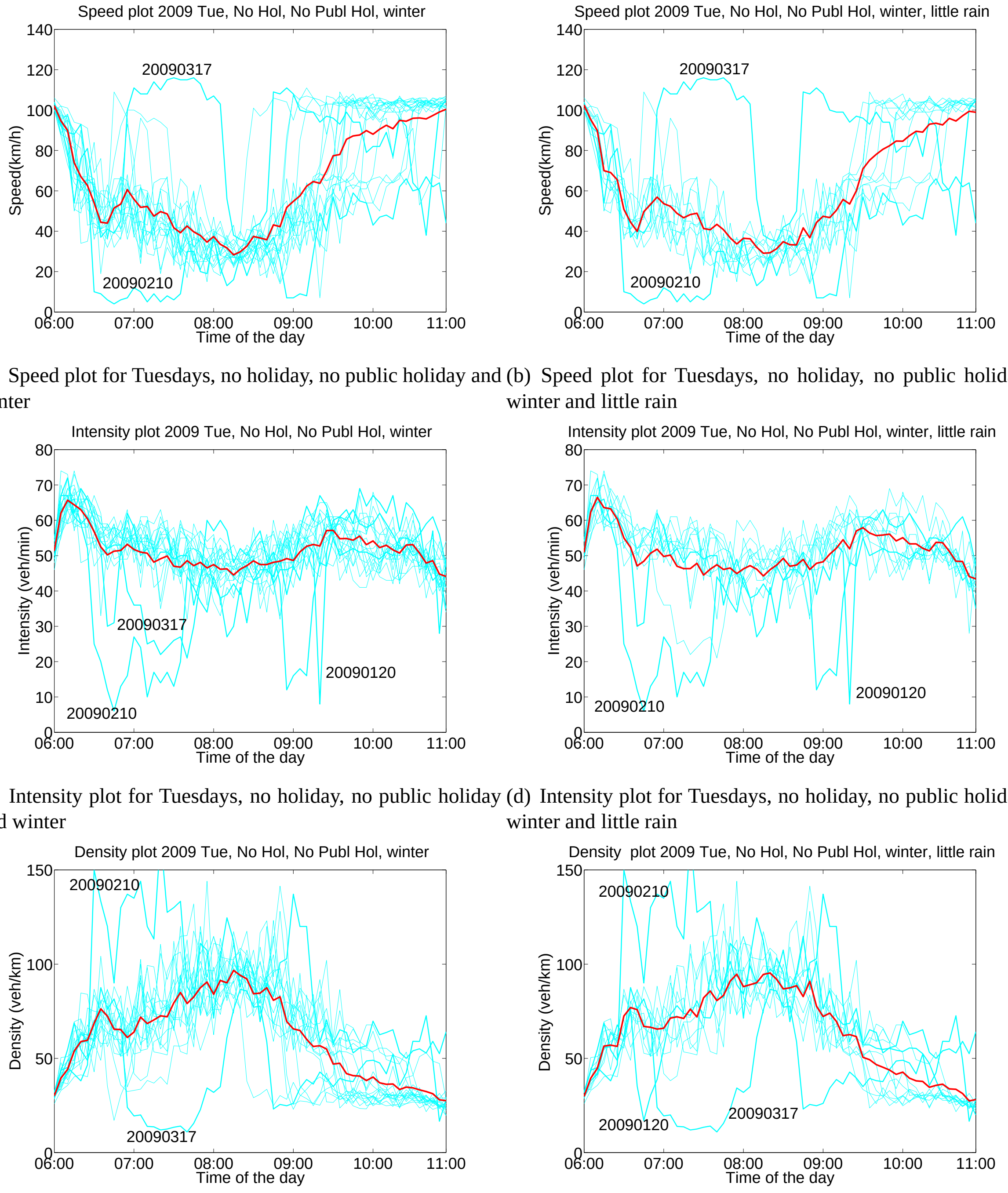
(c) Intensity plot 365 days

(d) Intensity plot Tuesdays

(e) Density plot 365 days

(f) Density plot Tuesdays

Figure 6.3: First level split fraction from total set to Tuesdays for speed, intensity and density



(a) Speed plot for Tuesdays, no holiday, no public holiday and winter

(b) Speed plot for Tuesdays, no holiday, no public holiday, winter and little rain

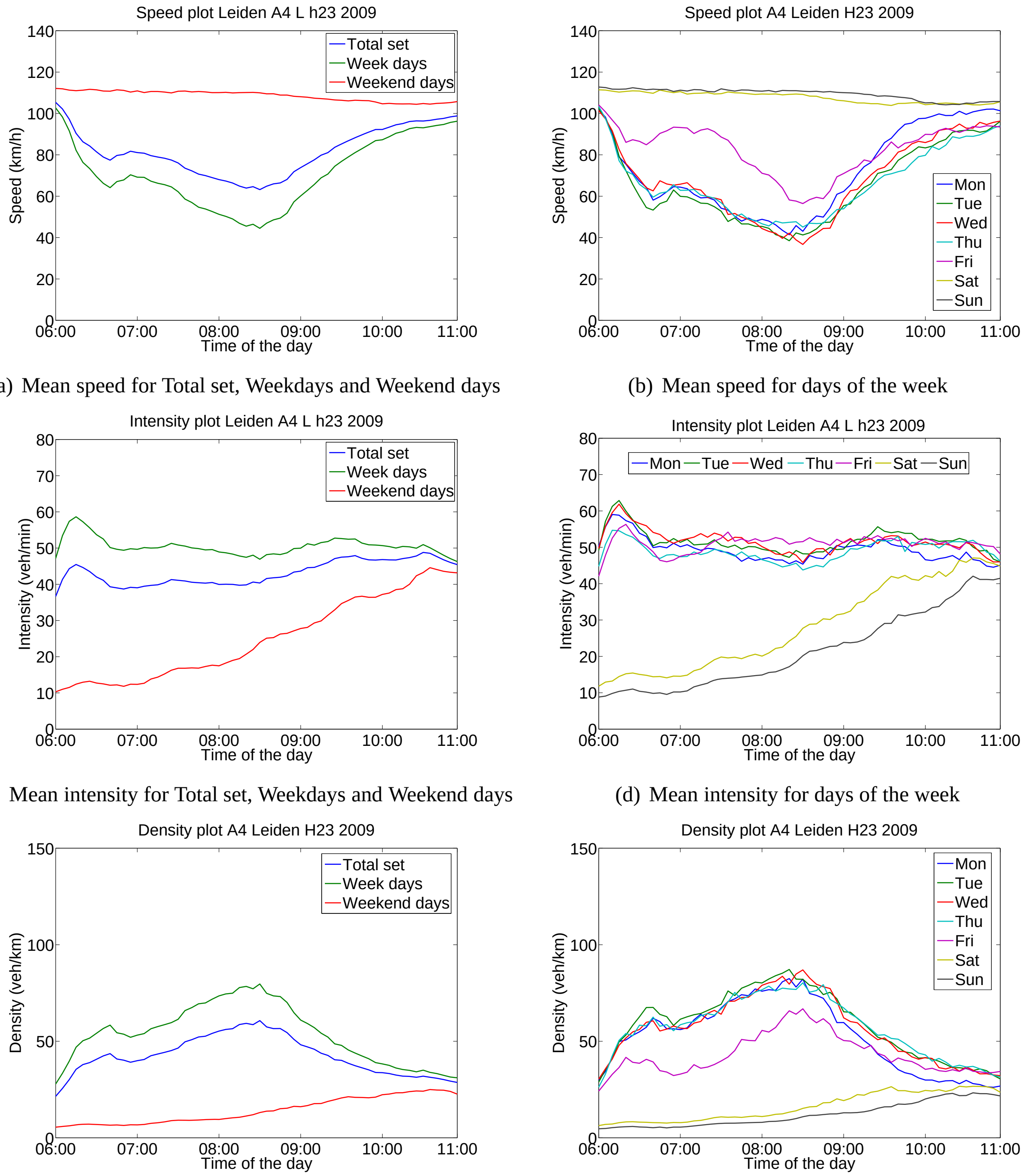
(c) Intensity plot for Tuesdays, no holiday, no public holiday and winter

(d) Intensity plot for Tuesdays, no holiday, no public holiday, winter and little rain

(e) Density plot for Tuesdays, no holiday, no public holiday and winter

(f) Density plot for Tuesdays, no holiday, no public holiday, winter and little rain

Figure 6.4: Higher level split fraction with little rain for speed, intensity and density



(a) Mean speed for Total set, Weekdays and Weekend days

(b) Mean speed for days of the week

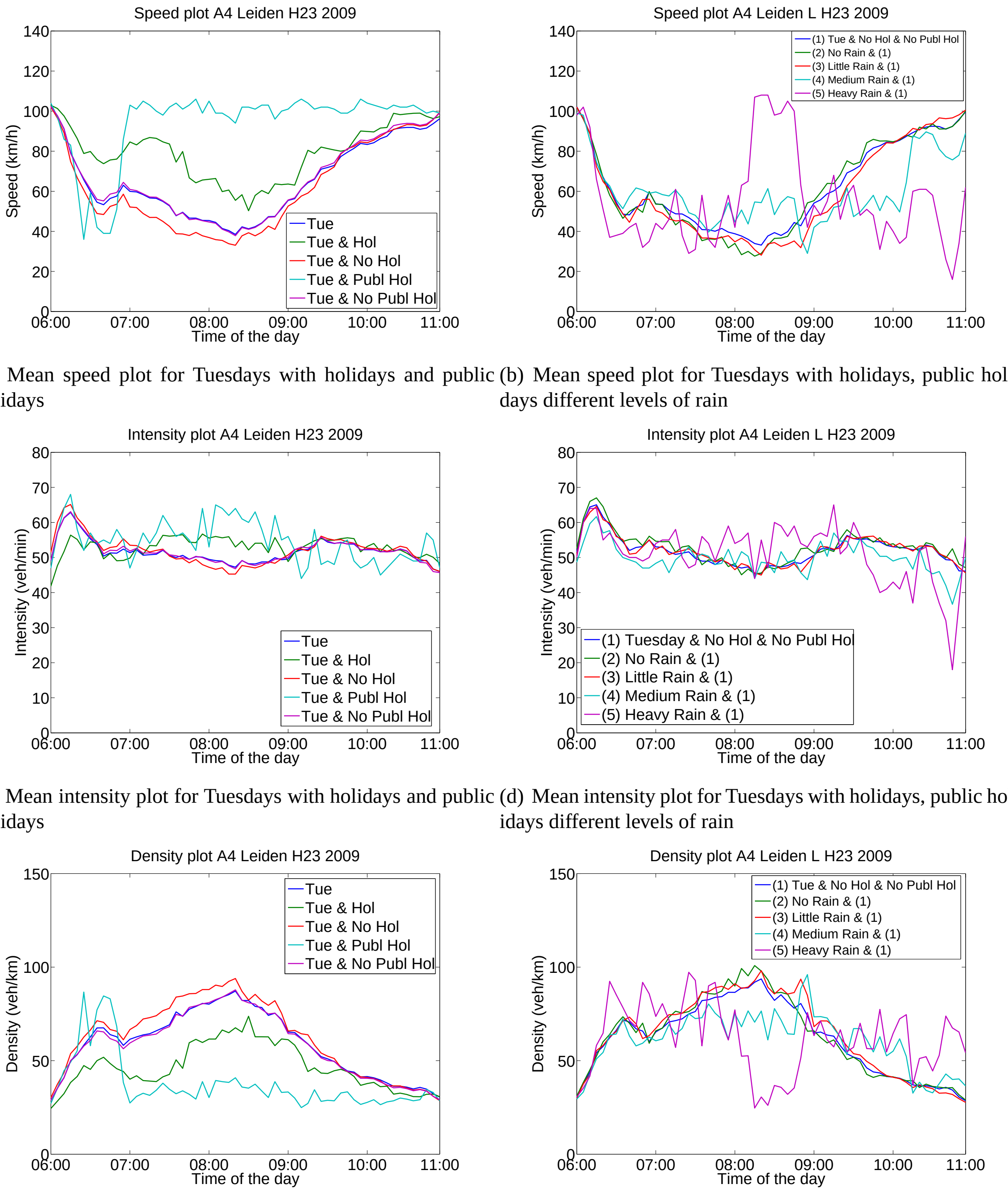
(c) Mean intensity for Total set, Weekdays and Weekend days

(d) Mean intensity for days of the week

(e) Mean density for Total set, Weekdays and Weekend days

(f) Mean density for days of the week

Figure 6.5: Mean plots for speed, intensity and density for the total set, weekdays, weekend days and days of the week



(a) Mean speed plot for Tuesdays with holidays and public holidays

(b) Mean speed plot for Tuesdays with holidays, public holidays different levels of rain

(c) Mean intensity plot for Tuesdays with holidays and public holidays

(d) Mean intensity plot for Tuesdays with holidays, public holidays different levels of rain

(e) Mean density plot for Tuesdays with holidays and public holidays2

(f) Mean density plot for Tuesdays with holidays, public holidays different levels of rain

Figure 6.6: Mean plots for speed, intensity and density on higher level split fractions

## 6.5 Data streaming by decision trees

Since the process of inspecting standard deviations and homogeneity of the data is a tedious and labor-intensive job as we have seen in section 6.3, there is a need for an algorithm that is capable of choosing and generating these clusters, or split fractions, automatically. A subsequent set of split fractions situated in a graphical tree-like structure, can be called a *decision tree*. After each split fraction in the decision tree, the remaining data should become more homogeneous. This can be illustrated with an example which is based on the well known decision tree example as proposed by Quinlan [27]. Let's say we have a dataset $D$ as in Table 6.5 which contains weather information for $M$ days. Each day has the characteristics: ID, outlook, temperature, humidity and windy. The days are labeled as P (positive) or N (negative), which indicates whether these days where positive or negative instances for some kind of activity, for example: sailing. By investigating certain combinations in this dataset, it is possible to form a decision tree. Lets say, for example, that 90% of the days which are sunny and have normal humidity, are also windy so a positive instance for sailing. Then it might be enough to know that it is sunny with normal humidity, to say that it is a good day for sailing, since it probably is also windy. This knowledge can be learned form the data.

Table 6.5: Example Data Matrix: forecast for sailing

| ID | Outlook | Temperature | Humidity | Windy | Class |
|---|---|---|---|---|---|
| 1 | rain | hot | high | false | N |
| 2 | sunny | mild | normal | true | P |
| ... | ... | ... | ... | ... | ... |
| M | rain | cool | normal | true | N |

If such knowledge is learned from the data, a new day could be classified as a positive or negative instance if a similar day in the data is already known. More general, if there are $M$ days in this data set and a new day $(M + 1)$ is given without its class description, it could be classified according to the information in this dataset. Lets say that 80% of the days in the dataset which are classified as positive are sunny, have a mild temperature with normal humidity and are windy. If day $M + 1$ is also sunny, has a mild temperature with normal humidity and is also windy, then it is highly likely that this day should also be a positive instance for sailing. In this way, new data can be classified according to the data.

If the data is conveniently structured in a tree, then all we have to do is just follow the nodes and leaves to find our label to make our decision: positive or negative. A possible decision tree, based on Quinlan's example [27], is given in Figure 6.7. For sailing activity, there is a need for wind. Apparently, many days have been classified as positive instances for sailing for sunny days with normal humidity, since this is a decision path in the decision tree in Figure 6.7. A decision path is a subsequent number of split fractions which lead to a conclusion for a class label, i.e. the leaf of the tree. For making conclusions correctly, it is essential to find the best possible split fractions. This can be done by investigating the information gain after each split fraction. In fact, we are looking for a split fraction which makes the data more homogeneous.

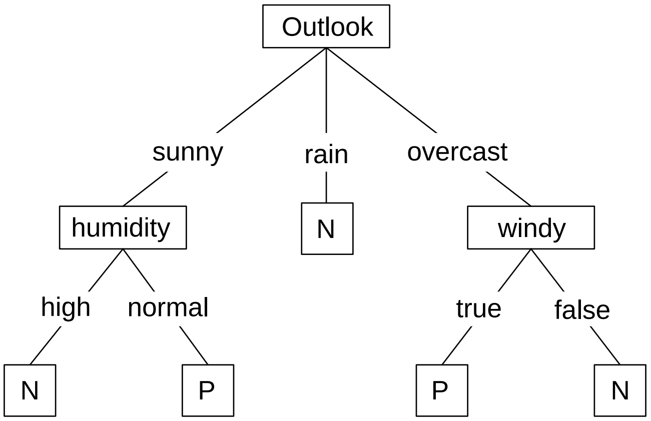


Figure 6.7: Example decision tree: forecast for sailing

This simple decision tree example illustrates an important principle: *extracting knowledge from data*. This idea can easily be translated into a traffic decision tree. The process of creating these trees automatically is called *decision tree learning* and is an active field in research areas such as data mining and machine learning. Usually, a decision tree is used as a predictive model which maps observations to conclusions. In literature, they are also called *classification trees* or *regression trees* [2].

In general, decision tree algorithms chose the best variable for a split fraction in each step. The best variable in this sense is the variable that splits the data set into parts which are as homogeneous as possible. A well known technique is Quinlan's C4.5 algorithm, which uses the *information gain*, based on the concept of entropy used in information theory [28]. An improved version of this algorithm can be found in [29]. This algorithm is based on the minimum description length (MDL) [26]. Quinlan states that the minimum description length states that the best theory to infer from a dataset is the one which minimizes the sum of (1) the length of the theory and, (2) the length of the data when encoded using the theory as a predictor for the data [26]. Both lengths are measured in bits and can be expressed as the Shannon Entropy with the log to the second base, as explained earlier in Chapter 5. The main idea of this algorithm, is that the best possible split fraction is the one with the highest Kullback-Leibler divergence. Details of this Kullback-Leibler divergence can be found in Chapter 5.

Compared to other data mining methods, decision trees have a few advantages [2]:

- they are easy to interpret and could therefore be seen as *white box* models as opposed to the well known *black box* models such as neural networks
- they are able to handle both numerical and categorical data
- they are robust
- they perform well with large data in a short time.

There are also some disadvantages. The problem of learning optimal decision trees is NP-complete [2]. Further, decision trees could become over-complex and they are not appropriate for every problem. Over-complexity can often be solved by pruning the tree. Details of this process can be found in [28].

There are numerous software packages available for decision tree learning. Given the size of the dataset, it was decided to use Rapid Miner from the Rapid-I Company. This software application has possibilities to parallelize processes and is able to handle large datasets.

A decision tree which represents the dataset adequately, is also able to predict new situations. Since this experiment only takes data of 2009 into account, it is not possible to create a general decision tree which generalizes all traffic situations. In fact, it is never possible to generate such a tree. Therefore, there is a need for an approximated decision tree which is general enough to classify new traffic situations.

## 6.6 Experimental setup

For this experiment, a dataset has been created which has the same format as presented in Table 6.6. The variable $V$ denotes speed which is measured at different time instants from $i, ..., N$. Each speed measurement represents a 5-minute average speed. The speed is measured from 06.00 am until 11.00 am at the A4 left side around Leiden at hectometer 23.5 and the dataset consist of 365 days in 2009. This is the same location as presented in Chapter 7.

Table 6.6: Data Matrix for decision tree learning

| ID Date | Workday | Day | Holiday | Public Holiday | Summer | Rain | $v_i$ | ... | $v_N$ |
|---|---|---|---|---|---|---|---|---|---|
| 20090101 | YES | THU | YES | YES | NO | No Rain | 119 | ... | 89 |
| 20090102 | YES | FRI | YES | NO | NO | Little Rain | 112 | ... | 56 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 20091231 | YES | THU | NO | YES | NO | No Rain | 111 | ... | 114 |

It is not convenient to represent every speed measurement as a variable, since it will result in a large number of variables which increases the sparsity of the data. Further, speed measurements ordered over subsequent time instants can be seen as a time series. Therefore, it was decided to model the speed measurements as a time series by making use of the windowing approach. A window function is a mathematical function that is zero-valued outside of some chosen interval. It was chosen to take a rectangular window with a size of 6 measurements. Each measurement is a 5-minute average, so the window size is 30 minutes. The prediction horizon was chosen to be 2 time steps, which represents 10 minutes. The window is shifted along the time axes with steps of 5 minutes. Each window has a label which consists of the speed measurement at 10 minutes later then the last time instant of the window. This process generates a dataset with speed windows and labels. The labels are the actual predictions which belong to the pre-occurring speed measurements in the window. In this way, the dataset is ready for classification purposes and decision trees can be learned.

There are numerous options available for different parameters which, in effect, all yield slightly different trees. It was chosen to use *Quinlan's C4.5* algorithm to learn the decision tree. Details of this algorithm can be found in [28], [27] and [29]. The criterion was set to *information gain*. The minimal size for a split fraction is set to 4 data points and the minimum leaf size is set to 2 data points. The minimal information gain is set to 0.1 and the maximum tree depth is set to 20. Speed measurements are discretized into 3 and 5 classes, which resulted in 2 different decision trees for comparison purposes. Both trees have exactly the same settings, except for the number of discretization classes.

## 6.7 Results and interpretation

The resulting decision trees are presented in Figure 6.8 an 6.9.In the first figure, *range1*, *range2* and *range3* represent low, medium and high speed respectively as the speed measurement are discretized into 3 bins. The latter figure models speed values in 5 classes and therefore *range1 - range5* represent the speed from low to high respectively. Variables $att\_1-i$ represents the speed value from the history at time $t-i$. Recall that one time step is 5 minutes, so $i$ represents a number of these steps.

It should be noted that the decision tree is optimized by cross validation where randomly drawn parts of the dataset are input for the decision learning process. This iterative process delivers the best decision tree in the end. The best decision tree, is the tree which classifies new data, which is excluded from the training process, the best.

From Figure 6.8 it becomes clear that the speed value at $t_0$ is the most important factor for predicting the average speed for over 10 minutes. If the speed is *medium* or *slow* at $t_0$, it is most likely that it will still be *slow* or *medium* after 10 minutes. This should not come as a surprise, since the traffic is probably congested. However, this decision tree is not able to predict a sudden increase in speed if the current speed is *medium* or *slow*. This means, that this tree is not able to predict the end of a congestion.

If the current speed is *fast*, then it depends on the speed at $t_{-4}$ whether the traffic will increase or decrease its speed. If the speed at $t_{-4}$ is *fast* as well, it depends on the day whether it is likely that the traffic will still drive fast after 10 minutes. As can be expected, Saturdays and Sundays are likely to have fast traffic, and surprisingly Fridays as well. As learned from section 6.3 and section 6.4, Fridays typically have a different day pattern and could therefore be seen as a separate cluster. This decision tree strengthens this claim. It can also be seen that a Tuesday or a Wednesday is also likely to have fast traffic after 10 minutes, but the divided color bar in the graph at these days shows that the data is not as homogeneous, as it is for Fridays.

This decision tree goes even further and creates split fractions for factors as *rain*, *holidays* and then tries to cluster the data even more based on speed values. The further a decision path gets to the leaves, the sparser the data becomes and the harder it is to make claims. But it is interesting to see that if its a Monday and the traffic is driving fast, it depends on the rain whether the traffic will keep on driving fast.

The decision tree in Figure 6.9 has a similar structure as the tree in Figure 6.8. If the current traffic is driving *fast*, it depends on the traffic a few minutes back whether the traffic will drive at a lower speed or not after 10 minutes. If the speed alternates between *fast*, *low* and *fast* in the last few minutes and then switches to *medium*, it depends on the fact if it is a holiday whether the traffic will increase or decrease its speed. At this stage, it is visible that the speed is in *range4* if there is a holiday, which could be due to the fact that there is less commuter traffic and therefore no congestion. Recall that 64% of congested traffic is commuter traffic in the Netherlands [30]. Fridays are again treated differently from other days, and, in general, traffic is driving faster if it is holiday according to this tree. There are more split fractions visible in this tree, but it should be kept in mind that the data becomes more sparse when divided into more discretization classes. This might be the cause that this decision tree is more complicated than the previous. Therefore, conclusions are hard to draw and this has to be done with caution.

In general, it was shown that it is possible to learn decision trees automatically from traffic data in this way. The structure does not come as a surprise, and has certain similarities with previously revelations. This process of decision tree learning should be done at different locations in the road network to get a feeling of the most important factors which are essential for traffic predictions.

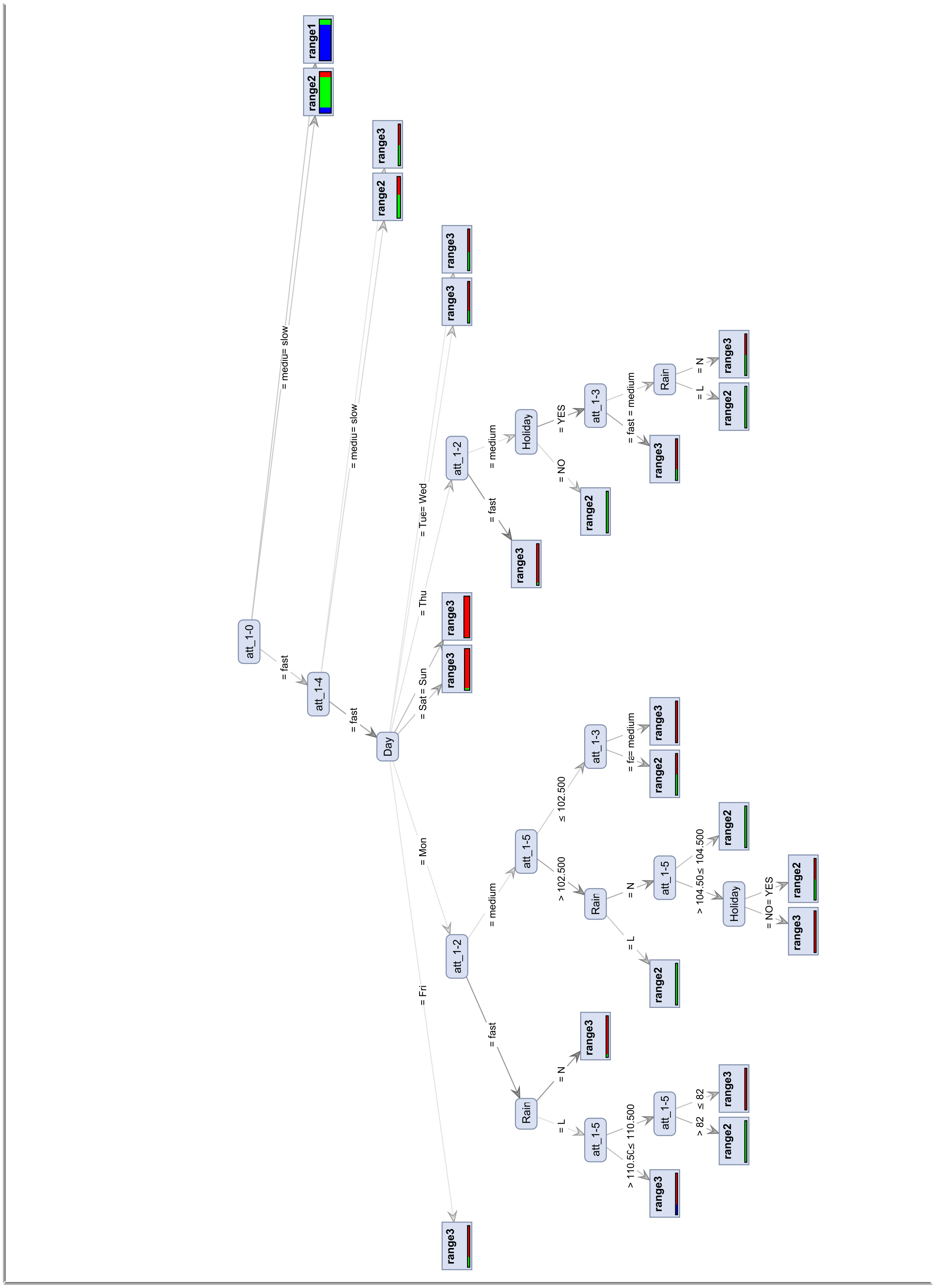


Figure 6.8: Decision tree with 3 discretized classes for speed

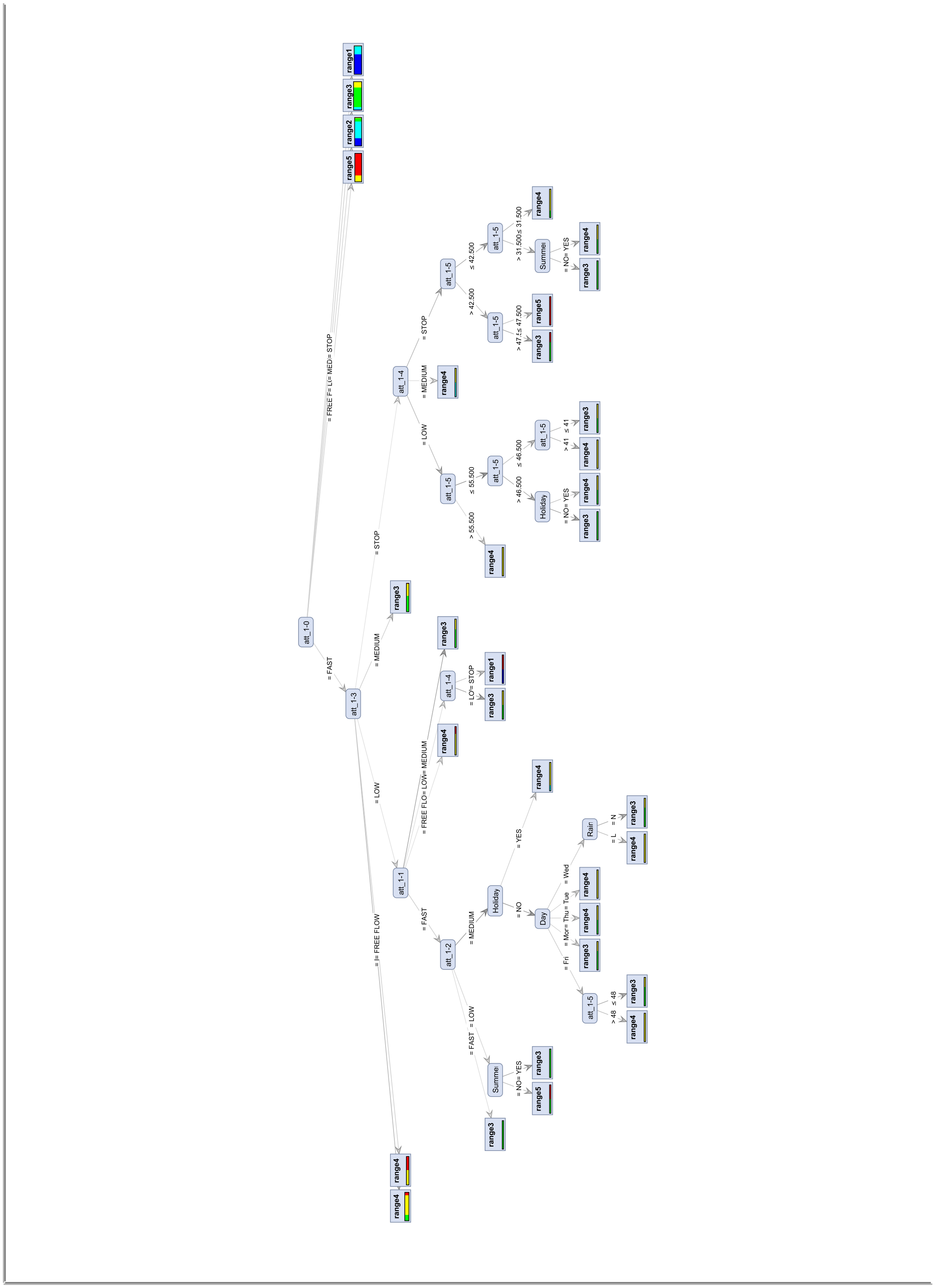


Figure 6.9: Decision tree with 5 discretized classes for speed

# Part III

# Model and Implementation Part

# Chapter 7

# Bayesian Traffic Modeling

In this Chapter, a case study is conducted to explore Bayesian Networks for traffic prediction purposes. A Bayesian Network consists of structured nodes as is explained in Chapter 3. For the experiment in this Chapter, the nodes represent traffic measurements for locations close to a location of interest. This location of interest is called the *prediction node*, which can be seen as the effect node. The average vehicle speed is predicted for this prediction node over different time horizons.

We expect that density measurements in combination with speed measurements are indicators for traffic congestions. These measurements could be taken from different locations and modeled in a Bayesian Network. We must choose the locations on the road which are modeled in the Bayesian Network wisely, since these locations should be able to indicate traffic congestions. From Chapter 4 we know that traffic information propagates against the driving direction in case of congested traffic. Therefore, we expect that modeling downstream locations on the road will yield the best results. This hypothesis is tested in this Chapter.

Section 7.1 describes how knowledge of traffic can be incorporated in Bayesian Models. This section also introduces the traffic situation where the experiments in this Chapter are based on. Section 7.2 describes the Bayesian Models which we propose. These models incorporate different locations on the road. The models need to be tested and evaluated in an experiment, from which the setup is described in section 7.3. The performance measures which are applied in the experiment are described in section 7.4. After conducting the experiment, the results are interpreted in section 7.5. In the end, section 7.6 describes the robustness of the proposed Bayesian Model.

## 7.1 Using traffic characteristics in Bayesian models

Traffic propagates through a road network which is limited by its capacity. If the traffic demand is low and the capacity is high, traffic can freely flow through the network. Traffic is in the state of *free flow* if the maximum speed can be reached. These free flow situations are typically seen during the weekends, holidays or at night. Free flow in traffic is often not possible in cases of rush hours, road works, accidents, or extreme weather, since the capacity could be decreased by these kinds of situations. So, the success of a road network depends on the road- and network capacity, as well as the traffic demand. Knowledge about these traffic characteristics, or variables, is essential to understand traffic propagation which is important for doing traffic predictions. Details of these theories are presented in Chapter 2.

This Chapter considers the well known traffic bottleneck at the A4 left side between Hoogmade and Roelofarendsveen. Recall that, since orientation is relative, the choice for left and right is included in the unique ILD ID (BPS code). A situation overview is presented in Figure 7.1. The driving direction is indicated and the yellow push pins denote the induction loop detectors (ILD's). From this figure it becomes clear that we inspect the road between hectometer location 28.3 and hectometer location 19.6. It also becomes clear that there are no exit lanes and there is no merging traffic between these two hectometer locations.

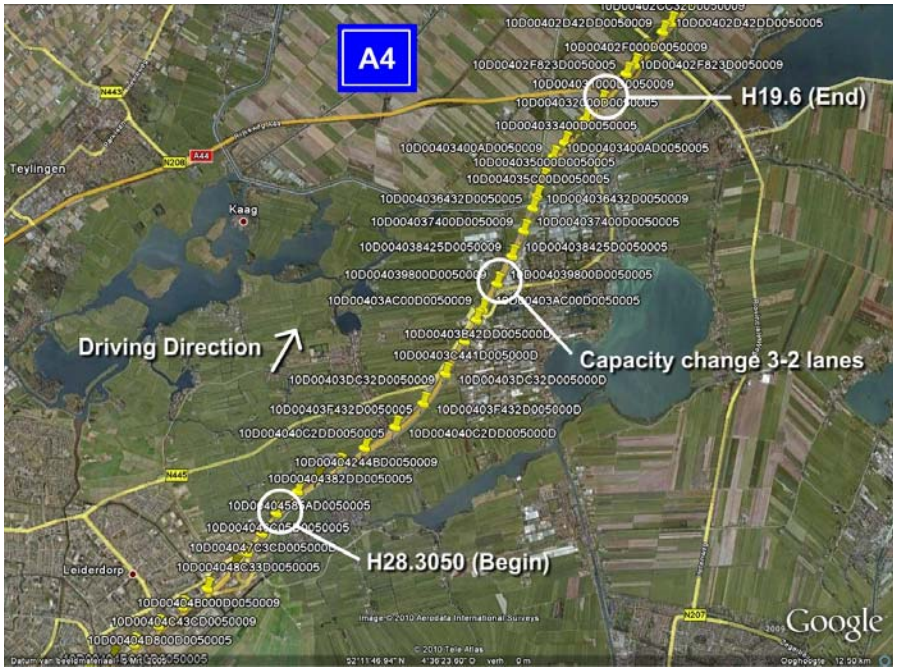


Figure 7.1: Situation overview A4

This part of the road is therefore likely to contain clean traffic data, which is not disturbed by a sudden decrease or increase of the traffic demand because of merging or exit lanes. Therefore, this location is a good starting point for a case study.

The prediction of average vehicle speed during morning rush hours is interesting, since commuter traffic travels between the Hague and Amsterdam in the morning during weekdays. Recall that commuter traffic is 64% during congestion in the Netherlands [30]. On this route, travelers have to pass the A4 between Hoogmade and Roelofarendsveen. This particular part of the road contains a capacity decrease from 3 to 2 lanes around hectometer location 23.0. This decrease in capacity yields traffic congestions regularly at this road.

By using common sense, it is plausible that the traffic flows freely in the early morning because the demand has not reached the capacity yet. At a certain moment, the roads become busier and the density (measured in vehicles per km) increases. Travelers are likely to drop their speed since the density becomes higher. It is plausible to say that a suddenly flip over point is nearing, where the speed drops down and the density goes up. This is the moment where the capacity is not sufficient enough to meet the traffic demand. The result is an unintentionally and frustrating traffic congestion.

By making use of these traffic characteristics, different Bayesian Models can be developed. The general model which we propose is illustrated in Figure 7.2, for $i$ time steps and $k$ locations. In this figure, $V$ denotes speed, $D$ denotes density and $T$ denotes the prediction horizon. This model incorporates speed and density measurements of several locations and at different times around a location of interest which will be predicted. This model is partly inspired by the work of Sun et al. [34], [35], [33] and by the work of Yu et al. [46], but the model we propose incorporates density measurements as well.

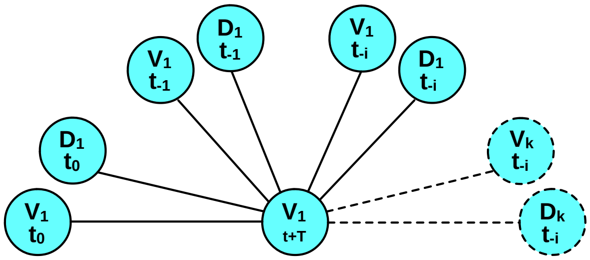


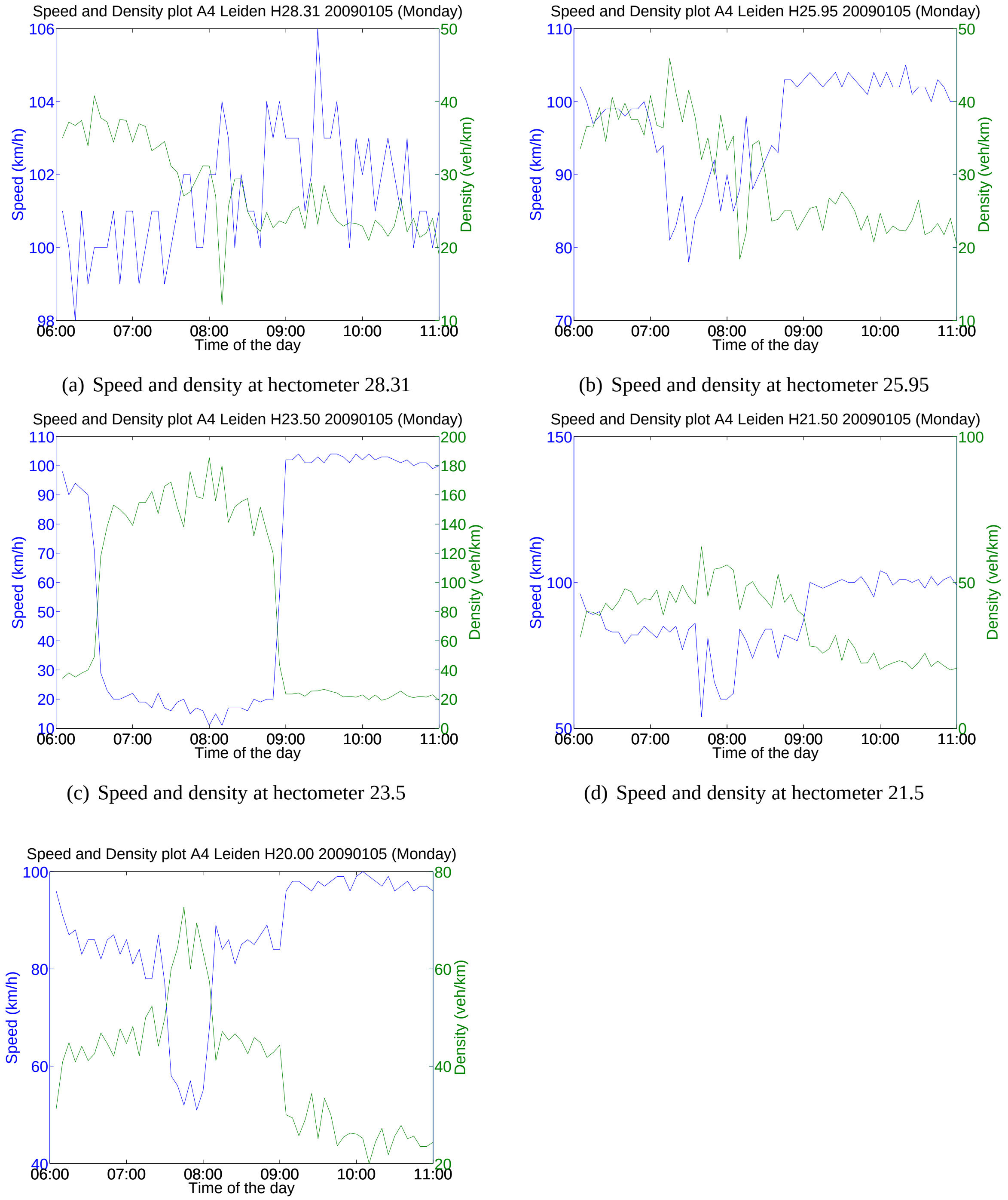
Figure 7.2: Proposed general Bayesian model

Before this model can be developed, there is a need to inspect the data whether this common sense is correct. Figure 7.3 represents combined speed and density plots for the locations at hectometer 28.31, 25.95, 23.5, 21.5 and 20.0 respectively for a regular Monday in 2009 between 06:00 am and 11:00 am. Recall that the driving direction of the traffic is from hectometer 28 to 19.

From these figures it becomes clear that there are indeed flipping points where the speed drops down and the density goes up. In the upstream location (hectometer 28.31) this flipping point is not visible. The speed graph only shows a stochastic behavior between 98 km/h and 106 km/h. The density shows a stochastic behavior between 10 veh/km and 50 veh/km. Further, it is clearly visible that there are flipping points in the downstream directions at hectometer location 23.5 and 20.0. At hectometer location 20.0 there is a speed drop just after 06:00 am. At the same time, the density goes up. This could indicate a switch from a free flow situation to a congested traffic situation. It is clearly visible that a congestion is present around 06:30 am at hectometer location 23.5, and a little bit later also further upstream. This meets our expectation that traffic information propagates against the driving direction during congested traffic as explained earlier in Chapter 4.

If traffic flow is seen as water flow, we are only able to choose our prediction horizon $T$ corresponding to the traffic speed. In a water flow model, upstream traffic is ’flowing’ downstream with a certain speed. If we predict the traffic situation at a location further downstream, we can only do so by calculating $T$ from the traffic speed. But, we assume that a traffic congestion grows by slow stages (gradually). Figure 7.3(c) shows a congestion which is already indicated earlier in the graph around 06:00 am. If a growth of a congestion is visible step by step in the graph, we are able to learn these indications in our Bayesian Network. Therefore, we do not see traffic as a water flow.

The graphs in Figure 7.3 show stochastic behavior after the flipping point between density and speed. We are interested whether these flipping points can be predicted using Bayesian Networks, but



(a) Speed and density at hectometer 28.31

(b) Speed and density at hectometer 25.95

(c) Speed and density at hectometer 23.5

(d) Speed and density at hectometer 21.5

(e) Speed and density at hectometer 20.0

Figure 7.3: Combined speed and density plots

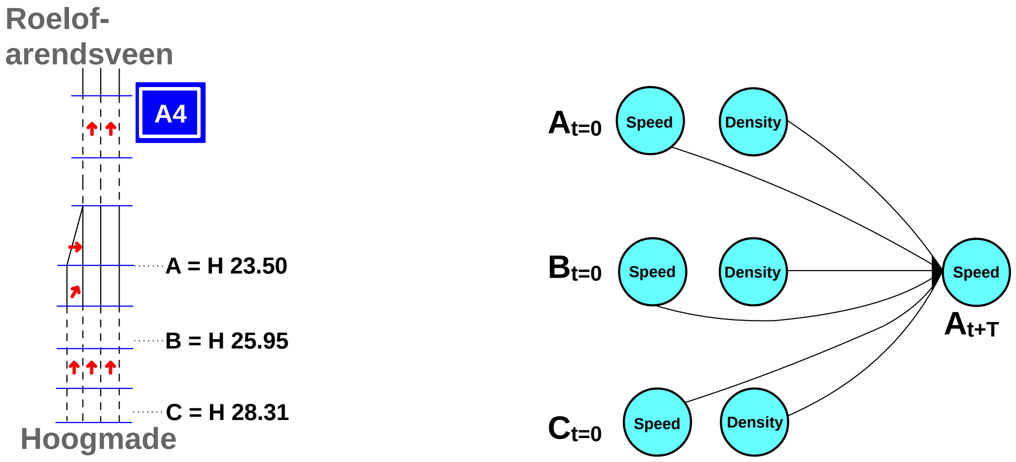
we do not tend to predict the stochastic behavior during a congestion or if traffic is in free flow.

## 7.2 Bayesian traffic prediction models

As explained earlier in Chapter 3, computing the exact conditional probability tables (CPTS) when there is missing data becomes intractable [10]. This said, it should not come as a surprise that there are limitations for the size and structure of Bayesian Networks. Therefore, at this stage of the research we decided to use a simplified network structure from the basic model we introduced in section 7.1. The simplified network consists of a maximum of 6 cause nodes which are causally connected directly to one effect node. For this, in total 6 arcs need to be drawn between the 6 cause nodes and the effect node. In total, there are 7 nodes, which gives us the opportunity to test simplistic models of the proposed general model in section 7.1. We expect that these simplistic models are capable of predicting average vehicle speed.

From Chapter 4 it becomes clear that traffic characteristics can propagate in the downstream and upstream direction. If a Bayesian Network needs to predict a traffic congestion (delay), i.e. a sudden drop of average driven speed for a certain time period, it can be trained on downstream or upstream locations. It would also be possible to train the network on both upstream and downstream locations. Different models are proposed in this section and further investigated throughout the remaining of this Chapter.

The first model which is presented only incorporates data in the upstream direction. A schematic overview and the Bayesian structure of this model are presented in Figures 7.4(a) and 7.4(b) respectively.



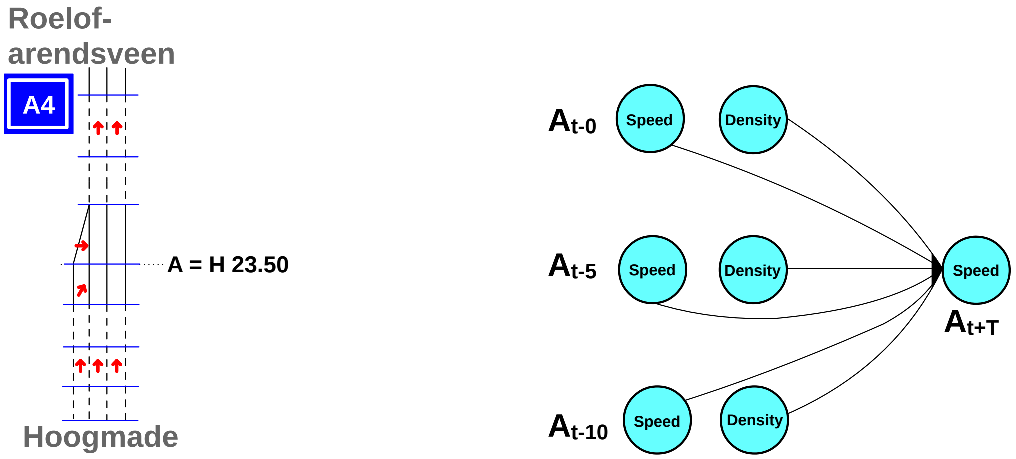
(a) Schematic overview for node locations A, B and C for Model 1

(b) Bayesian Model 1

Figure 7.4: Model 1 - A schematic overview

From these figures it becomes clear which road location the Bayesian nodes represent in the network. The aim is to predict the value of speed at location A over a certain prediction horizon $T$, with the help of location B and C.

Another important aspect in traffic propagation is time. The speed and density values are continuously changing over time, and from common sense it could be argued that previous speed and density values are good indicators for the future. Model 2 includes this time aspect and is presented in Figures 7.5(a) and 7.5(b).



(a) Schematic overview for node locations A, B and C for Model 2

(b) Bayesian Model 2

Figure 7.5: Model 2 - A schematic overview

Here, speed and density values of location A are taken at $t_0, t_{-5}$ and $t_{-10}$ where the time is measured in minutes. The aim is, to predict speed at location A, with the help of different time instants of speed and density data of location A.

Model 3 is presented in Figures 7.6(a) and 7.6(b) and incorporates both downstream and upstream data. The aim of model 3 is to predict speed at location A with downstream node C and upstream node B.

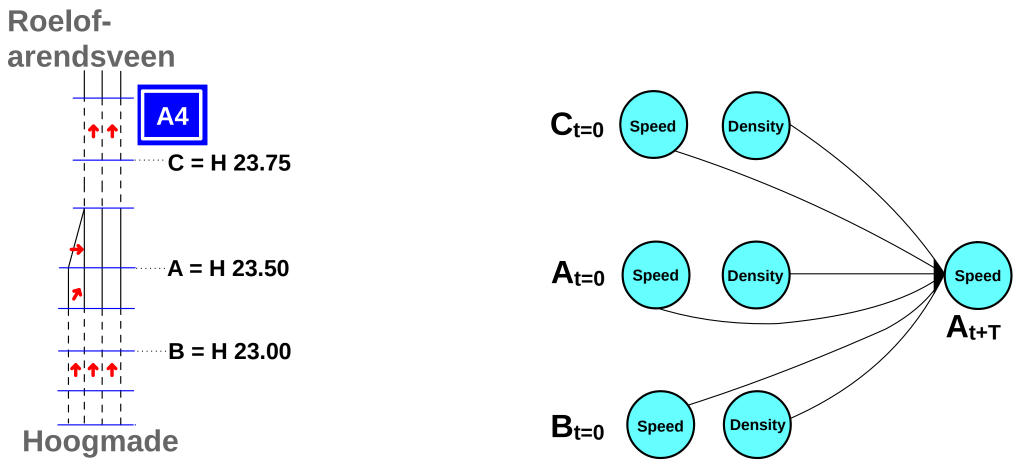
The 4th model considers only downstream nodes and is represented in Figures 7.7(a) and 7.7(b).

The aim of model 4 is to predict speed at location A, with the help of the downstream locations B and C.

## 7.3 Experimental setup

The Bayesian models, as introduced in section 7.2, are implemented and tested for evaluation in an experiment. It is important to explain the settings of this experiment for clarity purposes.

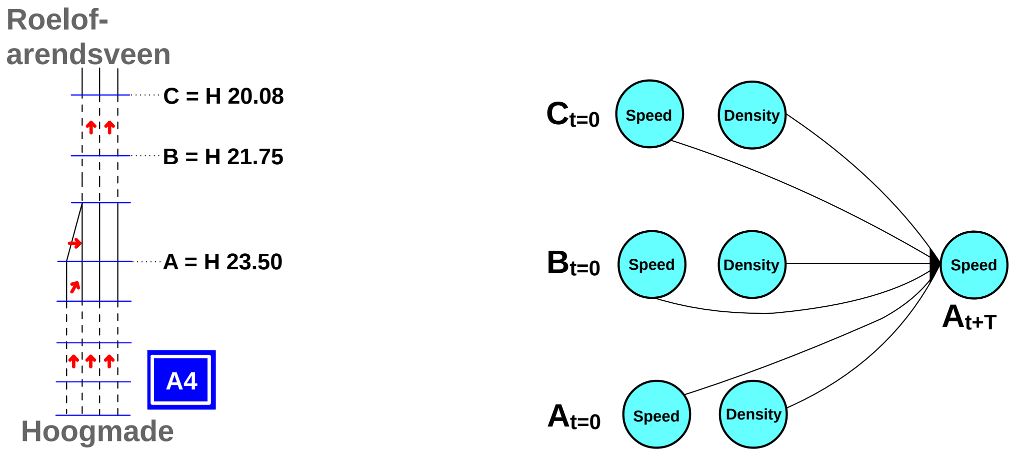
The nodes in the Bayesian networks represent the traffic variable speed or density. The speed is obtained from the MONICA ILD data, as explained earlier in Chapter 4. From this ILD data, the traffic intensities can also be obtained, to calculate the densities. Since the traffic data is aggregated



(a) Schematic overview for node locations A, B and C for Model 3

(b) Bayesian Model 3

Figure 7.6: Model 3 - A schematic overview



(a) Schematic overview for node locations A, B and C for Model 4

(b) Bayesian Model 4

Figure 7.7: Model 4 - A schematic overview

from lane data to road data, the intensities and densities are added over the lanes. The details of these calculations can be found in Chapter 2. The raw MONICA data is given in average values per minute. Since these values show strong stochastic behavior, it is decided to average this data over steps of 5 minutes. In this way, trends in the data become more visible and are more likely to represent the real traffic situation.

A MONICA ILD dataset is obtained for the year 2009. The data from the ILD's on the locations which are represented by the different nodes in the Bayesian models is selected and used for training and testing the Bayesian Networks. Since ILD's are not completely reliable, there is missing data. On average, 12% of the measurements are either missing or unreliable due to maintenance backlogs, temporal power or communication failure or for example incidents and accidents [41]. Bayesian Networks can handle incomplete datasets, since they offer a natural way to encode dependencies between the input variables [10]. Nevertheless, when there is too much data missing, it is hard to train a model to be capable for predictions. Therefore, the missing data has to be filled.

Filling missing values in traffic data is not a straightforward process. A sudden increase or decrease in speed can propagate in the downstream or upstream direction. Replacing missing values by average values is usually not adequate enough. Typically, a traffic congestion propagates upstreams with a speed of 15 km/h. These kind of traffic characteristics are incorporated in the Treiber and Helbing filter [38]. Details of this method can be found in Chapter 4. The traffic data used for this experiment is filtered with the Treiber and Helbing filter to fill in the missing values, by reconstructing the spatio-temporal traffic dynamics.

After applying the Treiber and Helbing filter, the data needs to be discretized for the Bayesian Models. Since we do not want to make assumptions about distributions, we decided to start with discrete Bayesian Networks. The speed values have a range from $0$ km/h to around $120$ km/h and are discretized into 6 classes (each of 20km/h). Density values typically lie in the range from $0$ veh/km to around $500$ veh/km and are discretized into 10 classes (each of 50 veh/km).

Since it is interesting to predict travel delays, i.e. traffic congestions, it is decided to concentrate on morning rush hours. Therefore, the traffic data is selected from 06:00 am until 11.00am, since the morning rush hours usually occur between these times.

The year 2009 has 365 days and it would be unfair to train and test the models on the same sets. Therefore, 80% of these days (292 days) are used for training and the remaining 20% (73 days) is used for testing. If the first 292 days would be selected for training, then the test set would contain only the last 73 days, which are winter days (Oktober, November and December). This might not be a correct representative set, because traffic might behave differently during winter and summer. Therefore, the days for the training set are randomly drawn from the total set. The remaining set represents the test set. In this way, it is tried to get a representative set.

The models are trained on the training data and tested on the test data. The performances of the models need to be computed and evaluated. Section 7.4 describes the performance measures which have been used.

## 7.4 Performance measures

There are numerous accuracy measures available for the estimation of the performance of predictors. A recently published overview reviews all accuracy measures from the last 25 years of research in [9]. A convenient overview which includes the most commonly used forecast accuracy measures is presented in table form in appendix A. The mean absolute percentage error (MAPE) is most often used for evaluating the accuracy of forecasts [37]. Tayman et al. state that any summary measure of error should meet 5 highly desirable criteria [37]:

1. Validity
2. Reliability

3. Easy to interpret
4. Clarity of presentation
5. Support of statistical evaluation.

It is claimed by Tayman et al., that the MAPE lacks the validity criterion for evaluating time series, since it has a tendency to overestimate the error. Details of this claim and an explanation can be found in [37]. The MAPE can be described in mathematical form as follows:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right|, \tag{7.1}$$

where $A_t$ is the actual value and $F_t$ is the forecast value.

From the formula, two important things can be seen. Firstly, MAPE is zero when there is a perfect fit. But in its upper level, there is no restriction. If some MAPE values are very high, it will influence the average and makes these values hard to compare. Secondly, if there are zero values, then there will be a division by zero. Therefore, all the values have been shifted with $+1$, so that there are no zero values present to eliminate this disadvantage.

There are alternatives to the shortcomings of the MAPE. The symmetrical MAPE (sMAPE) is an accuracy measure based on relative errors [8]. It is usually defined as follows:

$$sMAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t + F_t/2}\right|, \tag{7.2}$$

where $A_t$ is the actual value and $F_t$ is the forecast value.

A somewhat more convenient form of this error estimator is proposed in [1] and is expressed as follows:

$$sMAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t + F_t}\right|, \tag{7.3}$$

where $A_t$ is the actual value and $F_t$ is the forecast value. The sMAPE has an upper and a lower bound, which makes it very convenient during the comparison of different forecasting methods applied to data. Formula 7.3 has the extra convenience that the values lie between $0\%$ and $100\%$.

Besides comparing the prediction values with the real data, we introduce another, naive, prediction model for comparison purposes. The NAIVE model predicts the speed values according to the following formula:

$$V_{t+T} = V_t, \tag{7.4}$$

where $V$ denotes the variable speed, $t$ denotes the current time and $T$ denotes the prediction horizon. So the NAIVE predictor actually takes the current value and behaves if nothing happens in the future. This is actually a very powerful predictor in traffic, since traffic has many stable phases before and after changing phases. If the traffic is in free flow, it is very likely that it will be in free flow for the next $x$ minutes. After these $x$ minutes, the NAIVE predictor makes an error when the average speed drops down, but this usually only takes a few minutes. Usually, after a sudden speed drop for a few minutes, traffic becomes congested and stays congested for a period in which this NAIVE predictor is giving correct predictions, until another change of traffic is nearing.

This brings us to the point where it is important to explain what a good performing predictor is. A good predictor, does not necessarily need to predict the speed at every moment correctly, but it should be able to predict the start/end of a congestion accurately. If a sudden drop in speed can be predicted, this would be highly valuable to travelers, since they could anticipate for this. Actually, it is important that future traffic can be classified as congested or as free flow. For this, there are two possible errors: *false positives* and *false negatives*.

**Definition 8.** *False Positives are situations where a predictor predicts a congestion, while there is none.*

**Definition 9.** *False Negatives are situations where a predictor predicts no congestion, while there is one.*

At this stage, it is important to give a definition for the term: *congestion*. Since there are numerous definitions available in literature for congestion, we will not list them here. A very simple, but clear definition of a traffic congestion is defined by the Verkeers Informatie Dienst (Traffic Information Service) [1] in the Netherlands as follows:

**Definition 10.** *A congestion is a situation where the average vehicle speed is below 50 km/h for at least 2 minutes.*

A situation of *false negatives* is undesirable, since travelers have to cope with unforeseen delays. In the situation where there is a *false positive*, drivers will be happy since the predicted delay turns out to be absent. Therefore, the Bayesian Models and the NAIVE predictor are also evaluated on there false positives and false negatives.

## 7.5 Results and interpretation

The MAPE over the Bayesian Models 1-4 and the NAIVE model are presented in Table 7.1. As becomes clear from this table, the differences between these models are marginal. This conclusion should be taken with caution, since it is hard to compare these MAPE values, as explained in section 7.4.

Table 7.2 presents the sMAPE values for all models. This values lie in the range between $0\%$ and $100\%$, so they are easily comparable. In general, the models perform the same, but a difference can be seen when the time horizon grows to 120 minutes. The models 1-4 have a tendency to perform better then the NAIVE model in the longer term. This should not come as a surprise, since the NAIVE model is 120 minutes late in predicting the speed if the prediction horizon is 120 minutes.

More interesting to see is how good the prediction models perform on false positives and false negatives. The question is where and how these errors are made? It becomes clear from Table 7.3 that the differences between the models are marginal. In general, the models 1-4 perform slightly worse then the NAIVE model. A large difference in performance can be seen for the prediction for over 120

[1] More details can be found at www.vid.nl

Table 7.1: MAPE for models: 1-4 and the NAIVE model

| Pred hor (min) | Model 1 | Model 2 | Model 3 | Model 4 | NAIVE Model |
|---|---|---|---|---|---|
| 5 | 0.075 | 0.074 | 0.076 | 0.069 | 0.072 |
| 10 | 0.11 | 0.11 | 0.11 | 0.095 | 0.11 |
| 15 | 0.15 | 0.13 | 0.14 | 0.12 | 0.13 |
| 20 | 0.16 | 0.15 | 0.16 | 0.15 | 0.15 |
| 25 | 0.19 | 0.17 | 0.18 | 0.16 | 0.16 |
| 30 | 0.20 | 0.18 | 0.19 | 0.18 | 0.17 |
| 35 | 0.22 | 0.21 | 0.21 | 0.19 | 0.19 |
| 40 | 0.23 | 0.22 | 0.23 | 0.21 | 0.20 |
| 60 | 0.26 | 0.27 | 0.30 | 0.25 | 0.24 |
| 120 | 0.35 | 0.32 | 0.34 | 0.35 | 0.30 |

Table 7.2: sMAPE for models: 1-4 and the NAIVE model

| Pred hor (min) | Model 1 | Model 2 | Model 3 | Model 4 | NAIVE Model |
|---|---|---|---|---|---|
| 5 | 0.037 | 0.035 | 0.038 | 0.034 | 0.034 |
| 10 | 0.053 | 0.049 | 0.052 | 0.047 | 0.049 |
| 15 | 0.064 | 0.058 | 0.063 | 0.057 | 0.059 |
| 20 | 0.071 | 0.066 | 0.072 | 0.067 | 0.067 |
| 25 | 0.079 | 0.073 | 0.078 | 0.072 | 0.073 |
| 30 | 0.087 | 0.080 | 0.084 | 0.078 | 0.079 |
| 35 | 0.093 | 0.089 | 0.091 | 0.084 | 0.085 |
| 40 | 0.098 | 0.098 | 0.10 | 0.090 | 0.092 |
| 60 | 0.11 | 0.11 | 0.12 | 0.12 | 0.11 |
| 120 | 0.11 | 0.10 | 0.11 | 0.11 | 0.15 |

minutes. This could be due to the fact that after 2 hours, most of the congestions in the morning have dissolved. After inspection, it became clear that the Bayesian models have a tendency to predict free flow more often in this situation. If free flow is continuously predicted, then obviously there are less false positives, since there are less congestions predicted in situations where there is no congestion.

Table 7.4 presents the results of the false negatives of the models. It becomes clear that Model 4 outperforms all the other models. For example: for a prediction over 10 minutes, model 4 has an false negatives error of $4.2\%$, while the NAIVE model has an error of $12\%$. This means, that model 4 is far better in prediction traffic congestions then the NAIVE model, which is of high importance.

For the sake of completeness, several prediction results for Model 4 are visualized in Figure 7.8. Figure 7.8(a) represent a prediction for a regular Tuesday in November. It becomes clear that Model 4 is able to predict the traffic congestion, with a prediction horizon of 10 minutes, earlier then the NAIVE model. Further, it can be seen that Model 4 behaves comparable for the rest of the morning rush hour to the NAIVE model so there is a need to improve Model 4.

From Figure 7.8(b) it becomes clear that Model 4 as well as the NAIVE model both have a zero

Table 7.3: False Positives for models: 1-4 and the NAIVE model

| Pred hor (min) | Model 1 | Model 2 | Model 3 | Model 4 | NAIVE Model |
|---|---|---|---|---|---|
| 5 | 0.040 | 0.036 | 0.046 | 0.043 | 0.035 |
| 10 | 0.097 | 0.068 | 0.084 | 0.083 | 0.057 |
| 15 | 0.12 | 0.083 | 0.099 | 0.067 | 0.074 |
| 20 | 0.13 | 0.097 | 0.10 | 0.12 | 0.088 |
| 25 | 0.10 | 0.11 | 0.11 | 0.13 | 0.10 |
| 30 | 0.11 | 0.13 | 0.12 | 0.14 | 0.11 |
| 35 | 0.12 | 0.14 | 0.11 | 0.15 | 0.12 |
| 40 | 0.12 | 0.14 | 0.13 | 0.15 | 0.13 |
| 60 | 0.16 | 0.17 | 0.15 | 0.19 | 0.18 |
| 120 | 0.029 | 0.025 | 0.046 | 0.11 | 0.28 |

Table 7.4: False Negatives for models: 1-4 and the NAIVE model

| Pred hor (min) | Model 1 | Model 2 | Model 3 | Model 4 | NAIVE Model |
|---|---|---|---|---|---|
| 5 | 0.075 | 0.072 | 0.064 | 0.054 | 0.073 |
| 10 | 0.078 | 0.096 | 0.077 | 0.042 | 0.12 |
| 15 | 0.11 | 0.12 | 0.11 | 0.069 | 0.15 |
| 20 | 0.13 | 0.15 | 0.16 | 0.097 | 0.18 |
| 25 | 0.20 | 0.17 | 0.17 | 0.12 | 0.20 |
| 30 | 0.23 | 0.17 | 0.21 | 0.14 | 0.21 |
| 35 | 0.24 | 0.19 | 0.28 | 0.15 | 0.22 |
| 40 | 0.28 | 0.22 | 0.28 | 0.20 | 0.24 |
| 60 | 0.34 | 0.33 | 0.44 | 0.29 | 0.29 |
| 120 | 0.96 | 0.87 | 0.89 | 0.35 | 0.47 |

prediction error on a regular Saturday in January with a prediction horizon of 10 minutes. This is probably due to the fact that there is continuously a free flow situation. A congestion is not detected by Model 4, which is correct.

Figure 7.8(c) shows a Sunday congestion, probably due to road works, nice weather or holiday. The congestion starts just after 08:00 am, which is later then on a regular working day. It is clearly visible that Model 4 predicts this congestion (with a prediction horizon of 25 minutes) far earlier then the NAIVE model. Model 4 also succeeds in predicting the end of the congestion quite accurately.

Figure 7.8(d) shows the prediction results for T=30 for a regular Thursday. The figure shows that the Bayesian Network Model 4 is better in predicting the start and the end of the congestion, compared to the NAIVE model. During the congestion, the Bayesian Model shows stochastic behavior which has a negative effect on the MAPE and the sMAPE.

Figures 7.8(e) and 7.8(f) show the prediction result for a regular Friday and Monday respectively with prediction horizons of T = 40 and T= 60 respectively. Both figures show that the Bayesian Model 4 is able to predict the start and end of the congestion better then the NAIVE model, but the model shows stochastic behavior during the congestion. Again, this has a negative effect on the MAPE and sMAPE. The results of these prediction results are promising, but not convincing yet.

## 7.6 Robustness

This section investigates the robustness of Bayesian Model 4. Bayesian Model 4 clearly outperforms the other Bayesian models in situations of congested traffic. To get an impression of the robustness of this model, it has been tested for several different subsamples. A subsample of weekend data has been excluded, since the prediction error on weekends is almost zero since there are almost no morning congestions during weekends.

An experiment has been conducted for this section in which the Bayesian Network model 4 is trained for Tuesdays, Wednesdays and weekdays respectively. For each experiment, the performance of the NAIVE model is also inspected for comparison purposes.

Table 7.5 shows the MAPE for all models. It becomes clear that the MAPE for Wednesdays ia slightly higher, but the difference is marginal. The MAPE values for all models are comparable.

Table 7.6 shows the values for the sMAPE for all models. As with the MAPE values, there are only marginal differences between the models.

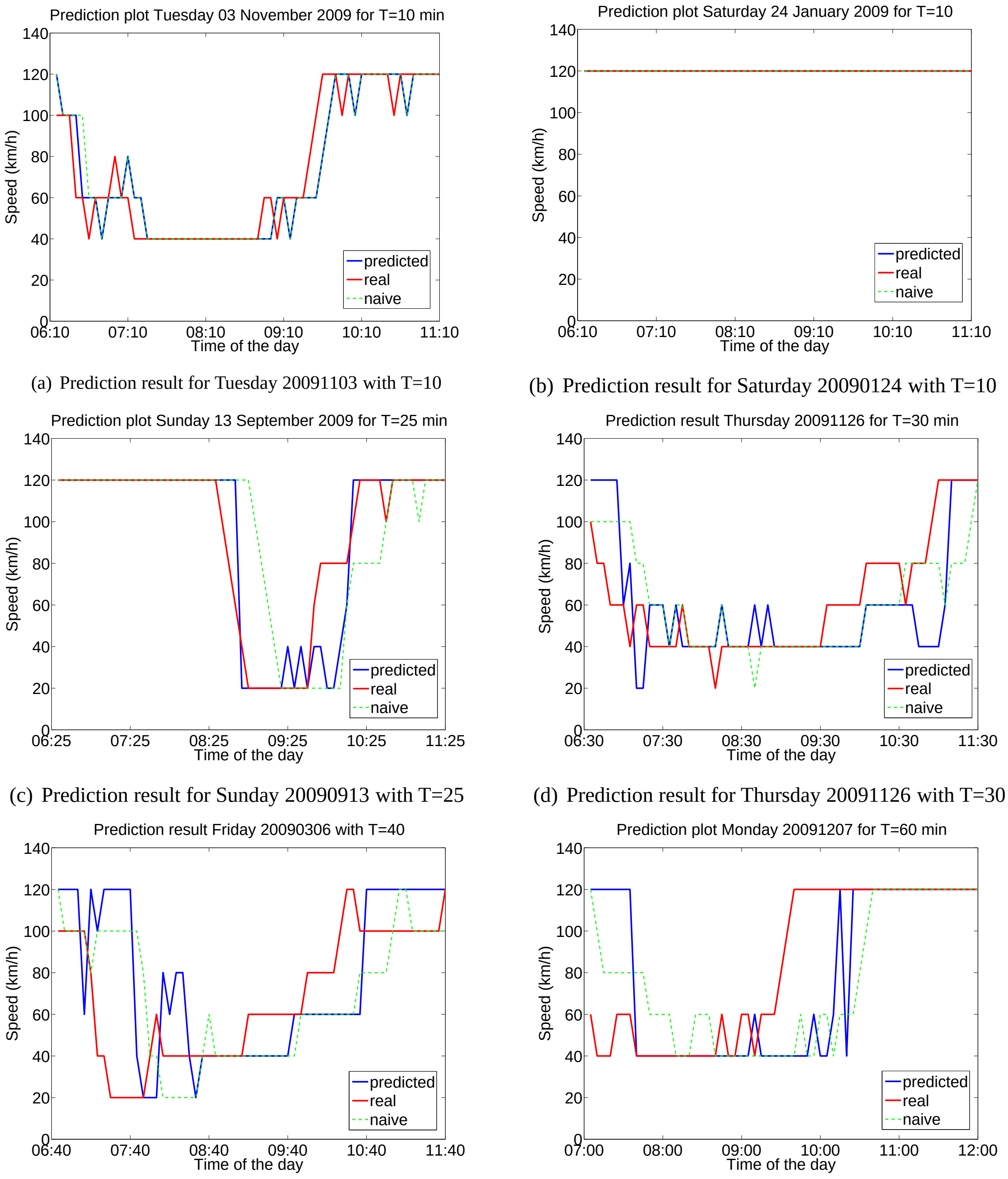
Table 7.7 shows the error percentages for the false positives. As becomes clear form these results, there are only marginal differences between the false positive percentages.

Table 7.8 shows the error percentages of the false negatives. Again, there are only marginal differences between the models visible.

Based on the results in this section, it seems that the performance of the Bayesian Network model 4 on the MAPE, sMAPE, false positives and false negatives remain the same for different subsamples of the data. Therefore, the Bayesian Model seems to be robust.

For comparison purposes, Figure 7.9 shows the graphs for the sMAPE, false positives and false negatives. Figure 7.9(b) shows that the performance of the NAIVE model is slightly lower then the Bayesian Model 4 until a prediction horizon of 40 minutes. After 40 minutes, the performance of the NAIVE model becomes significantly worse. Figure 7.9(c) shows that the Bayesian Model 4 performs better then the NAIVE model for weekdays and Tuesdays, but it seems harder to predict the traffic on Wednesdays.

In general, we found that Bayesian Network Model 4 is robust and generally performs slightly better then the NAIVE model. We conclude that traffic predictions with Bayesian Networks give promising results for the case study conducted in this Chapter, but there certainly is a need for improvements.



(a) Prediction result for Tuesday 20091103 with T=10

(b) Prediction result for Saturday 20090124 with T=10

(c) Prediction result for Sunday 20090913 with T=25

(d) Prediction result for Thursday 20091126 with T=30

(e) Prediction result for Friday 20090913 with T=40

(f) Prediction result for Monday 20091126 with T=60

Figure 7.8: Prediction results

Table 7.5: MAPE

| Pred hor (min) | NAIVE Tue | Mod 4 Tue | NAIVE Wed | Mod 4 Wed | NAIVE Week | Mod 4 Week |
|---|---|---|---|---|---|---|
| 5 | 0.081 | 0.089 | 0.10 | 0.094 | 0.11 | 0.097 |
| 10 | 0.12 | 0.12 | 0.14 | 0.13 | 0.15 | 0.14 |
| 15 | 0.15 | 0.16 | 0.17 | 0.18 | 0.19 | 0.18 |
| 20 | 0.17 | 0.16 | 0.20 | 0.22 | 0.21 | 0.22 |
| 25 | 0.19 | 0.18 | 0.22 | 0.25 | 0.24 | 0.25 |
| 30 | 0.21 | 0.20 | 0.26 | 0.27 | 0.26 | 0.27 |
| 35 | 0.22 | 0.23 | 0.28 | 0.30 | 0.28 | 0.29 |
| 40 | 0.24 | 0.23 | 0.30 | 0.35 | 0.31 | 0.32 |
| 60 | 0.29 | 0.27 | 0.37 | 0.43 | 0.37 | 0.38 |
| 120 | 0.36 | 0.43 | 0.55 | 0.64 | 0.45 | 0.56 |

Table 7.6: sMAPE

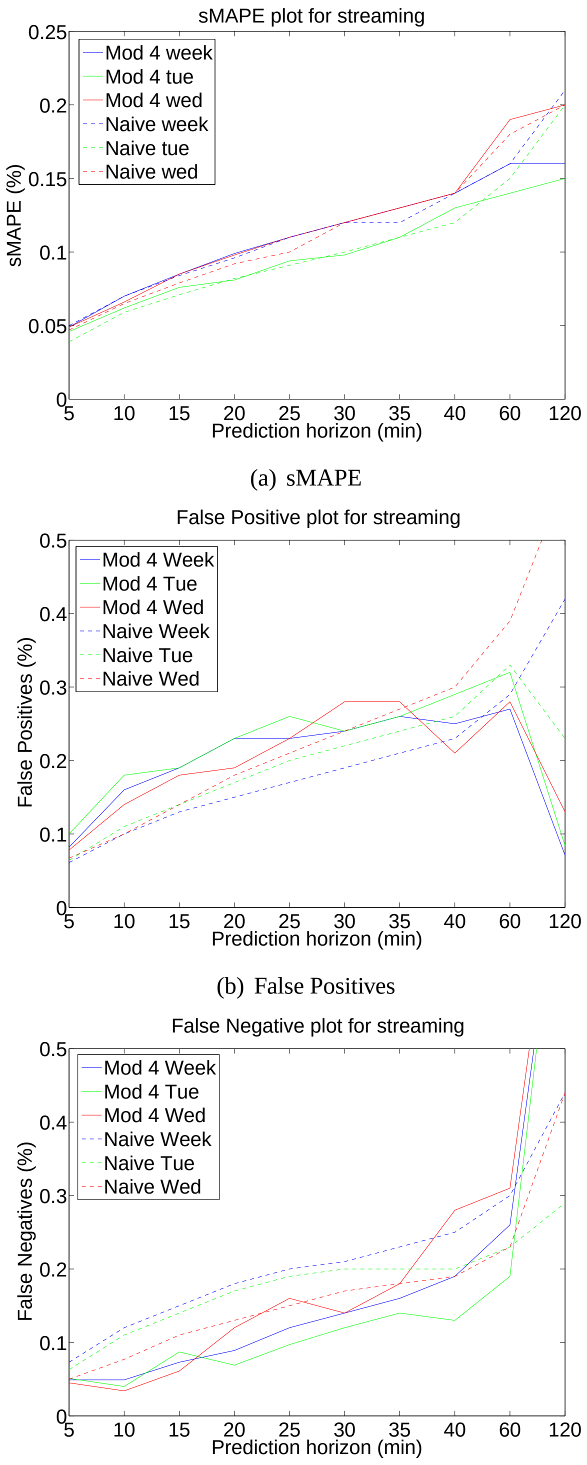
| Pred hor (min) | NAIVE Tue | Mod 4 Tue | NAIVE Wed | Mod 4 Wed | NAIVE Week | Mod 4 Week |
|---|---|---|---|---|---|---|
| 5 | 0.039 | 0.046 | 0.047 | 0.049 | 0.05 | 0.049 |
| 10 | 0.059 | 0.062 | 0.065 | 0.066 | 0.07 | 0.07 |
| 15 | 0.071 | 0.076 | 0.079 | 0.085 | 0.084 | 0.085 |
| 20 | 0.082 | 0.081 | 0.092 | 0.098 | 0.096 | 0.099 |
| 25 | 0.091 | 0.094 | 0.10 | 0.11 | 0.11 | 0.11 |
| 30 | 0.10 | 0.098 | 0.12 | 0.12 | 0.12 | 0.12 |
| 35 | 0.11 | 0.11 | 0.13 | 0.13 | 0.12 | 0.13 |
| 40 | 0.12 | 0.13 | 0.14 | 0.14 | 0.14 | 0.14 |
| 60 | 0.15 | 0.14 | 0.18 | 0.19 | 0.16 | 0.16 |
| 120 | 0.20 | 0.15 | 0.26 | 0.20 | 0.21 | 0.16 |

Table 7.7: False Positives

| Pred hor (min) | NAIVE Tue | Mod 4 Tue | NAIVE Wed | Mod 4 Wed | NAIVE Week | Mod 4 Week |
|---|---|---|---|---|---|---|
| 5 | 0.064 | 0.10 | 0.067 | 0.078 | 0.061 | 0.082 |
| 10 | 0.11 | 0.18 | 0.10 | 0.14 | 0.10 | 0.16 |
| 15 | 0.14 | 0.19 | 0.14 | 0.18 | 0.13 | 0.19 |
| 20 | 0.17 | 0.23 | 0.18 | 0.19 | 0.15 | 0.23 |
| 25 | 0.20 | 0.26 | 0.21 | 0.23 | 0.17 | 0.23 |
| 30 | 0.22 | 0.24 | 0.24 | 0.28 | 0.19 | 0.24 |
| 35 | 0.24 | 0.26 | 0.27 | 0.28 | 0.21 | 0.26 |
| 40 | 0.26 | 0.29 | 0.30 | 0.21 | 0.23 | 0.25 |
| 60 | 0.33 | 0.32 | 0.39 | 0.28 | 0.29 | 0.27 |
| 120 | 0.23 | 0.084 | 0.58 | 0.13 | 0.42 | 0.071 |

Table 7.8: False Positives

| Pred hor (min) | NAIVE Tue | Mod 4 Tue | NAIVE Wed | Mod 4 Wed | NAIVE Week | Mod 4 Week |
|---|---|---|---|---|---|---|
| 5 | 0.063 | 0.051 | 0.05 | 0.045 | 0.073 | 0.049 |
| 10 | 0.11 | 0.040 | 0.077 | 0.034 | 0.12 | 0.049 |
| 15 | 0.14 | 0.087 | 0.11 | 0.061 | 0.15 | 0.073 |
| 20 | 0.17 | 0.069 | 0.13 | 0.12 | 0.18 | 0.089 |
| 25 | 0.19 | 0.097 | 0.15 | 0.16 | 0.20 | 0.12 |
| 30 | 0.20 | 0.12 | 0.17 | 0.14 | 0.21 | 0.14 |
| 35 | 0.20 | 0.14 | 0.18 | 0.18 | 0.23 | 0.16 |
| 40 | 0.20 | 0.13 | 0.19 | 0.28 | 0.25 | 0.19 |
| 60 | 0.23 | 0.19 | 0.23 | 0.31 | 0.30 | 0.26 |
| 120 | 0.29 | 0.83 | 0.44 | 0.85 | 0.44 | 0.81 |



(a) sMAPE

(b) False Positives

(c) False Negatives

Figure 7.9: Robustness Results

# Chapter 8

# Historical Model

In this Chapter we propose a prediction model based on historical data. Since the Bayesian prediction model, as described in Chapter 7, does not perform well for longer prediction horizons, there is a need for a better model. Predicting traffic speed for longer prediction horizons becomes harder. We expect that historical knowledge should be taken into account, to predict the traffic speed more accurately for these longer prediction horizons. Since the MONICA database consists of several historical speed and intensity measurements, this data can be used to predict new situations.

The MONICA database consists of raw historical data. In Chapter 6 it was described how we could develop specific traffic models for different clusters. This Chapter combines the historical raw data and the models for the purpose of traffic prediction. The prediction model based on historical data we propose is tested and evaluated for its performance on traffic prediction.

Section 8.1 describes how the historical data can be used for traffic prediction purposes. Then, section 8.2 describes the prediction model based on historical data which we propose. The experiment is described in section 8.4 and the performance measures are described in section 8.5. The results and interpretations are given in section 8.6.

## 8.1 Using historical knowledge for traffic prediction

Chapter 6 investigates speed, intensity and density graphs in a large historical database for the well known traffic bottleneck at the A4 left side around hectometer location 23.5. This investigation showed that speed, intensity and density graphs for the year 2009 could be grouped into the following clusters:

- Monday-Thursday
- Friday
- weekend
- holiday or public holiday.

These clusters contain homogeneous data and the mean patterns are different for each cluster. These clusters can be made for the variables speed, intensity and density. More details about these clusters and data analysis can be found in Chapter 6.

Typical traffic models can be extracted from these clusters by taking the mean graph of each set. The main idea in this Chapter is to use these models for traffic prediction purposes. New traffic data can be similar to these traffic models. If, for example, today's speed measurements are known between 00:00 am and 06:00 am, a best fitting model in a database can be found by comparing distances to the models until 06:00 am. This best fitting model incorporates a full day of data known from the past. Therefore, a prediction for, lets say 06:30 am, can easily be read from this best fitting model.

A mean model of a cluster might not always be representative for the current day, since models can only represent average situations. In some cases, it might be useful to inspect a raw data base which consists of raw day data. This data is not averaged and is likely to contain outliers: typical days with incidents, accidents, heavy rain, etc. For example, if there is an accident today it would be unlikely that an averaged model can represent this day accurately. In the raw database there could be another day in the history on which there was a same kind of accident as well. The data of this day could be similar to that specific day in the raw data base, which might be a better model to use for prediction. Therefore, it is wise to compare distances to a *model database*, as well as to a raw *historical database*.

A travel prediction model based on historical data can be developed in different ways. There are numerous ways to calculate distances between models and to make use of model or raw data. Section 8.2 describes which models have been developed.

## 8.2 Historical models

The prediction model based on historical data is proposed in this section. A schematic overview of this prediction model based on historical data is presented in Figure 8.1. This model consists of an algorithm which receives the current traffic *data* and a prediction horizon *h* as arguments. The current traffic data could be speed measurements for day $d$ from time $t_1$ until $t_c$, where $t_c$ represents the current time. The prediction horizon parameter *h* indicates time $t_h$ for which a prediction is asked.

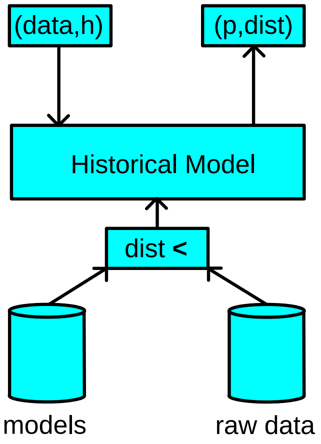


Figure 8.1: Prediction model based on historical data

The prediction model based on historical data calculates the distances from the current data to the models in the model database. This process yields the best fitting model which has a certain fitting distance.

Next, the algorithm of the prediction model based on historical data calculates the distances to all the historical day patterns in the raw historical database. This yields a raw day pattern and a distance to it as well. After this process, the distance to the best fitting model and to the best fitting historical raw day pattern is compared and the one with the lowest distance is returned. In the end, the algorithm reads the speed measurement at $t_h$ from the best fitting pattern or model, where $t_h$ is the time moment for which the prediction is asked. Then, the algorithm returns this prediction together with its distance to the best fitting model or raw day pattern.

There are numerous distance measures available to calculate the difference between two graphs. The first distance measure which is used, is the total absolute distance measure which can be expressed in formula form as in equation 8.1:

$$d = \sum_{t=1}^{c} \left| R_t - M_t \right| , \tag{8.1}$$

where $R_t$ is the real data at time $t$, $M_t$ is the model or historical raw data at time $t$ and $t_i = 1...c$ where $t_c$ is the current time. The second distance measure which is used in the prediction model is the sMAPE. This distance measure was also used to calculate the performance of the Bayesian Network in Chapter 7, but in this case the distance measure only calculates the difference until a certain time instant $t_c$:

$$sMAPE = \frac{1}{c} \sum_{t=1}^{c} \left| \frac{R_t - M_t}{R_t + M_t} \right| , \tag{8.2}$$

where $R_t$ is the real data at time $t$, $M_t$ is the model or raw data at time $t$ and $t_i = 1...c$, where $t_c$ is the current time.

The algorithm is asked for a new prediction if time is proceeding and new data points are received. The prediction model is able to switch between the models or raw day patterns at any time. Therefore, the algorithm automatically ’learns’ the best model and updates continuously. This adaptive algorithm will be explained in section 8.3

To evaluate the performance of this prediction model based on historical data, an experiment is conducted and described in section 8.4

## 8.3 Adaptive learning

Our prediction model, which is based on historical data, adapts itself to new data continuously. This means that the prediction model is able to switch from a certain historical model to, for example, a specific ’accident pattern’ or ’extreme weather’ pattern in the raw database in case of incidents.

Figure 8.2 shows a situation where the current time is around 08.00 am. The speed measurements are collected from 00:00 am. This figure shows that the real data is similar to a certain historical model in the database between 00:00 am and 06:00 am. Later that day, the real data becomes more similar to a certain day in the raw historical database. We can clearly see that the historical model is not representative anymore for the real data. We expect that most days are similar during the early morning, but along the day there could be an accident, extreme weather, or other incidents.

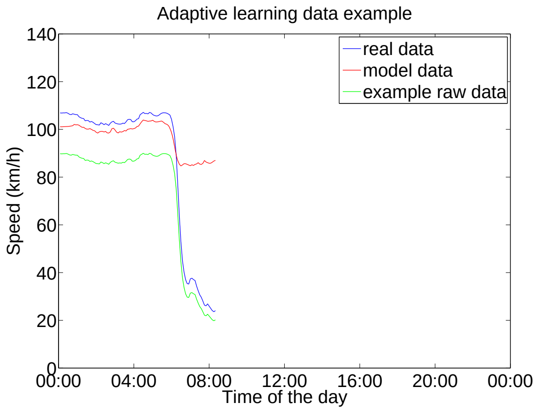


Figure 8.2: Adaptive learning model data example

This knowledge can be learned along the day. Therefore, we propose a prediction model which continuously measures the distance between the real data and the historical models as well as the raw database. In this way, our prediction model is able to select the most representative historical data.

## 8.4 Experimental setup

For the experiment in this Chapter, a training set has been extracted from the total set of traffic data in 2009 for 3 ILD locations at the A4 left side around hectometer location 23.5. These ILD locations correspond to the locations used in the Bayesian Model 4 as described in Chapter 7. From this traffic data, the speed and density measurements are selected and grouped into the clusters which are introduced in section 8.1. Figure 8.3 illustrates the models for speed and density for the 3 ILD locations.

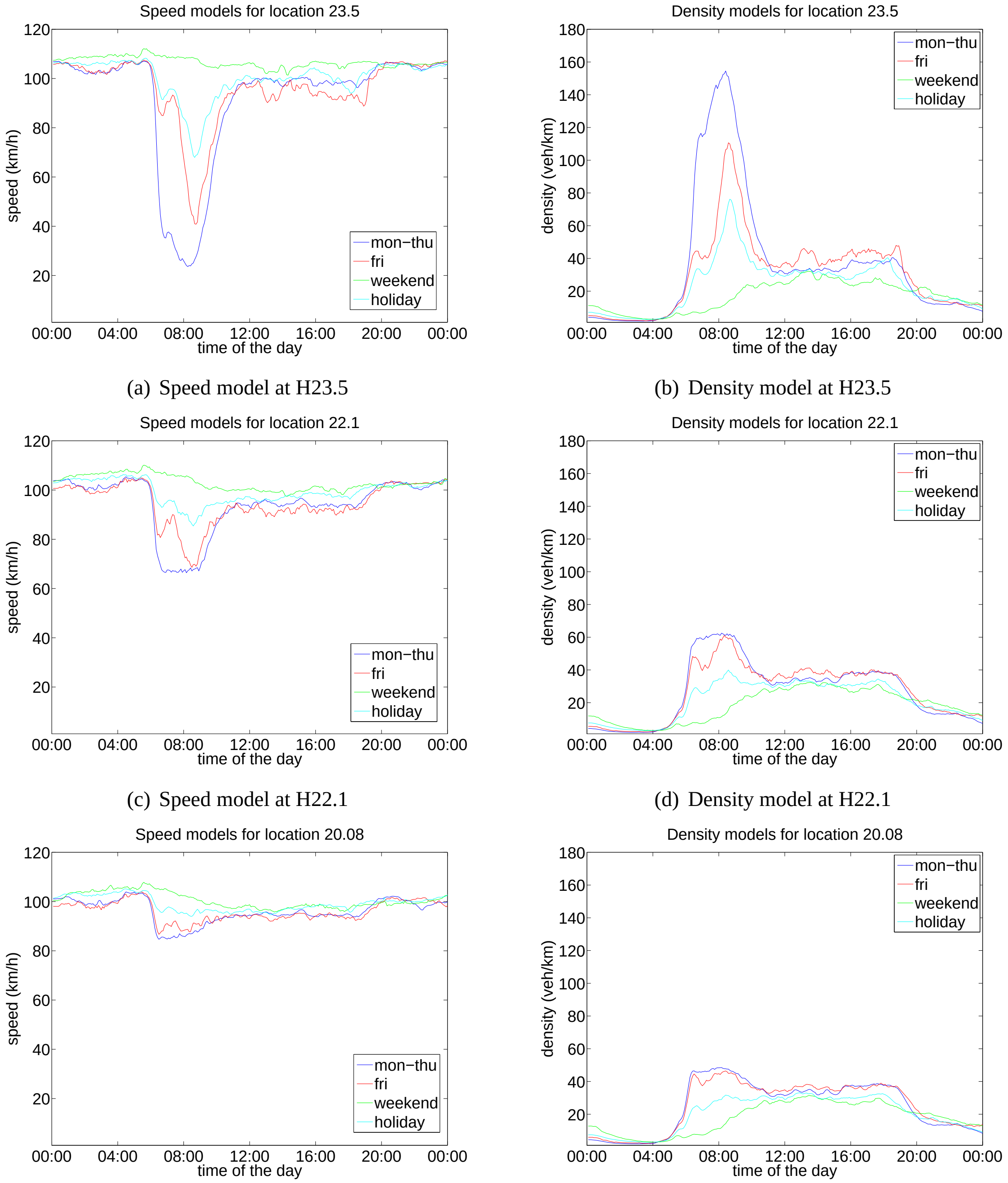
It becomes clear from this figure, that the differences between the models are the largest at the bottleneck itself (location 23.5), whereas the difference fades out at the other locations. Nevertheless, there are still differences visible between the models.

From the same train set, the historical raw database is created in which all the raw day data are stored. This data is, of course, not average and is likely to contain outliers such as accidents, extreme weather, or incidents.

Since traffic data, which is received by ILDs and collected in the MONICA database, contains missing values, the dataset has been filtered with the Treiber and Helbing [38] filter to fill in the missing data points. Details of this process are described in Chapter 4.

After filtering the data, the data is separated into a train and a test set which consists of 80% and 20% of the total set respectively.

In this experiment, predictions for the bottleneck at hectometer location 23.5 are investigated. The speed and density models, as visualized in Figure 8.3, are created only for this location. The predictions for the experiment are calculated between 06.00 am and 11.00 am, since these are the morning rush hours.



(a) Speed model at H23.5

(b) Density model at H23.5

(c) Speed model at H22.1

(d) Density model at H22.1

(e) Speed model at H20.08

(f) Density model at H20.08

Figure 8.3: Model database

To evaluate the performance of the prediction algorithm, the performance measures are described in section 8.5.

## 8.5 Performance measures

The performance measures which are used to evaluate this experiment are the same as the one used to evaluate the Bayesian Network models. Details of these performance measures can be found in Chapter 7. For readability purposes, the performance measures are briefly described below:

The performance of the prediction model based on historical data will be evaluated by inspection of the mean absolute percentage error (MAPE):

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right|, \tag{8.3}$$

where $A_t$ is the actual value and $F_t$ is the forecast value. Since the MAPE can be sensitive to outliers, the symmetrical mean absolute percentage error (sMAPE) is also investigated. The sMAPE is a relative error which lies between 0% and 100%:

$$sMAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t + F_t}\right|, \tag{8.4}$$

where $A_t$ is the actual value and $F_t$ is the forecast value.

To investigate the accuracy of the historical based prediction for congestions or non-congestions, the false positives and false negatives are calculated according to the following definitions:

**Definition 11.** *False Positives are situations where a predictor predicts a congestion, while there is none.*

**Definition 12.** *False Negatives are situations where a predictor predicts no congestion, while there is one.*

The definition of congestion can be found in Chapter 7.

## 8.6 Results and interpretation

The prediction model based on historical data has been evaluated for a prediction horizon of 60 minutes and 120 minutes. It has also been investigated which distance measure, as introduced in section 8.2, is the best. Since the prediction model 'learns' the best fitting model or raw historical day pattern, it is investigated whether the algorithm should start the learning process from the beginning (00:00 am) or maybe later that day. It can be assumed that the traffic is in free flow state during the night for each day on average. This is also visible in the models in Figure 8.3. Although the driven speed is high during the early morning, there is a little difference visible in the models between 00:00 am and 04:00 am. Therefore, it has been investigated whether the model should start to 'learn' from 00:00 am or 04:00 am. Recall that the prediction model based on historical data is able to switch between the models or the raw historical day patterns at any time.

Finally, it has been investigated whether it is necessary to include both a model data base as raw database. Therefore, the prediction model is tested with only the model database, only the raw database and for the combination of both.

The results for a prediction horizon of 60 minutes are presented in Table 8.1. The top line presents the error values for the NAIVE model. The NAIVE model is already introduced in section 7 and kept the same for this experiment. The NAIVE model assumes that the traffic will not change during the prediction horizon, therefore it will be always too late in predicting the start or an end of a congestion. The table shows that the NAIVE model has a high error, which should not come as a surprise as the prediction horizon is high (60 minutes).

There is not a large difference in performance if the model starts ’learning’ at 00:00 am or at 04:00 am. It is interesting to see that there is a large difference in false positives, if only the model database is used. A speed model represents an average of the data and is therefore less likely to contain extreme low or high values of speed. A false positive is a situation where the model predicts a congestion while there is none. If there is a strong congestion and a historical model is used to predict the speed for the future, the prediction will probably contain a speed value which is to high and the congestion cannot be predicted accurately. Therefore, a decrease of false positives and an increase in false negatives if only the model database is used is understandable. If the model database and the raw database are both included, the errors are not higher compared to only using the raw historical database. This means that the prediction model based on historical data is capable of deciding whether it should take a model or raw data on its own. In general, the errors of the historical model are a little lower then from the NAIVE model, but it is only marginal and not convincing.

Table 8.1: Results for Historical prediction model with prediction horizon 60 minutes

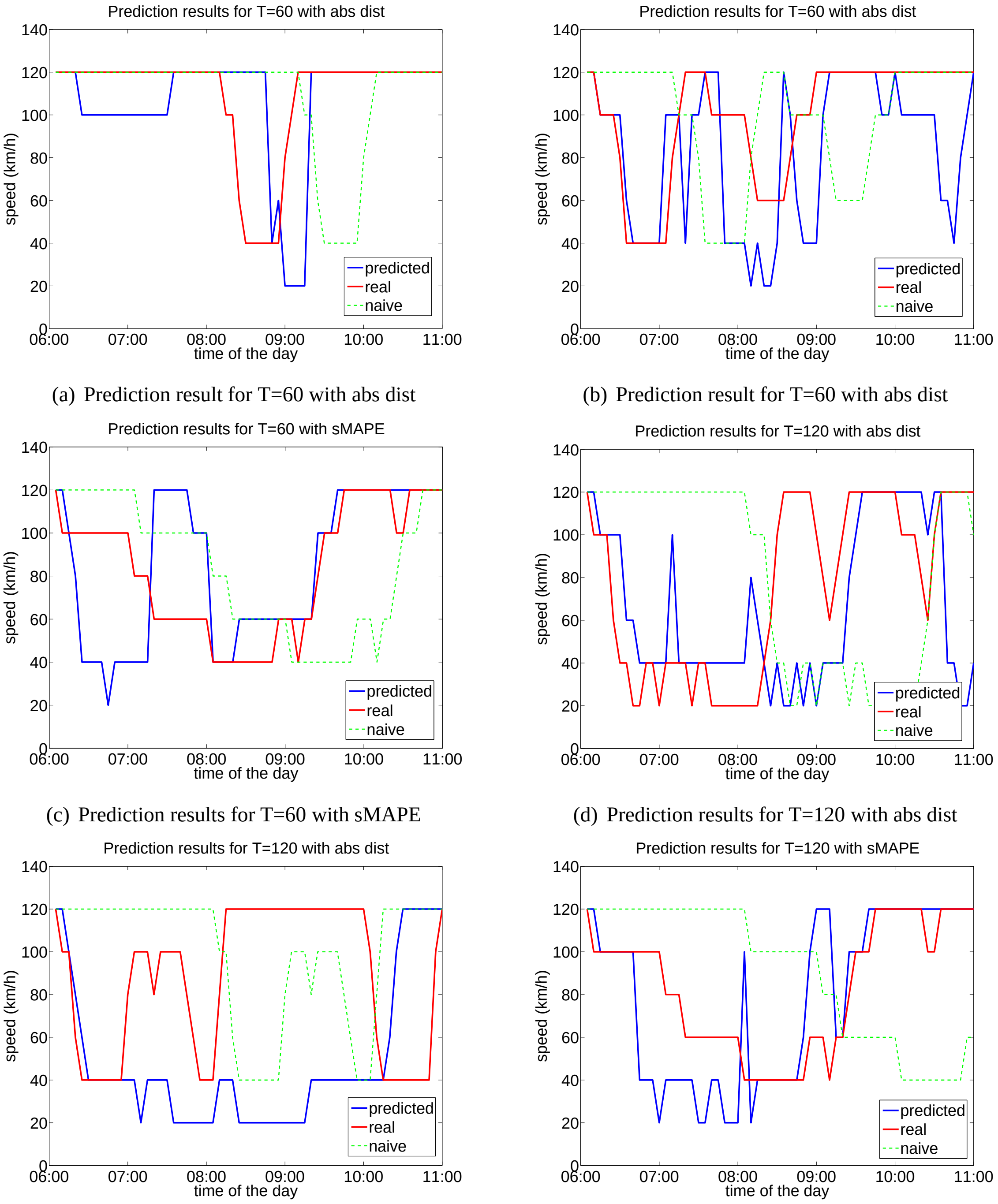
| | | | NAIVE model | 0.46 | 0.16 | 0.15 | 0.38 |
|---|---|---|---|---|---|---|---|
| Distance measure | Models | Database | start learning | MAPE | sMAPE | FP | FN |
| abs dist | yes | yes | 00:00 | 0.41 | 0.14 | 0.13 | 0.35 |
| abs dist | yes | no | 00:00 | 0.46 | 0.15 | 0.098 | 0.42 |
| abs dist | no | yes | 00:00 | 0.41 | 0.14 | 0.14 | 0.34 |
| sMAPE | yes | yes | 00:00 | 0.40 | 0.14 | 0.13 | 0.34 |
| sMAPE | yes | no | 00:00 | 0.46 | 0.15 | 0.094 | 0.43 |
| sMAPE | no | yes | 00:00 | 0.40 | 0.14 | 0.14 | 0.33 |
| abs dist | yes | yes | 04:00 | 0.38 | 0.14 | 0.16 | 0.30 |
| abs dist | yes | no | 04:00 | 0.46 | 0.15 | 0.096 | 0.43 |
| abs dist | no | yes | 04:00 | 0.39 | 0.14 | 0.17 | 0.30 |
| sMAPE | yes | yes | 04:00 | 0.39 | 0.14 | 0.16 | 0.31 |
| sMAPE | yes | no | 04:00 | 0.46 | 0.15 | 0.094 | 0.43 |
| sMAPE | no | yes | 04:00 | 0.39 | 0.14 | 0.17 | 0.30 |

Table 8.2 shows the results for a prediction horizon of 120 minutes. The results show that the historical based prediction model is able to outperform the NAIVE predictor for prediction over 120 minutes. There is still not a big difference between the learning start at 00:00 am or 04:00 am. Using the models as well as the raw data also seems to be the best option here.

Table 8.2: Results for Historical prediction model with prediction horizon 120 minutes

| | | | NAIVE model | 0.84 | 0.24 | 0.19 | 0.66 |
|---|---|---|---|---|---|---|---|
| Distance measure | Models | Database | start learning | MAPE | sMAPE | FP | FN |
| abs dist | yes | yes | 00:00 | 0.53 | 0.17 | 0.17 | 0.41 |
| abs dist | yes | no | 00:00 | 0.52 | 0.17 | 0.11 | 0.47 |
| abs dist | no | yes | 00:00 | 0.56 | 0.18 | 0.18 | 0.44 |
| sMAPE | yes | yes | 00:00 | 0.52 | 0.17 | 0.17 | 0.42 |
| sMAPE | yes | no | 00:00 | 0.52 | 0.17 | 0.17 | 0.42 |
| sMAPE | no | yes | 00:00 | 0.55 | 0.17 | 0.16 | 0.44 |
| abs dist | yes | yes | 04:00 | 0.51 | 0.17 | 0.20 | 0.41 |
| abs dist | yes | no | 04:00 | 0.55 | 0.17 | 0.12 | 0.50 |
| abs dist | no | yes | 04:00 | 0.51 | 0.18 | 0.20 | 0.40 |
| sMAPE | yes | yes | 04:00 | 0.52 | 0.18 | 0.19 | 0.42 |
| sMAPE | yes | no | 04:00 | 0.55 | 0.17 | 0.12 | 0.50 |
| sMAPE | no | yes | 04:00 | 0.51 | 0.18 | 0.20 | 0.40 |

Both tables give an overview of the performance of the historical based prediction model. It is interesting to see some specific prediction results to clarify its performance. Figure 8.4 shows 6 prediction results for different prediction horizons and different distance measures. It becomes clear that the historical model is able to predict a general trend of a congestion, but still makes errors. Further, the predictions show a stochastic behavior which could be explained by the fact that the model could be alternating between different models or raw historical data. In general, it can be stated that there is a need for some kind of smoothing process to get better results. Therefore, Chapter 9 proposes a hybrid approach.



(a) Prediction result for T=60 with abs dist

(b) Prediction result for T=60 with abs dist

(c) Prediction results for T=60 with sMAPE

(d) Prediction results for T=120 with abs dist

(e) Prediction results for T=120 with abs dist

(f) Prediction results for T=120 with sMAPE

Figure 8.4: Prediction results for T=60 and T=120 for different distance measures

# Chapter 9

# Hybrid Model

In this Chapter we propose a Hybrid approach to predict traffic data. In Chapter 7 we showed that average vehicle speed can be predicted by a Bayesian Network for short prediction horizons. In Chapter 8 we showed that average vehicle speed can be predicted with our proposed historical based prediction model for the longer prediction horizons. The results from this historical based prediction model were not convincing, so there is a need for another approach. We propose a Hybrid approach which combines the predictions done by the historical based predictor with the Bayesian Network model. The historical based prediction model yields a set of initial estimators which can be set in the Bayesian Network. The Bayesian Network is used to ’smooth’ these initial estimators and to predict average vehicle speed.

Section 9.1 explains our initial thought about a combination of a historical based predictor with a Bayesian Model. The Hybrid model, which we propose, will be presented in section 9.2. The Hybrid model is tested in an experiment, which is described in section 9.3. The performance measurements used for evaluation purposes are described in section 9.4. The results and interpretations of our model are presented in section 9.5.

## 9.1 Combining Bayesian models and historical data

From Chapter 7 it became clear that modeling traffic with Bayesian Networks gives promising results. The proposed network is robust and is able to predict a traffic congestion at the A4 at hectometer location 23.5 with accuracy ranges of 95-65% corresponding to prediction horizons from 5 minutes to 2 hours in morning rush hours respectively. Of course, accuracy rates of 65% are not desirable. It should not come as a surprise that predicting traffic becomes harder as the prediction horizon increases. During a longer prediction horizon, traffic situations can easily change due to the number of travelers, incidents, etc. Therefore, there is a need for a method which can handle longer horizons.

Chapter 6 investigates speed graphs in a large historical database. This investigation showed that speed graphs are similar (homogeneous) during weekends, during Fridays and from Monday-Thursday. There are also similarities in holidays and public holidays, but these are less visible. A database was formed in Chapter 8 with mean models, which represent the mean of homogeneous sets. The same Chapter proposes a historical based prediction model which finds the best fitting model and uses this as an prediction. If a best fitting model cannot be found, the algorithm searches through the raw historical data base with day speed graphs to find a corresponding day in history. The model is able to switch between the models or the raw historical day patterns at any time, which makes it a continuously adaptive model. Chapter 6 shows that the prediction accuracy of a Bayesian Network model for a prediction horizon of 1 or 2 hours is higher then for the NAIVE prediction model, but still not convincing. Therefore, there is a need for another method.

This Chapter proposes a Hybrid model which uses Bayesian Networks in combination with a historical based predictor. Although it is known that the historical based prediction model did not yield convincingly better prediction results compared to the Bayesian Network predictor, it is not investigated how a combination of both performs. The Bayesian Network Model, as described in Chapter 7, incorporates the prediction location and 2 other locations. A historical database can be created with models and raw data for every location. In total, this will result in 3 model databases and 3 raw databases. Each database contains models for *speed* and for *density*.

The historical model can be used to predict the speed and density for, lets say, the next hour at all three locations according to the method as is described in Chapter 8. These predictions for speed and density for three locations can be given to a trained Bayesian Network and set as its current beliefs. The Bayesian Network can be trained to predict for different horizons, lets say for 10 minutes. The prediction can be obtained by updating the beliefs and a prediction for the over 70 minutes in total is given. The first 60 minutes are predicted with the historical predictor, and the last 10 minutes with the Bayesian Network. The Bayesian Network is actually smoothing the predictions of the historical predictor in the learned Bayesian Network. Of course, this Hybrid model can be set up in different ways, which will be described in the next section.

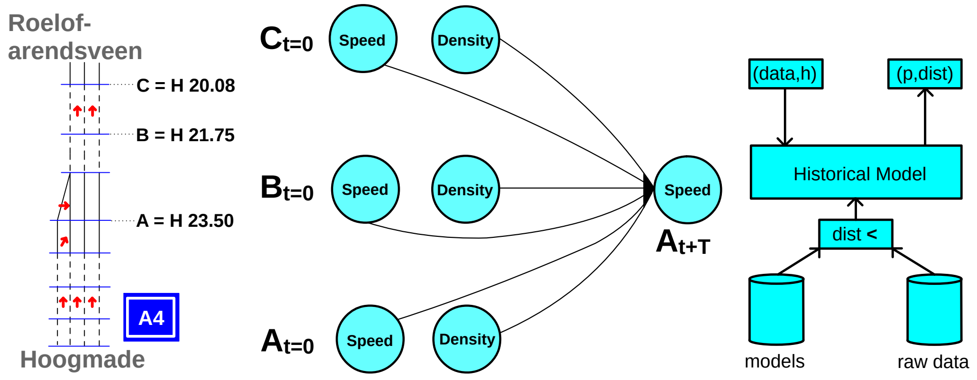
## 9.2 Hybrid models

The Hybrid model consist of a Bayesian Network and a historical model. From Chapter 7 it became clear that Bayesian Network model 4 clearly outperforms the other models. Therefore it has been decided to choose this Bayesian Network model. This model incorporates the speed and density for the prediction location and 2 locations further downstream.

Figure 9.1 presents the components of the Hybrid model. Figure 9.1(a) shows the locations on the road for the nodes in the Bayesian Network and Figure 9.1(b) shows the Bayesian Network itself. Figure 9.1(c) shows the historical model, as explained in Chapter 8.

In Chapter 8 it is described that the historical model can include a model database, historical raw data or even both at the same time. It has been investigated which of these options performs the best. This experiment showed that it is better to include both the model database as well as the raw data, since this yields the lowest false positives error. The historical prediction model could start the learning for a day at 00:00 am or for example at 04:00 am. The experiment showed us that there does not seem to be a significant difference between the performance for these options. It has been decided to start the learning at 00:00 am. The distance measure for the historical model, the sMAPE or the absolute total distance, seem to have no influence on the performance. But to be sure, it has been decided to test the Hybrid model for both the absolute total distance and the sMAPE.

In summary, it was decided to take Bayesian Network model 4 and the historical model including both the model database and the raw data for the absolute total distance measure and the sMAPE. An explanation of the experiment is given in section 9.3.



(a) Schematic overview for node locations A,B and C for Model 4

(b) Bayesian Model 4

(c) Historical Model

Figure 9.1: Hybrid model components

## 9.3 Experimental setup

A traffic dataset for the A4 left side for hectometer locations 20.08, 21.75 and 23.5 in the year 2009 has been extracted for this experiment. This dataset consists of speed and density measurements for 365 days and the Treiber and Helbing [38] filter is applied to fill in the missing values. Details of this Treiber and Helbing filter can be found in Chapter 4.

For evaluation purposes, the dataset is separated into a train and test set in a ratio of 80% / 20% respectively which corresponds with a train dataset of 292 days and a test dataset of 73 days. The days are randomly selected for the separation and both sets are mutually exclusive.

A model database has been developed for the following mutually exclusive clusters:

- Monday-Thursday
- Friday
- weekend
- holiday or public holiday.

The models are the means of the corresponding homogeneous sets, as explained in Chapter 8. The models are visualized in Figure 8.3. For a fair comparison of the performances, the models are developed from the train dataset and the test dataset is completely excluded from this.

Further, a raw database is developed. This database consists of the raw day patterns which is the union of all the data in the previously proposed clusters. Again, this raw database consists of only train data, from which the test data is hold out.

For this experiment, the Hybrid model is tested for different prediction horizons. Figure 9.2 illustrates how these horizons are set up. The current speed and density measurements for all three locations is collected from $t_0$ until $t_i$. For every prediction, the Hybrid model lets the historical based predictor compute initial predictions for speed and density for all three locations at $t_i + h$, where $h$ is

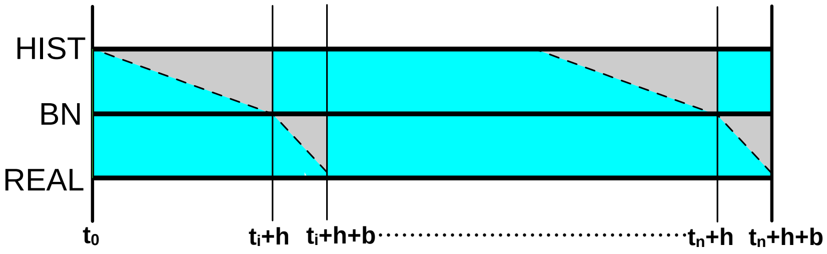


Figure 9.2: Hybrid prediction schematic overview

the prediction horizon for the historical model. After this calculation, the initial predictions are set as evidences in the Bayesian Network which is trained for a prediction horizon of $b$. After updating the beliefs in the network, the Bayesian Network yields a prediction for a prediction horizon of $b$, which is in total at time: $t_i + h + b$. The horizons $h$ and $b$ are varied to evaluate different combinations. To evaluate the performance of the Hybrid model, the performance measures are described in section 9.4.

## 9.4 Performance measures

The performance measures which are used to evaluate this experiment are the same as the one used to evaluate the Bayesian Network models in Chapter 7 and the historical based prediction models 8. For readability purposes, the performance measures are briefly described below:

The performance of the historical based prediction model will be evaluated by inspection of the mean absolute percentage error (MAPE):

$$MAPE = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{A_t - F_t}{A_t} \right|, \tag{9.1}$$

where $A_t$ is the actual value and $F_t$ is the forecast value. Since the MAPE can be sensitive to outliers, the symmetrical mean absolute percentage error (sMAPE) is also investigated. The sMAPE is a relative error which lies between 0% and 100%:

$$sMAPE = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{A_t - F_t}{A_t + F_t} \right|, \tag{9.2}$$

where $A_t$ is the actual value and $F_t$ is the forecast value.

To investigate the accuracy of the historical based prediction for congestions or non-congestions, the false positives and false negatives are calculated according to the following definitions:

**Definition 13.** *False Positives are situations where a predictor predicts a congestion, while there is none.*

**Definition 14.** *False Negatives are situations where a predictor predicts no congestion, while there is one.*

The definition of congestion is given in Chapter 7.

## 9.5 Results and interpretation

In total 36 tests have been conducted to evaluate the performance of the Hybrid model for different horizons. The tests are grouped into 4 subsets:

1. BN Model 4 and Historial Model with total absolute distance measure for prediction horizon from 65-90 minutes.
2. BN Model 4 and Historial Model with sMAPE distance measure for prediction horizon from 65-90 minutes.
3. BN Model 4 and Historial Model with total absolute distance measure for prediction horizon from 125-150 minutes.
4. BN Model 4 and Historial Model with sMAPE distance measure for prediction horizon from 125-150 minutes.

Table 9.1 presents the results for a prediction horizon in the range 65-90 minutes with a historical model which takes the total absolute distance measure to find similar models in the database. The H denotes the Hybrid model and the N denotes the NAIVE model. As becomes clear from this table, the Hybrid model clearly outperforms the NAIVE model with respect to its false negative error. It does not seem to matter what prediction horizon the Bayesian Network takes, the differences in errors are marginal. Although the table shows that the MAPE slightly increases when the prediction horizon increases. The error of the NAIVE model increases significantly when the prediction horizon increases, so that the Hybrid model becomes even better then the NAIVE model for further prediction horizons. Table 9.2 presents similar results which could indicate that there is no difference if the historical based prediction model uses the total absolute distance measure of the sMAPE distance measure to select its models. This corresponds with the findings in Chapter 8 where a convincing difference in performance by taking another distance measures could not be found.

Table 9.1: Results Hybrid Model with BN model 4 and Historical model with total absolute distance measure with prediction horizon from 60 minutes

| P | MAPE H | MAPE N | sMAPE H | sMAPE N | FP H | FP N | FN H | FN N |
|---|---|---|---|---|---|---|---|---|
| 65 | 0.31 | 0.50 | 0.16 | 0.17 | 0.19 | 0.16 | 0.14 | 0.40 |
| 70 | 0.30 | 0.53 | 0.16 | 0.18 | 0.21 | 0.17 | 0.12 | 0.42 |
| 75 | 0.34 | 0.56 | 0.17 | 0.19 | 0.22 | 0.18 | 0.13 | 0.44 |
| 80 | 0.35 | 0.59 | 0.17 | 0.20 | 0.24 | 0.19 | 0.12 | 0.47 |
| 85 | 0.36 | 0.62 | 0.17 | 0.21 | 0.24 | 0.20 | 0.14 | 0.48 |
| 90 | 0.37 | 0.65 | 0.18 | 0.21 | 0.24 | 0.21 | 0.15 | 0.50 |

The errors for the NAIVE model grow if the prediction horizon lies in the range 125-150 minutes as can be seen in Tables 9.3 and 9.4. Again there is no convincing difference in performance for the Hybrid model between the different distance measures the historical model could take to select its models. Interesting to see is that the performance of the Hybrid model not extremely lower in the range 125-150 minutes compared to the range 65-90 minutes. This could mean that the Bayesian

Table 9.2: Results Hybrid Model with BN model 4 and Historical model with sMAPE with prediction horizon from 60 minutes

| P | MAPE H | MAPE N | sMAPE H | sMAPE N | FP H | FP N | FN H | FN N |
|---|---|---|---|---|---|---|---|---|
| 65 | 0.31 | 0.50 | 0.16 | 0.17 | 0.22 | 0.16 | 0.15 | 0.40 |
| 70 | 0.31 | 0.53 | 0.16 | 0.18 | 0.23 | 0.17 | 0.14 | 0.42 |
| 75 | 0.35 | 0.56 | 0.17 | 0.19 | 0.24 | 0.18 | 0.14 | 0.44 |
| 80 | 0.35 | 0.59 | 0.17 | 0.20 | 0.25 | 0.19 | 0.12 | 0.47 |
| 85 | 0.36 | 0.62 | 0.18 | 0.21 | 0.25 | 0.20 | 0.14 | 0.48 |
| 90 | 0.38 | 0.65 | 0.18 | 0.21 | 0.26 | 0.21 | 0.14 | 0.50 |

model is capable of smoothing the initial estimators of the models correctly into a meaningful traffic pattern. The Hybrid model predicts a speed graph for over 2.5 hours with a false negative error of 15%. This means that 85 out of 100 traffic congestions are correctly predicted by the Hybrid model, which is promising.

Table 9.3: Results Hybrid Model with BN model 4 and Historical model with total absolute distance measure with prediction horizon from 120 minutes

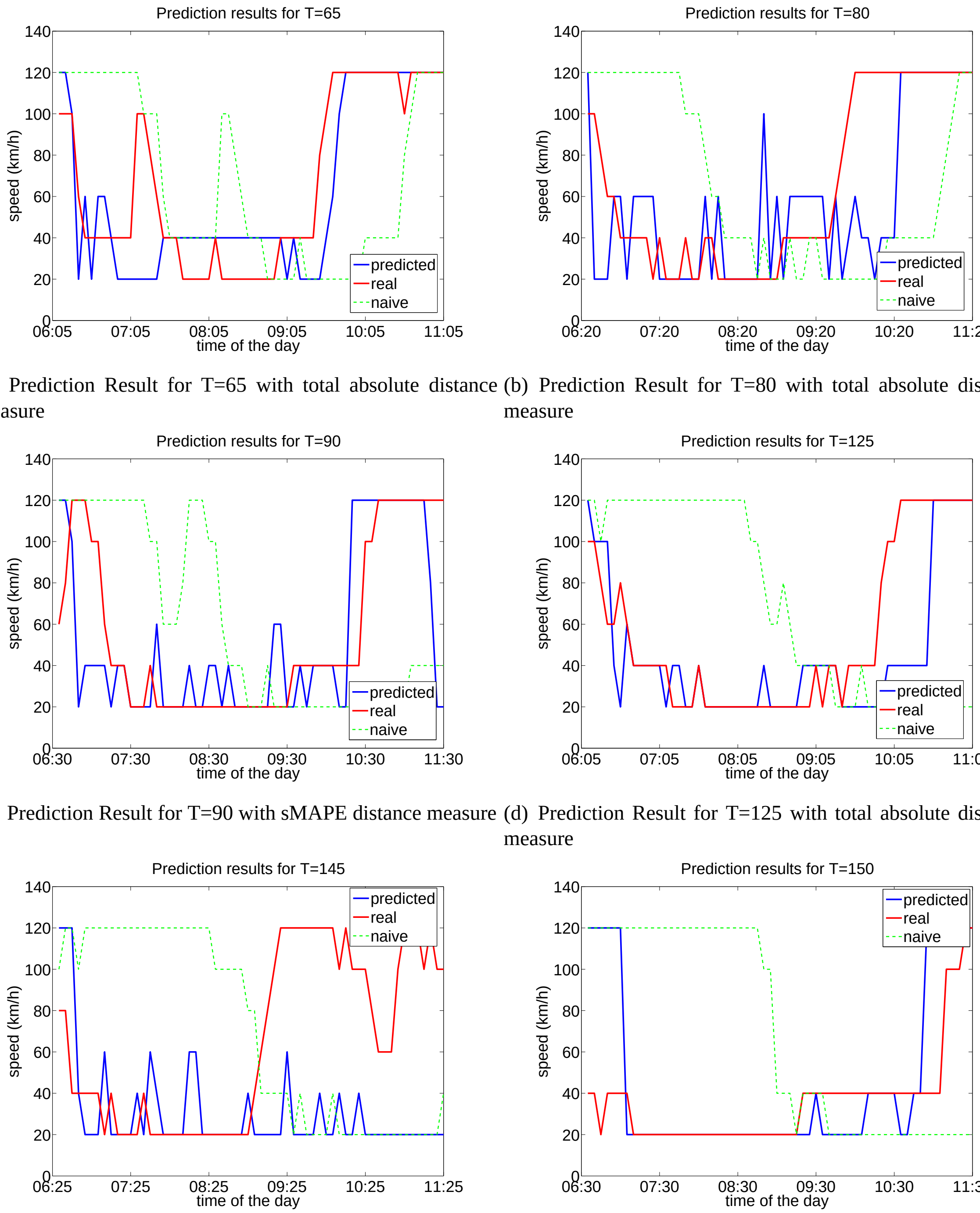
| P | MAPE H | MAPE N | sMAPE H | sMAPE N | FP H | FP N | FN H | FN N |
|---|---|---|---|---|---|---|---|---|
| 125 | 0.35 | 0.87 | 0.19 | 0.25 | 0.25 | 0.20 | 0.17 | 0.68 |
| 130 | 0.34 | 0.90 | 0.19 | 0.25 | 0.27 | 0.21 | 0.15 | 0.70 |
| 135 | 0.36 | 0.93 | 0.19 | 0.26 | 0.28 | 0.22 | 0.14 | 0.72 |
| 140 | 0.36 | 0.95 | 0.19 | 0.26 | 0.29 | 0.23 | 0.12 | 0.74 |
| 145 | 0.36 | 0.97 | 0.19 | 0.27 | 0.29 | 0.24 | 0.14 | 0.76 |
| 150 | 0.38 | 0.99 | 0.20 | 0.27 | 0.29 | 0.25 | 0.15 | 0.77 |

Table 9.4: Results Hybrid Model with BN model 4 and Historical model with sMAPE with prediction horizon from 120 minutes

| P | MAPE H | MAPE N | sMAPE H | sMAPE N | FP H | FP N | FN H | FN N |
|---|---|---|---|---|---|---|---|---|
| 125 | 0.37 | 0.87 | 0.20 | 0.25 | 0.28 | 0.20 | 0.18 | 0.68 |
| 130 | 0.36 | 0.90 | 0.20 | 0.25 | 0.30 | 0.21 | 0.16 | 0.70 |
| 135 | 0.38 | 0.93 | 0.20 | 0.26 | 0.31 | 0.72 | 0.15 | 0.72 |
| 140 | 0.38 | 0.93 | 0.21 | 0.26 | 0.31 | 0.22 | 0.13 | 0.72 |
| 145 | 0.39 | 0.97 | 0.21 | 0.27 | 0.32 | 0.24 | 0.14 | 0.76 |
| 150 | 0.42 | 0.99 | 0.21 | 0.27 | 0.32 | 0.25 | 0.15 | 0.77 |

Figure 9.3 shows some prediction graphs for the Hybrid models. Although it would be confusing to visuale all 73 test predictions, this figure shows a comprehensible overview for different predictions under different settings. The prediction horizons range from 65-150 minutes and different distance measures for the historical model.

It becomes clear that the Hybrid model is quite powerful in predicting speed graphs for longer prediction horizons. Further, the figure shows us that the Hybrid model predicts a stochastic-like behavior by continuously jumping between 2 speed classes. This could be due to the fact that the discretization into 6 classes might not be the best option, or that the historical based predictor is continuously changing its best fitting model since there are more similar models to choose from.



(a) Prediction Result for T=65 with total absolute distance measure

(b) Prediction Result for T=80 with total absolute distance measure

(c) Prediction Result for T=90 with sMAPE distance measure

(d) Prediction Result for T=125 with total absolute distance measure

(e) Prediction Result for T=145 with total absolute distance measure

(f) Prediction Result for T=150 with sMAPE distance measure

Figure 9.3: Prediction results for T=(65, 80, 90, 125, 145, 150)

# Chapter 10

# Software Models and Implementation

This chapter describes the software models, the implementation and used third party software of the research workbench. This workbench includes a collection of functions to select, prepare and analyze traffic data. Our data selection and preparation functions allow us to analyze and visualize the enormous amount of data in a matter of seconds. The workbench is also capable of running and testing traffic prediction models. For traffic researchers, this workbench could be used for implementing and testing their models. The workbench is set up generic, so that extending this research tool is possible.

First we describe our choice for the Bayesian Software which we used for this thesis in section 10.1. Section 10.2 gives an overview of the workbench system and its components. The data selection and preparation functionality, which is developed in Mathworks MATLAB, is presented in section 10.3. The research workbench is controlled from within Java in which the application is programmed. A detailed description of the class diagram and a sequence diagram are presented in section 10.4.

## 10.1 Bayesian Networks software

There are numerous software tools available in the research field of Bayesian Networks. A convenient overview of existing Bayesian Network software can be found in [17]. Since this thesis is done in a research project, only free or open source software has been taken into account for our decision.

We decided to use GeNIe/SMILE [1] engine which is developed at the Decision Systems Laboratory at the University of Pittsburgh. GeNIe (**G**raphical **N**etwork **I**nterface) is a development environment for building decision networks under Windows. SMILE (**S**tructural **M**odeling, **I**nference, and **L**earning **E**ngine) is its portable inference engine, consisting of a library of C++ classes [17].

Flexibility is important in our explorative research, since system design choices are bound to change during the process. Therefore, we decided to develop our workbench as generic as possible.

A convenient wrapper has been written to use SMILE from within Java, which eases the connection to other implementation languages. The GeNIe/SMILE engine is an adaptive system, since models, developed in GeNIe/SMILE, can be used in different software packages such as Hugin, Netica or Ergo, which gives us flexibility.

The fact that Bayesian Networks can be developed graphically in the GeNIe interface gives us the opportunity to rapid prototype Bayesian Networks, which eases the development process.

[1] More details can be found on http://www.sis.pitt.edu/˜genie/

## 10.2 System design

For this thesis, we developed a research workbench which could be seen as a system. This system is used for the experiments within this thesis. This research workbench has the possibility to investigate traffic data and different traffic prediction models.

Since many software libraries, packages and tools for data analysis already exist, our system should not be a reinvention of them. Different existing software libraries and tools are combined in this workbench and new software has been written to connect them to traffic related data and algorithms. There are numerous programming environments available, which all have their own advantages and disadvantages. The research workbench we developed for this thesis combines three different programming environments:

1. Oracle's Java
2. The Mathworks MATLAB
3. Rapid-I's RapidMiner.

Since is was decided to use the SMILE library of the University of Pittsburgh for the development of Bayesian Networks, the programming environment Java was used to develop, train and test Bayesian Networks. The SMILE library encloses a convenient wrapper written for Java which connects to the underlying and optimized SMILE C++ code in which Bayesian Networks can be programmed efficiently. Java is an object oriented programming language, which makes it possible for us to develop an object oriented structure which eases the coding process.

The Mathworks' MATLAB is a numerical programming environment which is optimized for matrix manipulations, which is of course convenient when working with large datasets. The Mathworks Builder for Java compiles MATLAB code into a Java library, so that MATLAB's convenient matrix environment can be used from within Java. The combination of both languages in connection to the SMILE Library of the University of Pittsburgh gives us the possibility to handle large datasets and to develop Bayesian Networks.

The RapidMiner environment is currently not actively connected to the research workbench. Details of how to connect RapidMiner to other Java applications are described in [15]. Currently, RapidMiner is only used for data exploration purposes but it can easily be connected to the research workbench in the future. Therefore, it is already encapsulated in the design process.

A system overview of the research workbench is given in figure 10.1. This overview shows the combination of the three programming environments and the MATLAB builder for Java which is enclosed in the wrapper environment.

Generally, developing functions in Mathworks MATLAB is done in a script language. This makes it hard to define a structure. New versions of MATLAB give the possibility to define an object oriented structure, but we decided to program an object oriented application and control in Java.

## 10.3 Matlab environment

The MATLAB environment is used to collect (retrieve) traffic measurements in (from) the MONICA (traffic database). These measurements are delivered in ASCII format by the Dutch Ministry of Transport, Public Works and Water Management.

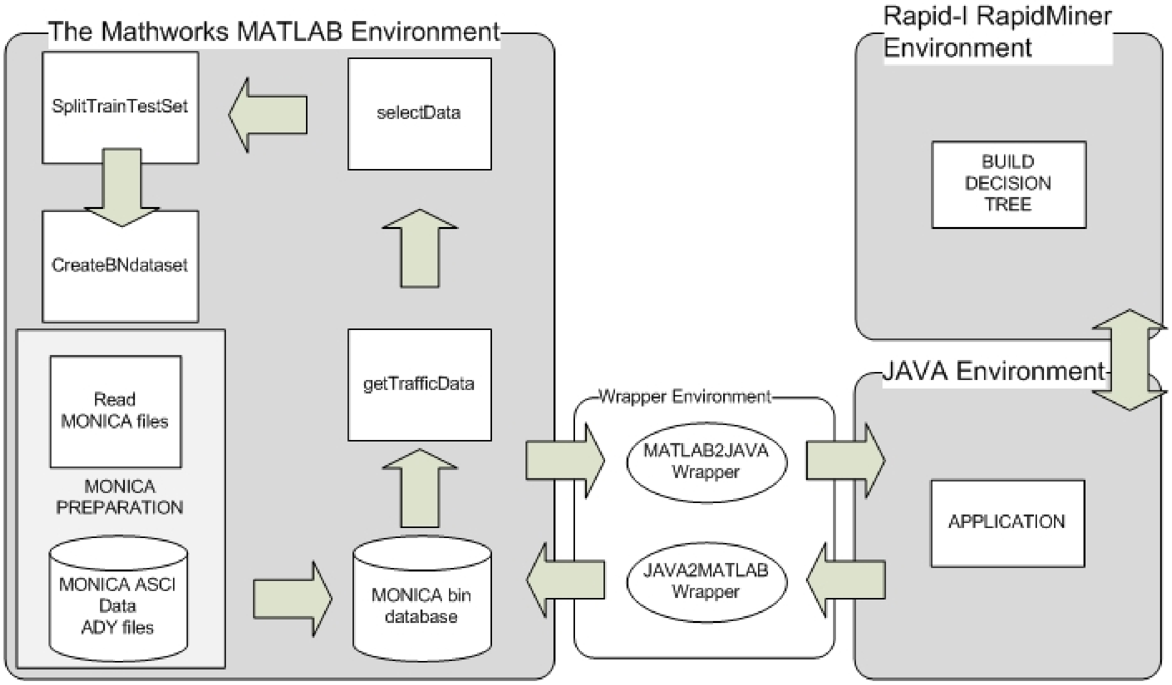


Figure 10.1: Research workbench system overview

As explained in chapter 4, there are around 21.354 ILDs in the Netherlands in the year 2009. This number can deviate from the real number since some ILDs are double counted by the MONICA System.

Every ILD measures variables such as average vehicle speed and average vehicle intensity for a particular ILD location per minute. The ILDs are divided into five regions. The MONICA system collects the measurements from each ILD every minute and creates a raw text file (ASCII) for every region. One days contains 1440 minutes. Simple mathematics show that the MONICA database for 2009 consists of $5 * 1440 * 365 = 2.628.000$ text files, since there are 5 regions, 1440 minutes in a day and 365 days in year. Each text file contains a line for every ILD for a region with its ID, which is called a BPS-code and a line with measurements. Therefore these text files contain thousands of lines. This is an enormous amount of data!

It should not come as a surprise that there are complications with this amount of data. Therefore, an intelligent approach is needed which is able to work efficiently with this data.

The first step of this intelligent approach is to convert these raw text files into a structured format and to save it in binary format for memory purposes. We ordered the measurements, which we extract from the raw text files, in time and group them per ILD for retrieval purposes. By a special coding, developed by TNO, the index for each ILD in the structured data can be calculated. In this way, the ILD of interest can be selected efficiently.

The data selection, and preparation process is a subprocess of the MATLAB environment, as visualized in figure 10.1. The information of a region for one day is structured into a MATLAB matrix. This matrix has size $N$ by $M$, where $N$ is the number of ILDs and $M$ is number of minutes for one day (1440). This process has to be done only once, but for every day which results in a database of 5 regions, each containing 365 matrix files.

The second step is to load to correct matrix in which the ILD of interest is located. Selecting the data of interest is not a trivial process. We programmed several data selection methods, to filter the data for the correct ILDs, dates and times. After selecting the ILD, the data can be extracted from a matrix by defining a time interval of interest. Since an ILD could be in any of the 5 region's, an intelligent approach is taken to find out which region contains the ILD of interest. This is done by creating Java HashMaps in MATLAB for every region, where the unique ID of the ILD's are saved as key and their index as value. In milliseconds it can be calculated in which region the ILD is located and its corresponding matrix can be loaded.

Furthermore, we developed convenient functions for data selection and visualization. We developed a system in which a full year of MONICA data for a number of locations of interest can be loaded. For this, we developed functionalities to visualize central tendencies and variances of the dataset for research purposes.

As becomes clear from figure 10.1, there are MATLAB functions programmed to select traffic data from the database and to create datasets for the Bayesian Network code in the Java environment. The wrapper environment adapts both environments for the passage of data and function calls.

## 10.4 Java environment

The research workbench includes a Java environment in which the control of the application is situated. The structure of the Java code is graphically visualized in a class diagram in figure 10.2. This class diagram only contains high level descriptions, since a deeper level of detail would only distract the reader from the key issues.

Since the Java application must be able to handle different traffic prediction models which have underlying similarities, a design pattern will help the development process. The *General Hierarchy pattern* [18] is chosen for this application. This pattern allows for each object to have zero or more objects above them in the hierarchy (superiors) and zero or more objects below them (subordinates) [18].

As becomes clear from the class diagram, the application contains a controller. This controller 'controls' the application in the sense that it can create models, connects to the traffic data in MATLAB, or it can create a trafficDataSettings object. The Model class contains the basic functionality which is the same for every subordinate class which inherits the Model class. A model should be able to set selection criteria for traffic data, import data and to evaluate its performance. The specific structure for each model is different. Therefore, a separation is done for Bayesian Models, Historical Models and Hybrid Models and each type of model has its own characteristics. Although a hierarchical relationship between the HybridModel and the HybridModel1 is not necessary, it is implemented for extension purposes in a later stage of this research. The same holds for the hierarchical relationship between the HistoricalModel and the HistoricalModel1 class.

The collaboration between the MATLAB en Java environment is not straightforward. For clarity purposes, a sequence diagram for creating, running and testing a Bayesian Network Model is enclosed in figure 10.3. This sequence diagram omits the details of the parameters in the functions for readability purposes. As becomes clear from this sequence diagram, the Controller class initiates

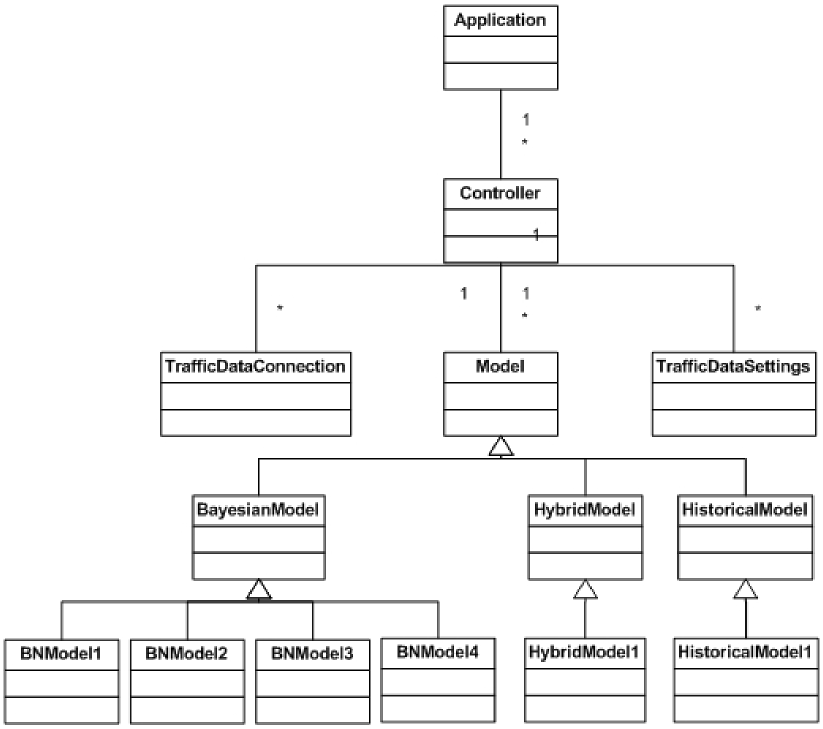


Figure 10.2: Java class diagram

almost all processes of the Bayesian Model. The Bayesian Model class strongly collaborates with the TrafficDataConnection class, which is the data portal to the the MATLAB Traffic database. The TrafficDataSettings object contains all settings which are necessary for the Bayesian Network, for example the discretization settings. The TrafficDataConnection class is programmed generic, so that these settings can be passed through MATLAB to get the data of interest with the correct discretization settings. The controller could create more TrafficDataSettings for different models and even more TrafficDataConnections.

To present all possible sequence diagrams would be too much for this thesis. This sequence diagram represents the structure in which the system is developed. In general, all functionality programmed in MATLAB is accessible from Java by the TrafficDataConnection class. The system is programmed generic, so that it can easily be extended for new ideas or models.

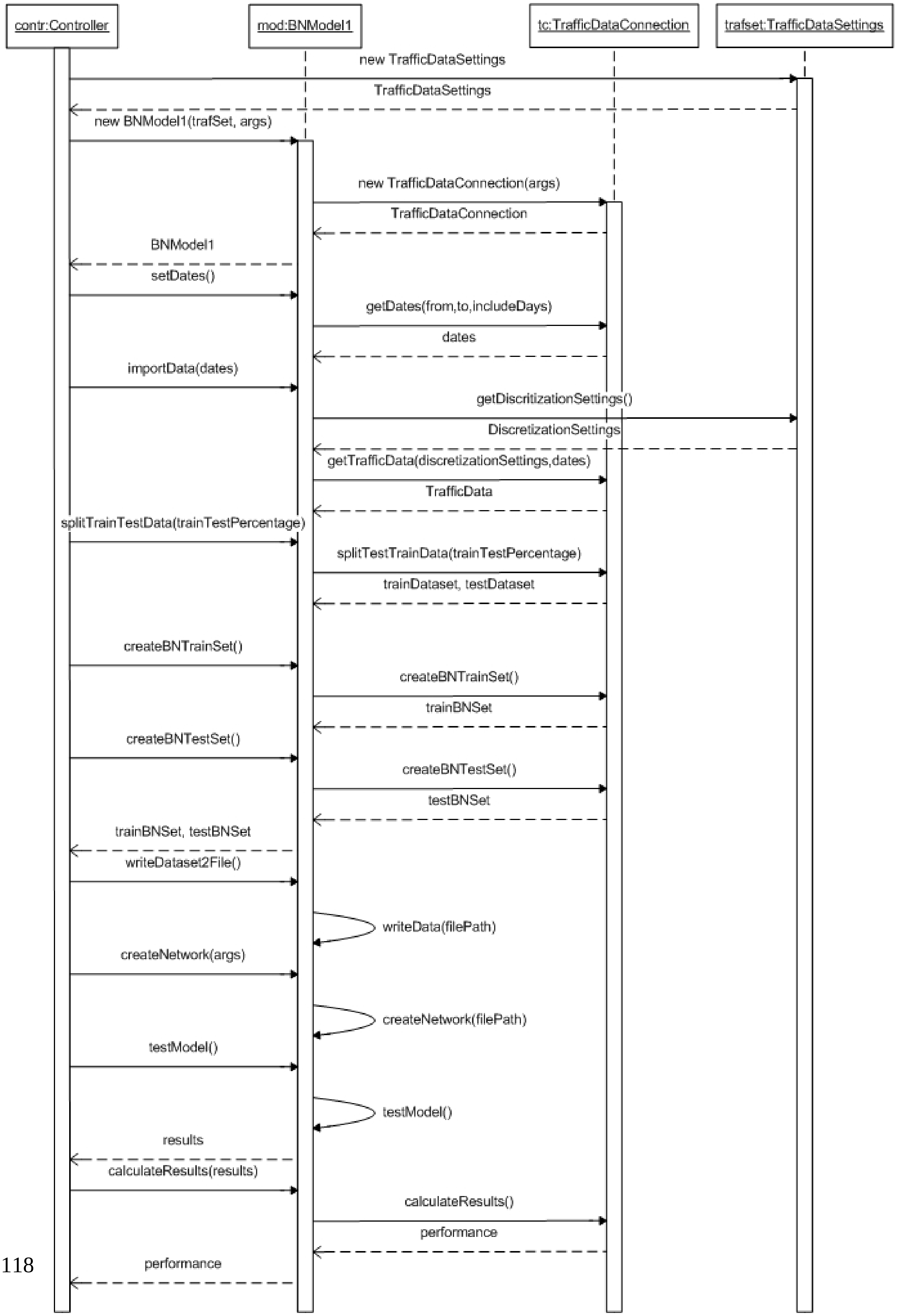


Figure 10.3: Sequence diagram for testing a Bayesian model

# Part IV

# Results and Final Remarks

# Chapter 11

# Conclusions and Discussion

After having completed the research as proposed in the introduction in Chapter 1, there are several conclusions which could be drawn. These conclusions are structured in the following parts:

1. A *theoretical part* which concludes (1) the investigation of the current literature on Traffic Modeling and (2) the investigation of the theoretical background of Bayesian Networks.
2. A *data exploration* part which concludes (3) the data acquisition and preparation, (4) the data sensibility analysis and (5) the data streaming analysis.
3. A *model and implementation part* which concludes (6) modeling traffic by Bayesian Networks, (7) modeling traffic by historical based models, (8) the combination of historical modeling with Bayesian Networks and (9) the description of the developed software for this thesis.

A detailed description and explanation of the conclusions for every part will be given in the remaining of this section.

## 11.1 Theoretical part

**1. The investigation of the current literature on Traffic Modeling:** From literature it became clear that *trajectories*, *speeds*, *intensities* and *densities* are the most important variables in traffic research for freeways [11], [41]. These measurements are often used to calculate travel times, which is an important measure of the performance of a road network.

Most of the traffic data currently used is acquired by Inductive Loop Detectors (ILDs) [39], which measure speed and intensity on evenly and widely spread locations on the Dutch freeways. Since speed, intensity and density are related, only two of them need to be known to calculate the other.

The relation between speed, intensity and density becomes visible in a fundamental diagram, for which the relation between intensity and density is important for the detection of traffic congestions [11].

An important conclusion about ILDs is that dual ILDs are prefered over single ILDs, as dual ILDs are capable of measuring average vehicle speed directly whereas single ILDs are only capable of estimating average vehicle speed.

Further, from this Chapter it can be concluded that there are three main approaches if it comes to traffic modeling and prediction:

1. *Instantaneous approach*: From literature [41] it became clear that the instantaneous approach is only useful if the traffic situation remains stationary for the time the prediction holds. The Piece-wise Constant Speed Based (PCSB) method is currently the state-of-the-art travel time predictor which can only be used for current traffic situations or predictions with an extreme short time horizon because the assumption is that the traffic remains stationary. It is, however, a fact that traffic usually does not remain stationary [20]. Therefore, the instantaneous approach is usually only taken for off-line computations [19].

2. *Model approach*: The model approach is often taken by Civil Engineers, since it implies extensive knowledge about the traffic domain [41]. The model approach assumes that the behaviour of each variable of a traffic model is a function of traffic conditions and its environment [12]. It can be concluded that the model approach is only suitable for off-line computations as well, since it implies computational complexity [12] because there is usually a large collection of variables involved.

3. *Data driven approach*: The data driven approach for traffic prediction systems work with inductive techniques. Data driven methods have a top-down approach in which it is tried to correlate observed traffic measurements to current and past traffic data to predict traffic [41]. There are numerous data driven models, which all have their own advantages and disadvantages. A promising and rather new method in the field of traffic predictions is the use of Bayesian Networks. A few applications for predicting traffic with Bayesian Networks already exist [34], [35], [33], [46] and [25].

**2. The investigation of the theoretical background of Bayesian Networks:** Bayesian Networks provide a way to represent and reason about an uncertain domain [17] like the traffic domain. A standard notation is formalized which is used in most literature about Bayesian Networks [10].

There are different methods to train or learn Bayesian Networks, but it can be concluded that the Expectation Maximization (EM) algorithm is the most used training algorithm for Bayesian Networks which contain missing values for certain combinations of states in the conditional probability tables. [10].

## 11.2 Data exploration part

**3. The data acquisition and preparation:** The Dutch Monitoring Casco (MONICA) system which acquires traffic data from dual Inductive Loop Detectors (ILDs) contain on average 12% missing data due to maintenance, incidents, accidents, or power or communication failures [41]. There are several ways to fill in these missing values. We can conclude that replacing missing values by applying a Treiber and Helbing [38] filter is the best method currently to fill in missing values because it takes the spatio tempораral information into account.

Further it can be concluded that the only correct way to average vehicle speed is to use the *harmonic* mean [41].

To explore traffic data, we developed a system in which researchers can select traffic data of interest of a certain ILD, day and time in only a matter of seconds. A huge database, containing more then 2 million text files, has to be searched for this data of interest. These 2 million files belong to only one year of traffic data. Searching in this data is certainly not trivial. Instead of using database techniques, we developed our own structure which is able to select the data of interest in an intelligent way.

**4. The data sensibility analysis:** From this study, it can be concluded that most of the traffic congestions occur in the Randstad, an agglomeration of the four largest Dutch cities: Amsterdam, Rotterdam, The Hague and Utrecht in which around half of the Dutch population lives.

Further it can be concluded that traffic measurements in subsequent locations on Dutch freeways are likely to be correlated, but this correlation decreases if the distance between these locations increases.

We developed several functions to visualize traffic distribution plots and congestion graphs. Because of our intelligent approach for data selection and preparation, we are able to use MONICA data in its full richness.

This sensibility study investigated speed graphs of several morning rush hours on the A4 motorway in the Netherlands in 2009. It can be concluded that these speed graphs seem to follow certain patterns for certain days, but these patterns are dependent on their ILD location.

Several histograms for the distribution of speed during morning rush hours at this A4 have been inspected, to see what the probability of an average vehicle speed is for a certain day. It can be concluded that Wednesdays seem to have stronger congestions than Fridays for example for the A4. Since face validity on these speed graphs is a tedious job, the speed distributions have been compared by Kullback Leibler divergences. This experiment showed that it the probability of a traffic congestion strongly depends on the location and on the day of the week.

**5. The data streaming analysis:** Inspired by the results of the sensibility study, it was investigated whether data streaming has any effect on the homogeneity of the data. It was expected that the total traffic dataset for speed, intensity and density measurements contains a huge variance.

Our developed system for selecting and analyzing traffic data gave us the possibility to visualize different traffic patterns. By selecting subgroups for several structural variables such as: day of the week, holiday, rain etc it can be concluded that these subgroups became homogeneous. The day of the week seem to give a homogeneous basic pattern for every subgroup.

By splitting the data into subsets for more detailed variables such as extreme weather conditions, homogeneity could not be found. This could be due to the fact that the data becomes more sparse since there are not much data points which contain extreme weather and a single out lier disturbs the subgroup.

Further it can be concluded that there are typically 4 homogeneous subgroups to be found in traffic data:

1. Mondays-Thursdays
2. Fridays
3. weekends
4. holidays or public holidays.

In the end of this analysis it can be concluded that these homogeneous subgroups can be found automatically by creating decision trees with Quinlan's C4.5 algorithm [28], [29]. We combined MONICA data with a data analysis software called RapidMiner to build these decision trees which is unique in the Netherlands.

## 11.3 Model and implementation part

**6. Modeling traffic by Bayesian Networks:** It can be concluded that the combination of speed and density measurements can announce possible traffic congestions. If roads become busier, the average vehicle speed drops down and the density goes up. This could be an indication for a possible traffic congestion. By making use of these indications, we can develop Bayesian Network models which are trained on examples of these indications.

For this thesis, we applied Bayesian Networks models on Dutch traffic data (MONICA) for the first time in the Netherlands. In literature, there is only little to be found about Bayesian Networks applied on traffic data. In this thesis, four Bayesian Models are proposed. The model which only takes downstream traffic information into account performs the best. This corresponds with the findings of Treiber and Helbing in [38] in which it is stated that traffic information propagates against the driving the direction with a constant speed of 15 km/h if the traffic is in a congested state.

The proposed Bayesian Model is robust as it gives more or less the same results for different tests with different samples. The Bayesian Model is able to predict traffic congestions for over a time horizon of 5 minutes with a false negative percentage of only 5% and for a prediction horizon of 120 minutes the false negatives percentage is 35% which is promising.

**7. Modeling traffic by historical based models:** Our prediction model, which is based on historical data, shows that it is possible to predict traffic more accurate then a Bayesian Network by making use of historical patterns, if the prediction horizon becomes longer then 30 minutes. The difference is marginal, but is expected that the historical based predictor eventually outperforms the Bayesian Network if the horizon is set to a few hours or even days.

However, it can be concluded that this historical model is able to predict traffic congestions with a false negatives error of 20% for a prediction horizon of 120 minutes in our case study. This is better then the Bayesian model, but still not convincing.

**8. The combination of historical modeling with Bayesian Networks:** In this thesis, we propose a new model: the hybrid model. A combination of a prediction model based on historical data and a Bayesian Network shows to be a very powerful one. The prediction model based on historical data first calculates raw initial estimates of all input nodes for the Bayesian Network. After this first step, the Bayesian Network is able to use its learned conditional probabilities tables to smooth the initial historical based prediction to a more correct traffic prediction.

The prediction performance of the hybrid model is convincing, since it has a false negatives error of only 15% for a prediction horizon for over 150 minutes.

**9. The description of the developed software for this thesis:** The research workbench, which we developed for this thesis, showed to be a very powerful one. All models which are proposed in this thesis could easily be implemented in this research workbench. In the end, the workbench is set up generic, so that it can easily be extended in the future. Our conclusion is that this research workbench is able to combine different existing software libraries and tools to ease the process to traffic modeling.

## 11.4 General conclusion

In general it can be concluded that Bayesian Networks are quite powerful in predicting the average vehicle speed of traffic by making use of downstream information. If the Bayesian Network is strengthened with a initial estimation for its nodes based on a historical based prediction model, it is even able to predict traffic congestions more accurately.

This thesis has only taken the well known traffic bottleneck at the A4 from The Hague to Rotterdam into account as a case study, which shows promising results. Of course, the methods and algorithms developed in this thesis should be evaluated for different locations to get a more convincing overview of the performance.

But, for now, the Hybrid approach which combines Bayesian Networks with a historical based prediction algorithm shows promising results!

A little story...

*In the end, the aim is to provide travelers up-to-date traffic information and accurate predictions. Imagine two persons, named Henk and Ingrid, who are planning to take the car from Venlo to The Hague on a regular Wednesday morning. The time is 07:25 am and Henk just ate his breakfast. Ingrid tells Henk that she will finish her breakfast in the car, since they will probably be in a traffic congestion anyway. The trip should take around 2 hours and they are planning to leave at 07:30 am. Ingrid tells Henk that she will make some sandwiches for lunch, because she does not believe the trip can be made in 2 hours. Henk starts an application on his intelligent navigation device and noticed that it is likely that there will be a traffic congestion somewhere along their route in about 1 hour. The last traffic announcement on the radio did not tell Henk about this, but his intelligent navigation device did! This intelligent device combined historical knowledge about traffic situations along their route and current traffic measurements. Furthermore, the software used a Bayesian Network to predict the speed at subsequent parts of the route and predicts a congestion. Henk tells Ingrid not to make any sandwiches, because his navigation device advises another route, which is only 10 minutes longer then the shortest route but is not likely to become congested. Henk and Ingrid take the car and arrive only 10 minutes late in The Hague. Although we are not there yet, the fundamentals for this intelligent navigation software have been laid in this thesis.*

# Chapter 12

# Future Work

Since the time to complete this thesis is finite, we discuss the future work in this chapter. In the introduction, we explained that the reason for modeling average vehicle speed is to compute travel times which can be used for dynamic routing. Predicting traffic speed is the foundation for predicting travel times. An origin and destination are necessary for predicting travel times, as well as speed predictions for different locations along the route between origin and destination.

We propose to train our models on different locations on Dutch highways. It is interesting to test our models on more complex traffic situations with, for example, merging and exiting lanes. Further, it would be interesting to include inductive loop detectors (ILD) in the Bayesian Network at locations which have a longer distance to the prediction location. In this way, changing traffic situations might be noticed earlier.

Further, we propose to create a bottleneck map of the Netherlands. This map shows the most important bottlenecks for the Dutch road system. Maybe, this map can be a function of the time, since the bottlenecks could be different for different days of the week, different seasons, different weather conditions, etc. Then, our Hybrid model needs to be trained for every bottleneck, so that each bottleneck has its own prediction model. These models are then able to predict the average vehicle speed, so travel times can be computed. If travel times are computed, we know the traveling delay in case of congested traffic. If the delays are known for different routes, travelers can be rerouted dynamically to reach the shortest traveling time. An important issue which should be kept in mind, is that the prediction models in this thesis are not trained for this dynamical rerouting process. If, lets say, 5% of the travelers is rerouted in case of traffic congestions, our historical models might not be representative anymore. The models are trained on a historical database in which there was no dynamic routing system active. It can not be predicted beforehand how our models cope with this, and this should be further investigated.

In the future, we hope that more travelers can be tracked by GPS or other communication systems to get floating car data (FCD). In urban traffic we could track taxis or buses, but at highways we need to track individual travelers. It would even be better if a dataset can be formed out of vehicle navigation systems, since this data consists of real time speed measurements, travel times and the names of origins and destinations. For prediction purposes, it would be beneficial to know the origins and destination of travelers. If the origins and destinations are known, the shortest route can be calculated. For these routes, the travel times can be computed. Since the routes and travel times of the current travelers are known, it can be predicted where the travelers will be driving in the future. The only unknown variables here, are the new travelers who could be joining the current traffic in future.

# Bibliography

[1] R. R. Andrawis and Atiya A.F. A new Bayesian formulation for Holt's exponential smoothing. *Journal of Forecasting*, 28:218–234, 2009.

[2] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[3] K.M. Chaloner and G.T. Duncan. Assessment of a Beta prior distribution: PM Elicitation. *Journal of the Royal Statistical Society, Series D.*, 32:174–180, 1983.

[4] C.E. Cortes, R.O. Lavanya, Jun-Seok, and R. Jayakrishnan. General purpose methodology for estimating link travel times with multiple-point detection of traffic. *Transportation Research Board*, 1802:181–189, 2002.

[5] J.C.C. de Ruiter and W.J.J.P. Schouten. Berekenen van reistijden door het MoniBas systeem. Technical report, Ministerie van Verkeer en Waterstaat, Rijkswaterstaat (AVV) Rotterdam, 2002.

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

[7] M.O. Finkelstein and W. B. Fairley. A Bayesian approach to identification evidence. *Harvard Law Review*, 83:489–517, 1970.

[8] B.E. Flores. A pragmatic view of accuracy measurement in forecasting. *Omega*, 14:93–98, 1986.

[9] J.G. Gooijer and R.J. Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22:443–473, 2006.

[10] D. Heckerman. A tutorial on learning with Bayesian Networks. Technical report, Microsoft Research, Advanced Technology Devision: Microsoft Corporation, Redmond, US, 1995.

[11] S.P. Hoogendoorn. Traffic flow theory and simulation, lecture notes vk4821, Faculty of Civil Engineering and Geosciences - Transportation and Traffic Engineering Section, Delft University of Technology, 2004.

[12] S.P. Hoogendoorn and P.H.L. Bovy. State-of-the-art of vehicular traffic flow modelling. *Journal of Systems & Control in Engineer*, 215:283–303, 2001.

[13] F.V. Jensen, S.H. Aldenryd, and K.B. Jensen. Sensitivity analysis in Bayesian Networks. In *Lecture Notes in Computer Science*, 1995.

[14] H.J. Kim, J.T. Kim, and K.Y. Hwang. Test of IR-DSRC in measuring vehicle speed for ITS applications. *Lecture Notes in Computer Science*, 4097:1012–1020, 2006.

[15] R. Klinkenberg. Approaching vega: The final descent. how to extend RapidMiner 5.0. Technical report, Rapid-I, 2010.

[16] G. Kloot. Melbourne's arterial travel time system. In *Proceedings of the 6th World Congress on ITS (CD-ROM)*, 1999.

[17] K.B. Korb and A.E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, UK, 2004.

[18] T.C. Lethbridge and R. Laganiere. *Object-Oriented Sofware Engineering*. Nc Graw Hill, 2005.

[19] R. Li, G. Rose, and M. Sarvi. Evaluation of speed-based travel time estimation models. *Journal of Transportation Engineering*, 132:540–547, 2006.

[20] C.D.R. Lindveld and R. Thijs. On-line travel time estimation using inductive loop data: The effect of peculiarities on travel time estimation quality. In *Proceedings of the 6th ITS World Congres, Toronto, Canada*, 1999.

[21] J.W.W.J. Loeppky and J.L. Sacks. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51:366–376, 2009.

[22] C. Oh, S. Park, and S.G. Ritchie. A method for identifying rear-end collission risks using inductive loop detectors. *Elsevier's Accident Analysis and Prevention*, 38:295–301, 2006.

[23] S. Oh, S.G. Ritchie, and C. Oh. Real time traffic measurements from single loop inductive signatures. In *peer-reviewed publication of the Transportation Research Board: Papers presented at the TRB 81st Annual Meeting*, 2002.

[24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, California: Morgan Kaufmann, 1988.

[25] C.M. Queen and C.J. Albers. Intervention and causality: Forecasting traffic flows using a dynamic Bayesian Network. *Journal of the American Statistical Association*, 104:669–681, 2009.

[26] J. R. Quinlan and R.L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[27] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[28] J.R. Quinlan. *C4.5 programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[29] J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[30] Rijkswaterstaat. Profiel van de spitsrijder. Wie rijdt er in de spits? Technical report, Rijkswaterstaat Adviesdienst Verkeer & Vervoer, 2006.

[31] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

[32] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, 1948.

[33] S. Sun, C. Zhang, and G. Yu. A Bayesian Network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7:124–132, 2006.

[34] S. Sun, C. Zhang, G. Yu, N. Lu, and F. Xiao. *Bayesian Network Methods for Traffic Flow Forecasting with Incomplete Data*. Springer Berlin, 2004.

[35] S. Sun, C. Zhang, and Y. Zhang. Traffic flow forecasting using a spatio-temporal Bayesian Network predictor. In *Artificial Neural Networks: Formal Models and Their Applications - ICANN*, 2005.

[36] H. Taale. Analyzing loop data for quick evaluation of traffic management measures. In *European Transport Conference*, 1998.

[37] J. Tayman and D.A. Swanson. On the validity of mape as a measure of population forecast accuracy. *Population Research and Policy Review*, 18:299–322, 1999.

[38] M. Treiber and D. Helbing. Reconstructing the spatio- temporal traffic dynamics from stationary detector data. *Cooperative Transportation Dynamics*, 1:3.1–3.24, 2002.

[39] S.M. Turner, W.L. Eisele, R.J. Benz, and D.J. Holdener. *Travel Time Data Collection Handbook*. Texas Transportation Institute, 1998.

[40] H. van Knotsenburg. Monitoring casco (monica): Handleiding afnemers dynamische gegevens. Technical report, Ministerie van Verkeer en Waterstaat (The Dutch Ministry of Transport, Public Works and Water Management), 2005.

[41] J.W.C. van Lint. *Reliable Travel Time Prediction for Freeways*. PhD thesis, The Netherlands TRAIL Research School, 2004.

[42] J.W.C. van Lint. Voorspelde reistijden op DRIPs. Technical report, Technische Universiteit Delft, Faculteit Civiele Techniek en Geowetenschappen, Sectie Transport and Planning, 2005.

[43] L.D. Vanajakshi. *Estimation and prediction of travel time from loop detector data for intelligent transportation systems and applications*. PhD thesis, Texas A&M Univeristy, 2004.

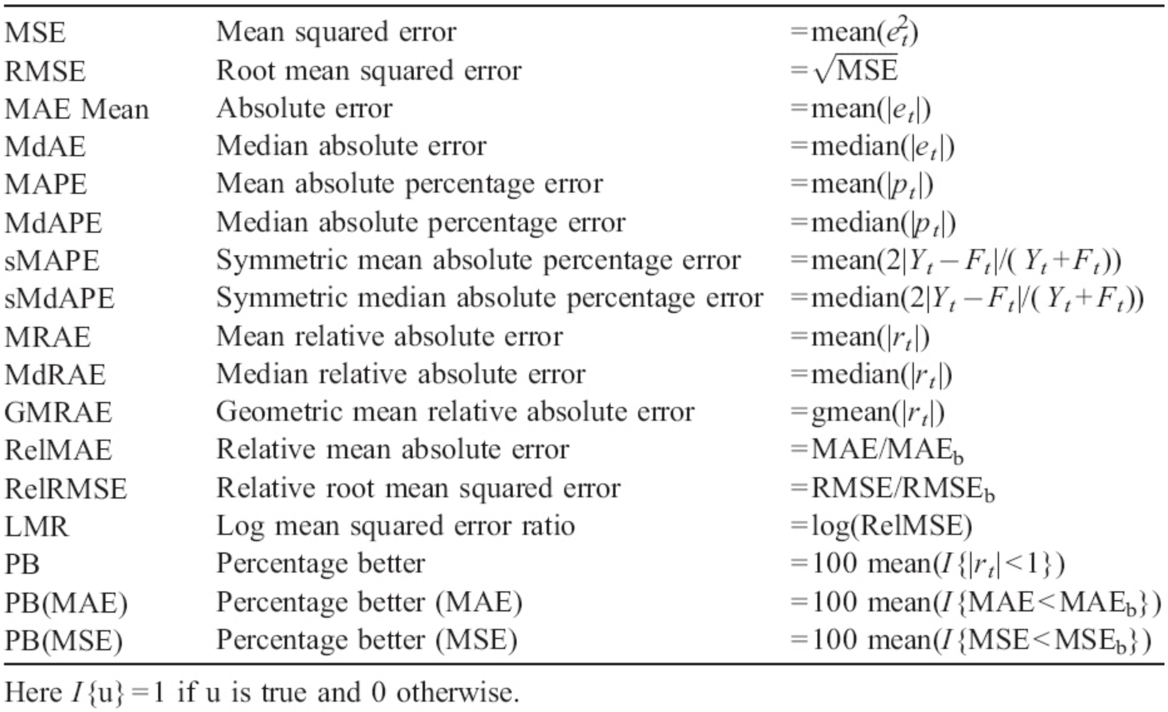
[44] L.D. Vanajakshi, B.M. Williams, and L.R. Rilett. Improved flow-based travel time estimation method from point detector data for freeways. *Journal of Transportation Engineering*, 135:26–36, 2009.

[45] R.L. Winkler. The assessment of prior distributions in Bayesian analysis. *Journal of the American Statistical Association*, 62:776–800, 1967.

[46] Y.J. Yu and M. Cho. A short-term prediction model for forecasting traffic information using Bayesian Networks. In *Third International Conference on Convergence and Hybrid Information Technology*, 2008.

# Appendix A

# Commonly used Forecast Accuracy Measures



| | | |
|---|---|---|
| MSE | Mean squared error | $=\text{mean}(e_t^2)$ |
| RMSE | Root mean squared error | $=\sqrt{\text{MSE}}$ |
| MAE Mean | Absolute error | $=\text{mean}(\lvert e_t \rvert)$ |
| MdAE | Median absolute error | $=\text{median}(\lvert e_t \rvert)$ |
| MAPE | Mean absolute percentage error | $=\text{mean}(\lvert p_t \rvert)$ |
| MdAPE | Median absolute percentage error | $=\text{median}(\lvert p_t \rvert)$ |
| sMAPE | Symmetric mean absolute percentage error | $=\text{mean}(2\lvert Y_t - F_t \rvert/(Y_t + F_t))$ |
| sMdAPE | Symmetric median absolute percentage error | $=\text{median}(2\lvert Y_t - F_t \rvert/(Y_t + F_t))$ |
| MRAE | Mean relative absolute error | $=\text{mean}(\lvert r_t \rvert)$ |
| MdRAE | Median relative absolute error | $=\text{median}(\lvert r_t \rvert)$ |
| GMRAE | Geometric mean relative absolute error | $=\text{gmean}(\lvert r_t \rvert)$ |
| RelMAE | Relative mean absolute error | $=\text{MAE}/\text{MAE}_\text{b}$ |
| RelRMSE | Relative root mean squared error | $=\text{RMSE}/\text{RMSE}_\text{b}$ |
| LMR | Log mean squared error ratio | $=\log(\text{RelMSE})$ |
| PB | Percentage better | $=100\ \text{mean}(I\{\lvert r_t \rvert < 1\})$ |
| PB(MAE) | Percentage better (MAE) | $=100\ \text{mean}(I\{\text{MAE} < \text{MAE}_\text{b}\})$ |
| PB(MSE) | Percentage better (MSE) | $=100\ \text{mean}(I\{\text{MSE} < \text{MSE}_\text{b}\})$ |

Here $I\{\text{u}\}=1$ if u is true and 0 otherwise.

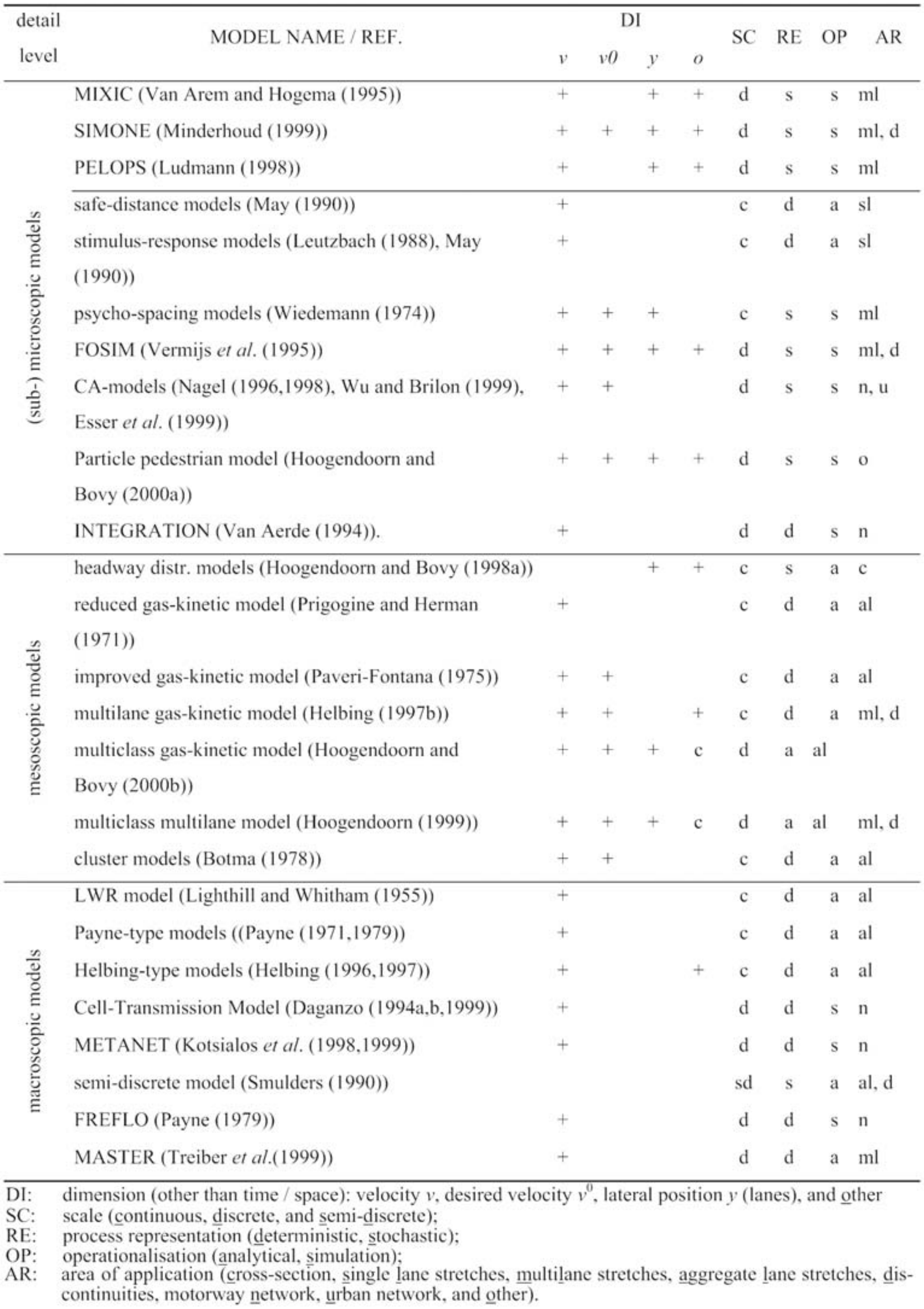
Figure A.1: An overview of commonly used forecast accuracy measures [9]

# Appendix B

# An overview of Traffic Flow Models

Figure B.1 shows a convenient overview of traffic flow models. This is an overview since the 1950's. The overview is grouped by:

- (sub)- microscopic models
- mesoscopic models
- macroscopic models.



| detail level | MODEL NAME / REF. | DI | | | | SC | RE | OP | AR |
|---|---|---|---|---|---|---|---|---|---|
| | | *v* | *v0* | *y* | *o* | | | | |
| (sub-) microscopic models | MIXIC (Van Arem and Hogema (1995)) | + | | + | + | d | s | s | ml |
| | SIMONE (Minderhoud (1999)) | + | + | + | + | d | s | s | ml, d |
| | PELOPS (Ludmann (1998)) | + | | + | + | d | s | s | ml |
| | safe-distance models (May (1990)) | + | | | | c | d | a | sl |
| | stimulus-response models (Leutzbach (1988), May (1990)) | + | | | | c | d | a | sl |
| | psycho-spacing models (Wiedemann (1974)) | + | + | + | | c | s | s | ml |
| | FOSIM (Vermijs *et al*. (1995)) | + | + | + | + | d | s | s | ml, d |
| | CA-models (Nagel (1996,1998), Wu and Brilon (1999), Esser *et al*. (1999)) | + | + | | | d | s | s | n, u |
| | Particle pedestrian model (Hoogendoorn and Bovy (2000a)) | + | + | + | + | d | s | s | o |
| | INTEGRATION (Van Aerde (1994)). | + | | | | d | d | s | n |
| mesoscopic models | headway distr. models (Hoogendoorn and Bovy (1998a)) | | | + | + | c | s | a | c |
| | reduced gas-kinetic model (Prigogine and Herman (1971)) | + | | | | c | d | a | al |
| | improved gas-kinetic model (Paveri-Fontana (1975)) | + | + | | | c | d | a | al |
| | multilane gas-kinetic model (Helbing (1997b)) | + | + | | + | c | d | a | ml, d |
| | multiclass gas-kinetic model (Hoogendoorn and Bovy (2000b)) | + | + | + | c | d | a | al | |
| | multiclass multilane model (Hoogendoorn (1999)) | + | + | + | c | d | a | al | ml, d |
| | cluster models (Botma (1978)) | + | + | | | c | d | a | al |
| macroscopic models | LWR model (Lighthill and Whitham (1955)) | + | | | | c | d | a | al |
| | Payne-type models ((Payne (1971,1979)) | + | | | | c | d | a | al |
| | Helbing-type models (Helbing (1996,1997)) | + | | | + | c | d | a | al |
| | Cell-Transmission Model (Daganzo (1994a,b,1999)) | + | | | | d | d | s | n |
| | METANET (Kotsialos *et al*. (1998,1999)) | + | | | | d | d | s | n |
| | semi-discrete model (Smulders (1990)) | | | | | sd | s | a | al, d |
| | FREFLO (Payne (1979)) | + | | | | d | d | s | n |
| | MASTER (Treiber *et al*.(1999)) | + | | | | d | d | a | ml |

DI: dimension (other than time / space): velocity *v*, desired velocity $v^0$, lateral position *y* (lanes), and other
SC: scale (continuous, discrete, and semi-discrete);
RE: process representation (deterministic, stochastic);
OP: operationalisation (analytical, simulation);
AR: area of application (cross-section, single lane stretches, multilane stretches, aggregate lane stretches, discontinuities, motorway network, urban network, and other).

Figure B.1: Overview of traffic flow models by hoogendoorn since the 1950’s [12]