Bayesian Data Assimilation for Improved Modeling of Road Traffic

Chris van Hinsbergen

Bayesian Data Assimilation for Improved Modeling of Road Traffic

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College van Promoties,
in het openbaar te verdedigen op dinsdag 16 november 2010 om 12:30 uur

door

Christopher Philip IJsbrand VAN HINSBERGEN

civiel ingenieur
geboren te Enschede

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. H.J. van Zuylen
Prof. ir. F.M. Sanders

Samenstelling promotiecommissie:

Rector Magnificus	Voorzitter
Prof. dr. H.J. van Zuylen	Technische Universiteit Delft, promotor
Prof. ir. F.M. Sanders	Technische Universiteit Delft, promotor
Dr. ir. J.W.C. van Lint	Technische Universiteit Delft, co-promotor
Prof. dr. ir. J.H. van Schuppen	Technische Universiteit Delft
Prof. dr. T.M. Heskes	Radboud Universiteit Nijmegen
Prof. dr. P.B. Mirchandani	Arizona State University
Prof. dr. L.R. Rilett	University of Nebraska-Lincoln

This dissertation thesis is the result of a Ph.D. study carried out from 2007 to 2010 at Delft University of Technology, Faculty of Civil Engineering and Geosciences, Transport & Planning Department. The research was sponsored by the Advanced Traffic MOnitoring (ATMO) project under the Transumo (TRANsition SUstainable MObility) program. For more information, please visit www.atmo.tudelft.nl.

TRAIL Thesis Series T2010/9, the Netherlands TRAIL Research School

TRAIL
P.O.Box 5017
2600 GA Delft
The Netherlands
Phone: +31 (0) 15 278 6046
Fax: +31 (0) 15 278 4333
E-mail: info@rsTRAIL.nl

ISBN 978-90-5584-132-5

Printed in the Netherlands

Voor Pauline

Preface

In April 2006, 10 minutes after I had graduated from my Master's, I was approached by my professor Frank Sanders with the question whether I would like to perform a PhD. Within a few days, I was talking to Hans van Lint about travel time prediction and a possible PhD position. Although the start was a bit cautious, very soon the cooperation between me and Hans became very positive (although Hans had to advise me to follow a few additional courses on traffic theory). Soon I understood that without money, a position would be hard to obtain. I started to write my first proposal for a grant (more would follow), a Casimir grant in this case which I applied for together with Vialis in Haarlem and which was denied because the plan was 'too ambitious'. Nonetheless, the contacts with Vialis were that good, that we all decided that a combined research program was worth a shot. Therefore, the first of January 2007 I officially started my PhD career. I would like to thank Vialis, and especially Wim Broeders, for enabling this start of what was to become a very positive working experience, even though the combined research program between the TU and Vialis didn't work out in the long run.

And so the work had started. Besides the everyday pleasure of trying to solve a Hessian of some new model I was trying to use, the papers I delivered put me in great places. The first paper I ever wrote immediately delivered me a trip to China, which I very much enjoyed. I got to know my colleagues better, and still remember vividly how we celebrated Pauline's birthday elevated 400 meters above the ground in Shanghai, just after Serge, Huizhao and I had a near-death experience in a taxi driving backwards on the highway. One year later, I had the opportunity to present a paper in Surfer's Paradise, Australia. The one month trip that me and Pauline made afterwards was without a doubt the best holiday I ever had. Of course, the trips were not always full of sunshine - one 'Hoegaarden Grand-Cru' too much made me swear never to drink again at the DTA conference in Leuven, Belgium.

I was the luckiest of PhD's with my supervisors. I valued my independence, and I suppose they valued my independence too, but whenever I needed help, it was available. Hans, I was always in close contact with you, being office neighbors, and I have always valued greatly your ability to find time to help me out when I was stuck. Also, I valued the very quick responses of Hans, Henk and Frank Sanders whenever I had written a new paper. Within days, I could expect very serious and thorough remarks and questions

which have helped to get my papers published. I would like to thank all three of you for letting me do what I thought was good, but still steering me whenever I needed steering.

Already after a few months, I got to know Frank Zuurbier as not only a good pingpong-player and a hard worker, but also as an entrepreneur, and more importantly, as a friend. Very quickly, we both decided that we did not want to work for a boss after our PhDs, and we started making plans for billion-euro companies based on (in random order) GPS-dating, track & tracing, horizontal candles, a website for trading puzzles, a GSM with medical check-up abilities and Hitman Mobile, but (surprisingly) all ideas appeared not to be good enough. Then, in 2008 we started a new adventure. For some months, we had had the idea to use the model JDSMART after our PhDs to deliver new traffic information services. Soon we decided that the perfect domain name for this was Fileradar.nl, but discovered that the domain name was taken. Luckily for us, we managed to contact the owners of the domain name, two young entrepreneurs of the faculty of Aerospace engineering who had just given up on the whole idea of being an entrepreneur. They were so kind to share their experiences with us, as well as hand over the domain name, over a very expensive diner in Amsterdam. I would like to thank both of them for being so kind (Frank, you still owe me half the diner!). The adventure then took off. We started to apply for funding wherever we could. We entered a competition called the Academic Year Prize, for which we had to work our butts off but which was an extremely positive experience, winning the second prize. From there on, we submitted for a Valorisation Grant at STW (granted), we competed in the Delft Design and Engineering Award (did not even make it to the finals) and, together with TomTom, worked on a tender for the NDW (denied). Our last effort, the IMM-subsidy from the Ministry of Transport, Public Works and Water Management, was finally granted in September 2010. The coming period will therefore be very intensive, but I have very high confidence in a positive outcome, although I still have my doubts about the billion-euroness of the undertaking. All in all, Frank, without you there would be no Fileradar and I probably would have ended up working for a boss. I am extremely grateful for having you as a colleague and a friend, and I hope that we will continue to work together for many years.

Through the years I got to know several great roommates: Minwei, Femke, Thomas, Maaike, Nina, Victor and Leila. Not only did they help me on many occasions when I had questions regarding Matlab, Latex, mathematics or traffic science in general, but also they ensured that there was a very positive atmosphere in room 4.31. Femke, I would like to apologize for all those occasions where I shouted out a bit too loud looking for my coffee-card or keys. Also, I would like to apologize to the entire department (and the departments on Floor 3 and 5) for my enthusiasm at the pingpong table. Again, I might have shouted out a bit too loud on occasions. On a more positive side, all the practice at the pingpong table helped me to get to know most people at the department, helped me to empty my brain between the Hessians and the programming of JDSMART, and helped me to become the Transport & Planning Pingpong Champion 2010. I might enter the

world championship next year.

Besides my roommates, I got to know other colleagues as friends. Adam, thank you for your laughs, for fitness and tennis, and for dealing with me storming into your room when I needed a break. The same goes for the other inhabitants of ‘the smart-room’ Olga, Niels, Hao and Tamara. Kees and Theo, thank you for teaching me how to play pingpong (can I now finally borrow that book about it?), and sorry for the remarks that I might have made about your age. All Chinese colleagues, (especially Hao Liu, Hao Li and Huizhao) thank you for teaching me your fantastic language and for the many great diners at the Chinese restaurants in the Hague and Rotterdam. Mario, thank you for sharing your life experience with me during our trip to Washington.

During my PhD, I could always count on the full support of my family. Rix, thank you so much for always being there for me. Douwe, thank you for sharing your scientific views with me and thank you for suggesting that I should make my thesis paper-based, and Lars, you are a fantastic brother. Both of your, thank you for being my paranymphs. Peter, thank you for your continuous support. I love you all, not to forget all your partners and my niece and nephew. My family-in-law, Cees and Tonny, Suzanne and Sipco and the kids, I am a lucky man to have been married into your family.

Finally, I would like to dedicate a paragraph to the love of my life, Pauline. First of all, I would like to thank you from the bottom of my heart for supporting me in all those years and for being so patient, also when Frank and I had to spend evening after evening to finish a document or presentation. Also, I would like to thank you for listening to me explaining something difficult that I was doing, even if it was something extremely technical which I didn’t even fully understand. Not only did you support me, but you also gave me the energy to keep going, and you put my feet back on the ground in those cases where I became overenthusiastic. The last months have been very intensive, with me working hard on the PhD, trying to build a company, and most importantly, the birth of our fantastic daughter Eefje. Of course, without you neither of these three would have been possible. I love you with all my heart and I thank you so much for loving me back.

Contents

1 Introduction **1**
- 1.1 Model formulation and data assimilation . . . 1
- 1.2 Objectives and scope of this thesis . . . 4
- 1.3 The three tasks of data assimilation . . . 5
  - 1.3.1 Model validation and identification . . . 5
  - 1.3.2 Model calibration . . . 5
  - 1.3.3 Estimation and prediction . . . 6
- 1.4 Belief, reasoning and evidence . . . 6
- 1.5 The Bayesian framework for data assimilation . . . 7
- 1.6 Outline of this thesis . . . 10
- 1.7 Contributions of this thesis . . . 12
  - 1.7.1 Scientific and methodological contributions . . . 12
  - 1.7.2 Practical contributions . . . 14

2 Bayesian calibration and comparison of car-following models **17**
- 2.1 Introduction . . . 18
  - 2.1.1 State of the art in model calibration and comparison . . . 18
  - 2.1.2 Structure of this chapter . . . 20
- 2.2 Methodology . . . 21
  - 2.2.1 Bayesian Inference: from prior to posterior . . . 21
  - 2.2.2 Description of posterior distribution of parameters . . . 22
  - 2.2.3 Bayesian framework for model comparison . . . 23
  - 2.2.4 Evidence for CHM model . . . 24
  - 2.2.5 Evidence for Helly model . . . 26
- 2.3 Experiment . . . 29
- 2.4 Results . . . 29
- 2.5 Discussion and conclusion . . . 31

3 Bayesian committee of regression models to predict travel times **35**
- 3.1 Introduction . . . 36
- 3.2 Methodology . . . 37

3.2.1 Bayesian framework for model fitting and comparison 37
3.2.2 Approximating the model evidence 39
3.2.3 Normalizing the likelihood 40
3.2.4 The Occam factor for the multidimensional case 41
3.2.5 Combination strategies 41
3.3 Proof of concept: two simple models 42
3.3.1 Model 1: Linear Regression 43
3.3.2 Model 2: Locally Weighted Linear Regression 45
3.4 Results 47
3.5 Discussion and conclusion 49

4 Bayesian committee of neural networks to predict travel times 51
4.1 Introduction 52
4.1.1 Committees of prediction models 52
4.1.2 Artificial neural networks 53
4.1.3 Objective of this study 53
4.2 Methodology 54
4.2.1 Feed forward neural networks for travel time prediction 54
4.2.2 Bayesian trained neural networks for travel time prediction 57
4.2.3 The evidence framework for committees of neural networks . . . 59
4.2.4 Error bars on each committee member's predictions 62
4.2.5 Step-by-step procedure: committee of neural networks 63
4.3 Experiment 63
4.3.1 Data 64
4.3.2 Parameters 64
4.4 Results 65
4.5 Discussion 67
4.6 Conclusion 68

5 Bayesian committee of state space neural networks to predict travel times 71
5.1 Introduction 72
5.2 Methodology 72
5.2.1 State Space Neural Networks for travel time prediction 72
5.2.2 Neural network training formulated as Bayesian inference 75
5.2.3 Determination of the gradient 76
5.2.4 Determination of the Hessian 77
5.3 Experiment 79
5.3.1 Data 80
5.3.2 Stopping criterion 80
5.4 Results 81
5.5 Discussion and conclusion 82

6 Bayesian calibration of the Extended Kalman Filter 85
6.1 Introduction . . . 86
6.2 Methodology: Bayesian estimation of noise parameters . . . 87
6.2.1 Bayesian derivation of the EKF . . . 88
6.2.2 Bayesian derivation of $\alpha[k]$ and $\beta[k]$ . . . 91
6.3 Experiment . . . 95
6.4 Discussion and conclusions . . . 96

7 The Localized Extended Kalman Filter for fast traffic state estimation 99
7.1 Introduction . . . 100
7.2 Methodology . . . 101
7.2.1 The LWR model solved by the Godunov scheme . . . 101
7.2.2 Extended Kalman Filter . . . 104
7.2.3 Global Extended Kalman Filter . . . 106
7.2.4 Localized Extended Kalman Filter . . . 106
7.3 Experiment 1: synthetic data . . . 111
7.4 Experiment 2: real data . . . 117
7.5 Discussion and conclusion . . . 120

8 Conclusions and recommendations 123
8.1 Conclusions . . . 123
8.1.1 Current state of practice . . . 124
8.1.2 Bayesian framework for data assimilation . . . 126
8.1.3 Car-following behavior . . . 128
8.1.4 Travel time prediction . . . 128
8.1.5 Extended Kalman Filter parameters . . . 129
8.1.6 Localized Extended Kalman Filter . . . 129
8.2 Implications for practitioners . . . 130
8.3 Recommendations and future research . . . 131
8.3.1 Bayesian framework for data assimilation . . . 131
8.3.2 Car-following behavior . . . 132
8.3.3 Travel time prediction . . . 133
8.3.4 Extended Kalman Filter parameters . . . 133
8.3.5 Localized Extended Kalman Filter . . . 134

A Exact gradient and Hessian for Recurrent Neural Networks 135
A.1 Determination of the gradient . . . 135
A.1.1 Determination of $\partial y/\partial w$ . . . 136
A.1.2 The gradients for each layer . . . 138
A.2 Determination of the Hessian . . . 139
A.2.1 Both output layer . . . 140

A.2.2 Output layer and hidden layer 140
A.2.3 Output layer and context layer 140
A.2.4 Both hidden layer 141
A.2.5 Hidden layer and context layer 142
A.2.6 Both context layer 143
A.3 Outer product approximation of the Hessian 144

Bibliography **145**

Summary **159**

Samenvatting **163**

Curriculum Vitae **169**

TRAIL Thesis Series **173**

List of Figures

1.1 Schematic representation of the circle of model development 2
1.2 The focus of this thesis is on data assimilation in road traffic 4
1.3 Four uses if traffic data and the process of data assimilation 8
1.4 The structure of this thesis' chapters 11

2.1 Cumulative distribution of γ 27
2.2 Cumulative distribution of α and β 28
2.3 The log evidence for the two models for 9 of the 229 drivers 30
2.4 The actual versus predicted speed for driver 47 and 48 33

3.1 Framework for combining prediction models 37
3.2 The evidence for the one-dimensional case 40
3.3 Schematic illustration of locally weighted regression 45
3.4 The A12 network from Zoetermeer to Voorburg, the Netherlands 47
3.5 Prediction results of the single and combined models on April 16, 2007 . 50

4.1 Schematic representation of a neural network 54
4.2 Evidence and test error during training of a neural network 61
4.3 The A12 motorway from Zoetermeer to The Hague 64
4.4 Log evidence versus test error for 84 neural networks 65
4.5 MAPE versus committee size 65
4.6 Prediction results with prediction intervals for an example day 67

5.1 Schematic representation of a state space neural network 73
5.2 The A12 motorway from Zoetermeer to The Hague 80

6.1 Ground truth network and network with 'process noise' 96
6.2 Errors for constant and Bayesian hyperparameters 96

7.1 Example of the Smulders fundamental diagram 102
7.2 Progression of error covariance in different conditions 107
7.3 Schematic representation of the L-EKF 109

7.4 Network for the synthetic data experiment 112
7.5 Ground truth, distorted and corrected density patterns (1) 114
7.6 Ground truth, distorted and corrected density patterns (2) 115
7.7 Results in accuracy and computation times for the different filters 116
7.8 Network for the real-world experiment: Rotterdam's freeways 117
7.9 Sorted speeds for a detector, used for calibration 118
7.10 Real-world results in computation times for the different filters 120

List of Tables

2.1 Probabilities of both models averaged over all 229 drivers 31

3.1 Results of the different prediction models 48

4.1 Performance of individual models versus committee 66
4.2 Effects of early stopping 66

5.1 Results for 5 and 15 minute prediction horizon 81
5.2 Committee errors 82

7.1 Parameter settings of network for the synthetic data experiment 112
7.2 Comparison of accuracy of L-EKF and G-EKF in real-world experiment . 119

Chapter 1

Introduction

In this thesis, the focus will be on *road traffic*. Road traffic has played a major role in everyday human life over the past decades. It has led to economic growth and an increased mobility and freedom of people, but it has also led to a number of negative side effects such as unsafety, traffic congestion and pollution.

1.1 Model formulation and data assimilation

Many different phenomena of the road traffic system have been studied over the decades, to improve the performance of the traffic system or to alleviate some of the negative side effects of traffic. For the description of these phenomena, over the years scientists have proposed mathematical models to describe these phenomena, either deductively (by reasoning based on axioms, laws, etc.), or inductively by investigated and interpreting traffic data. In general, traffic science is mostly an empirical science where the inductive formulation of models appears to be dominant. In Figure 1.1 this process is depicted schematically. On the left the real-world is drawn, which in this case represents the road traffic system. Sensors are used to measure certain aspects of this reality, which could be (average) speeds of passing vehicles somewhere in the road network. Using such sensors, data is collected, which usually needs to be cleaned and checked for validity after which it is stored. This cleaned data can then be interpreted to analyze regularities in order to develop new theories and concepts, as Bruce Greenshields for example already in 1934 analyzed that with an increasing density of vehicles the average speed has the tendency to decrease (Greenshields, 1934). Based on these new theories, new hypotheses are proposed to explain the observed phenomena, which are usually formulated in mathematical models. These models are then used to make predictions of reality.

There are important interactions between the three steps of data acquisition, the development of theories and concepts and model formulation. First of all, when a new mathematical formulation is formed based on a hypothesis, the model needs to be validated

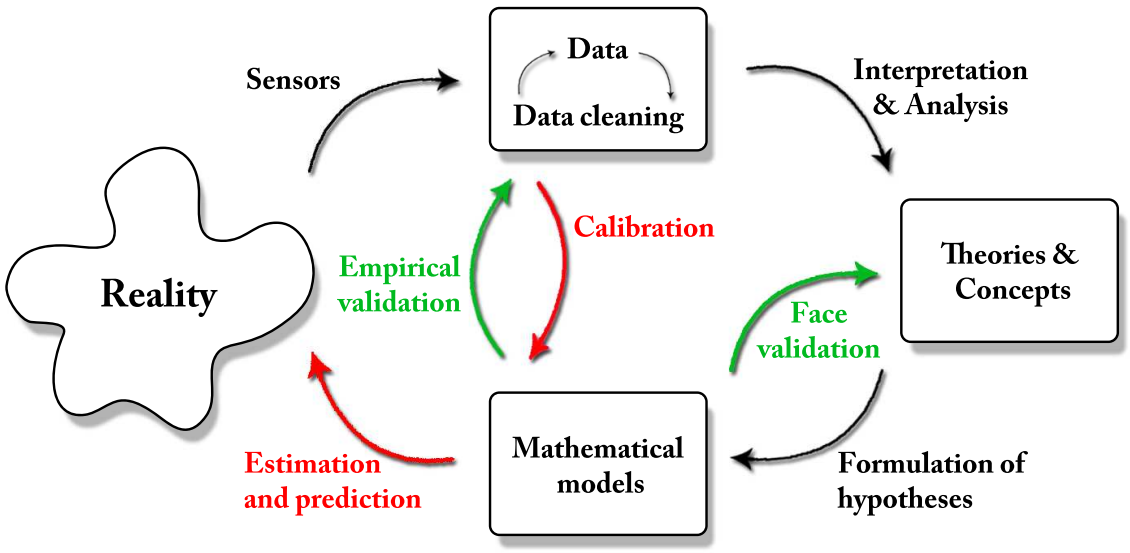


Figure 1.1: Schematic representation of the circle of model development

against the original theory that was developed. This is called *face validation*. Generally, the next step is to compare the model against (new, unseen) measurements of reality, to see if the model is able to reproduce these measurements of reality. This is called *empirical validation*, or as Miguel de Cervantes wrote in 1615: 'the proof of the pudding is in the eating'[1]. In order for a model to make a prediction, it generally needs historical and/or real-time data for calibration and as input for predictions. The validated and calibrated model can then make estimates and predictions of reality which can be used for a variety of applications. All of these interactions are indicated in Figure 1.1 by arrows.

However, traffic is a system with stochastic properties because it is the result of human behavior subject to temporarily or permanently changing external conditions (due to incidents, large events, changing weather, globalization, changing political landscape or the credit crunch to name a few). Furthermore, there are different types of roads (anywhere from a one lane farm road to a twenty-six-lane freeway[2]) on which behavior of drivers cannot be expected to be equal. Notably, there are also large differences of travel and driving behavior between countries (Pucher, 1988; Golias and Karlaftis, 2001; Özkan et al., 2006), as well as large international differences in road layout, traffic laws and traffic management. For many phenomena more than one plausible theory has been proposed on the basis of the same empirical evidence. This has subsequently led to a multitude

[1] In fact, his original text was not about pudding but about eggs: 'al freír de los huevos lo verá', or 'it will be seen in the frying of the eggs' (de Cervántes Saavedra, 1615).

[2] For example, the Katy Freeway in Houston, Texas is currently being widened to 26 lanes.

of mathematical models describing the same phenomena. For example, in Ossen (2008) eight different models were identified for the task of modeling car-following behavior, and in van Hinsbergen et al. (2007) over a hundred different models were identified for the prediction of single traffic variables such as flow or travel time.

Because of the fact that so many models exist for the description of the same traffic phenomenon, a user that wants to model a certain traffic phenomenon generally has to make a choice from a wide variety of models. When choosing between models one possible solution is to analyze them on certain properties such as mathematical simplicity, numerical stability or computation speed (Daganzo, 1995b; Aw and Rascle, 2000; Ran, 2000; Nagel et al., 2003; Vlahogianni et al., 2004). A second approach is to choose models based on their ability to predict reality (or measurements of reality) (Smith and Demetsky, 1997; Lee et al., 1998; Huisken and van Maarseveen, 2000; Nikovski et al., 2005). The former approach is close to *face validation* of multiple models, while the second can be interpreted as the *empirical validation* of multiple models as indicated in Figure 1.1.

From the previous, it appears that data is used for different tasks in the circle of model development: it is used to formulate new theories, to validate individual models against reality, to choose one or more models from a selection of available validated models, to calibrate the models so that they make optimal predictions, and finally as input to calibrated models for optimal estimates or predictions of reality. Because choosing between models can be seen as validation of each model individually and comparing the relative outcomes, often these two steps are taken together. It is clear that there are strong interactions between all these steps: models can only be properly validated if they are calibrated properly, and predictions are only expected to be accurate if the models are validated first, and if they are calibrated.

This thesis deals with the use of data together with models that describe phenomena that have been analyzed by scientists studying road traffic. The simultaneous treatment of data and models is often termed *data assimilation* (Robinson and Lermusiaux, 2001), although the term has been used for various meanings depending on the field of interest. In this thesis, data assimilation is defined as follows:

> *"Data assimilation is the use of techniques aimed at the treatment of data in coherence with models in order to construct an as accurate and consistent picture of reality as possible. It comprises the use of data for model validation and identification (choosing between models), model calibration and estimation and prediction and specifically deals with the interactions between all these tasks."*

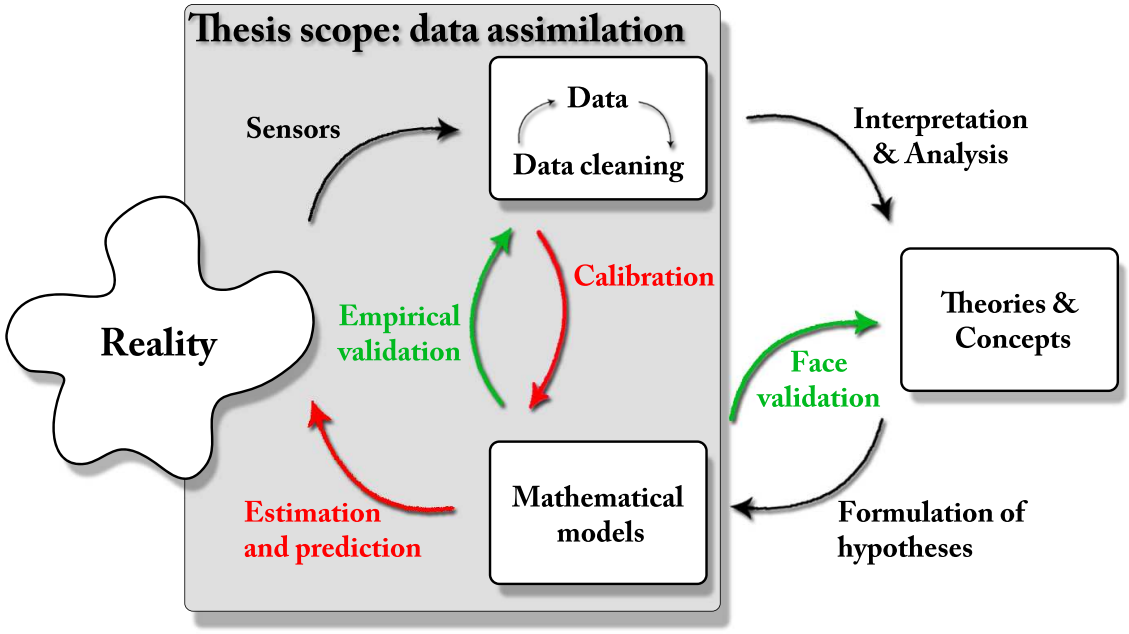


Figure 1.2: The focus of this thesis is on data assimilation in road traffic

1.2 Objectives and scope of this thesis

As the title of this dissertation indicates, this thesis focusses on data assimilation in road traffic. Extensive literature studies, as will be presented in Chapters 2 - 7, reveal that no method exists in the field of (road) traffic science that consistently deals with all steps of data assimilation. The goal of this thesis is therefore to find a methodology that allows for structural treatment of data in coherence with models. The methodology is required to be applicable to a wide variety of models, because different types of data and models are used throughout all subfields of road traffic science. This scope is represented in Figure 1.2 by the gray box.

This thesis proposes a unified method for the three tasks of data assimilation: model validation & identification - calibration - estimation & prediction, applied to different traffic phenomena. The thesis will specifically not use data to develop new models, but will use data only to improve applications with existing models. Therefore, the goal of this thesis is defined as follows:

> *"to find a unified methodology for data assimilation for a wide range of models describing different road traffic phenomena, so that more accurate and consistent predictions can be made of the road traffic system".*

The remainder of this chapter is organized as follows. In 1.3 the three tasks of data assimilation that were identified above are treated more extensively. Then, in 1.5 a unified

framework for data assimilation is described. In 1.6 the outline of this thesis is treated, followed by a description of the contributions of this thesis in 1.7.

1.3 The three tasks of data assimilation

There are three tasks in data assimilation: (1) the validation and identification of (the best) model(s) for a certain application, (2) the calibration of the chosen model(s) for best performance and (3) the use of data as an input to the chosen and calibrated model(s) for an estimate or a prediction of reality. Each of these three are described in more detail below.

1.3.1 Model validation and identification

A scientist who has formulated a new model needs to put his or her model to the test: *validation* is required. Usually, models are validated first by analyzing the properties of their outcomes to see if they do what they are intended to do and if they are internally consistent ('face validity'), and second by comparing the model with (measurements of) the real traffic system ('empirical validity'). In case models already exist for the task at hand, it is even more interesting to see whether the newly developed model outperforms existing models.

Even in the case when no new model has been developed, model *identification* needs to take place: a scientist or practitioner who wants to model a certain phenomenon will have to choose one or more models from all available models. Literature on model performance generally is not conclusive about 'the best model' due to differences in the performance measures that are used, in the types, location (different countries) and layout of roads that the models are applied to, in the type and size of the data sets used for the comparison and in the methods used for calibration of the models (van Hinsbergen et al., 2007).

Usually, models are chosen based on whatever is available or whatever the scientist or practitioner is familiar with. This thesis proposes to use a more systematic way for choosing between multiple models using traffic data. Furthermore it should be noted that apart from choosing between models, also the option exists of using multiple models in parallel for the same task, and to combine their predictions using for example a weighted average of the individual outcomes of the models. This is called a *committee* or *ensemble*. In this thesis such a committee will be used on different occasions.

1.3.2 Model calibration

When a model is applied to a real world application, it almost always requires *calibration*. The small or large amounts of data that are available to a scientist or a practitioner need to

be used to tune the model to the specific situation that the user is building an application for. This is done by setting the *parameters* to specific values that are expected to result in the highest performance of the chosen model(s). The performance measure used to define what the highest performance exactly is, as well as the method to optimize this measure, is user defined and therefore heavily influences the outcomes of the calibration task.

1.3.3 Estimation and prediction

Finally, when the user has identified the model(s) that is/are best suited for his or her application, and all models have been calibrated using data, then the chosen model(s) can be used to make estimates or predictions about certain aspects of the traffic system.

In application of a calibrated model a separation between *estimation* and *prediction* needs to be made. Estimation can be defined as a 'prediction in the past', in other words, a reconstruction of reality, while a prediction is in the future. The process for the two is exactly the same: data (either historic data for estimation, or real-time data for prediction) serves as input to the model, which then makes a prediction based on its mathematical structure and its parameter values that were obtained through calibration.

Figure 1.3 schematically depicts the different steps of data assimilation and their interrelations. In order to systematically describe the mechanisms and interrelationships of the steps identified in Figure 1.3, first define H_q to be the underlying assumptions of a certain model q, which is part of a collection of models $q \in \boldsymbol{M}$. H_q equals the model paradigm, i.e. the blueprint of the model, including for example the mathematical structure of the model, the type of data that should be used as input, the number and type of parameters it contains and the variable(s) that is/are predicted with the model. Let us first take for example the model validation step. Validation of such a hypothesis can be seen as trying to find the user's degree of belief that the hypothesis H_q is correct. Such belief is based on evidence in favor of or against the hypothesis. In 1.4, different ways of reasoning to quantify such a degree of belief are treated.

1.4 Belief, reasoning and evidence

Several mathematical "theories of evidence" have been proposed for the quantification of someone's degrees of belief in something, such as *Bayesian Inference*, *Dempster-Shafer's theory* or the *Transferable Belief Model* (TBM) (Brachman and Levesque, 2004). Each of these frameworks tries to combine objective information such as statistical probabilities with subjective information such as prior belief to express the user's confidence in some outcome. Dempster-Shafer's theory is a generalization of Bayesian inference, and the TBM is again an elaboration of Dempster-Shafer's theory. The main difference between these frameworks is that Dempster-Shafer's theory also deals with concepts such as ignorance and confidence which is not part of the Bayesian inference framework, and that

the TBM specifically includes the "open-world assumption": it may well be that the set of available alternatives is not exhaustive, so that there is reason to believe that an event not described in the set of alternatives will occur, i.e. $P(\emptyset) \geq 0$. For example, when tossing a coin one usually assumes that Head or Tail will occur. The open-world assumption is that the coin could also land on its side, be hit by an accidentally passing bullet or spontaneously dissolve in thin air so that neither Head nor Tail occurs. Apart from these three frameworks, there are also related concepts such as *Fuzzy Logic* (Zadeh, 1965) and *Possibility Theory* (Zadeh, 1978).

Which framework to use is part a rather complex and long-running debate which is inappropriate to repeat here. Eventually, it boils down to a personal preference as it is hard to win this debate based on arguments and because each of these techniques can be interpreted in so many ways that there is for example even debate on whether Dempster-Shafer's theory is a theory or not (Smets, 1993). In this thesis, Bayesian inference is chosen where everything is expressed as probabilities rather than for example possibilities. Just as the outcomes of Bayesian inference are only as good as the assumptions that were made on for example its probability distributions, so are the outcomes of the analyses made in this thesis only as good as this choice for Bayesian reasoning; if the reader agrees with this choice, then he or she will also agree with the outcomes, but if he or she disagrees, then he or she will not support the outcomes. As most of the other belief models are extensions on or related to Bayesian inference, the author believes that it is possible to apply the alternative methods to the same problems.

The basis for the Bayesian inference framework for data assimilation has been laid by the seminal work of Mackay (1992a, 1995). The book of Bishop (1995) deals with the same subjects. In chapter 10 of that book the Bayesian framework is explained very well, along with good examples and a thorough description of its pros and cons. Finally, Thodberg (1993) has written on the subject, and chooses a slightly different perspective on the matter which helps for a better understanding. Each of these references are highly recommended material for any interested reader. These works will also be referenced many times throughout this thesis.

1.5 The Bayesian framework for data assimilation

In this subsection the Bayesian inference framework that has been chosen as a tool for data assimilation will be described from the very beginning. The derivation below is based on the papers of Mackay (1992a, 1995) and the book of Bishop (1995). Following the order of the data assimilation steps from Figure 1.3 from top to bottom, the framework will be described.

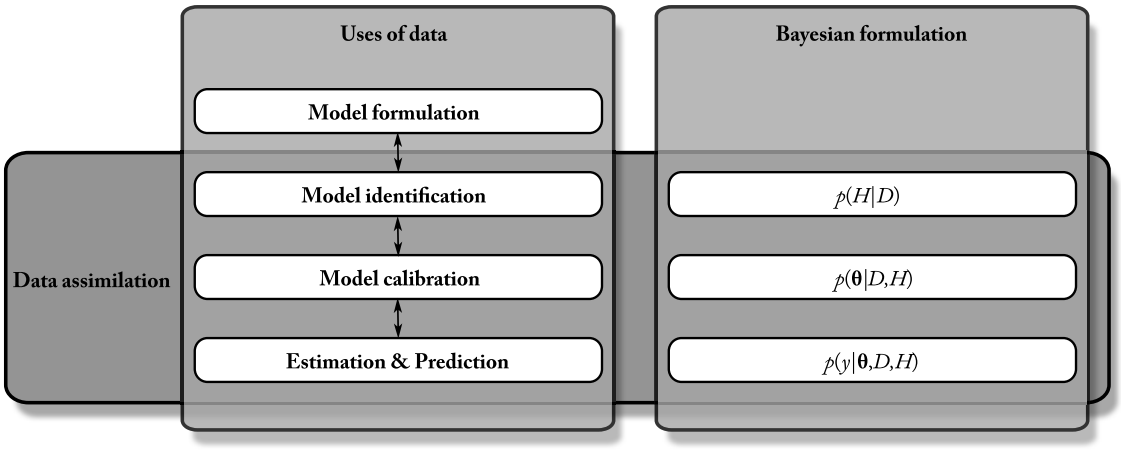


Figure 1.3: The four uses of traffic data and the process of data assimilation

Model validation

In Bayesian inference, validation equals evaluating the *probability* that the hypothesis H_q is correct. This probability is denoted by $P(H_q)$. This probability quantifies the certainty (to the user) that model q correctly describes the phenomenon under consideration, i.e. that its blueprint is perfect for the description of the phenomenon. In other words, *validating* model q equals *evaluating* $P(H_q)$.

As noted before, data can be used for validation, in which case it is an empirical validation. Define D to represent a certain data set that will be used for validation. The interest is now in finding the conditional probability $P(H_q|D)$, i.e. the probability that the assumptions underlying model q are right, given that the data set D is representative for the underlying traffic process.

Using Bayesian inference, an expression for the conditional probability $P(H_q|D)$ can be found. This probability is usually called the *posterior* probability, indicating that it is an outcome of the inference process. Bayes' theorem in this case states:

$$P(H_q|D) = \frac{P(D|H_q)P(H_q)}{P(D)} \tag{1.1}$$

In other words, the probability that the assumptions H_q that were made for model q are correct, given it has been calibrated to a dataset D is known in case the following three terms are known:

- $P(D|H_q)$ equals the probability that the data D can be produced by the model q, which is usually known as the *likelihood*.
- $P(H_q)$ equals the probability that the assumptions that were made for the model q are correct in itself, usually termed the *prior*.

- $P(D)$ equals the probability that the data itself are observed, usually termed the *normalization factor*.

Model comparison

As stated, evaluating $P(H_q|D)$ can provide an answer to how valid a model is. However, the relative probabilities of two models, for example $P(H_q|D)$ for model q and $P(H_r|D)$ for another model r, can also be used to *compare* the models. In other words, the model identification step is very similar to the validation step.

Prior to this comparison, a user may have belief that a certain model is more likely to be correct in its predictions than another, for example because of earlier experience or because of a literature study. The prior $P(H_q)$ allows for such belief to be incorporated in the choice process. However, if a modeler has no ability or wish to include prior information, then the term $P(H_q)$ can be set equal for all models under consideration in which case it can be omitted. Because the normalization factor $P(D)$ is independent of the model assumptions H_q, this term can also be omitted when comparing models. Therefore, if no prior is included, the model identification can be performed by investigating the likelihood term alone:

$$P(H_q|D) \sim P(D|H_q) \tag{1.2}$$

In this case, the likelihood term is sometimes also called *evidence* for model q.

Model calibration

As stated before, for a fair comparison of models, the best possible parameter values need to be found for each model. Define the set of parameters values of model q by the vector $\boldsymbol{\theta}_q$. If the same dataset D is used to find optimal values for these parameters, the result of the calibration procedure can be described by the conditional probability $p(\boldsymbol{\theta}_q|D, H_q)$. This posterior probability describes the probability that certain parameter values are correct, given the assumptions H_q that describe what function the parameters have, and given the data set that has been used. This posterior distribution of the parameters can also be found using Bayes' theorem:

$$p(\boldsymbol{\theta}_q|D, H_q) = \frac{p(D|\boldsymbol{\theta}_q, H_q)p(\boldsymbol{\theta}_q|H_q)}{p(D|H_q)} \tag{1.3}$$

The evidence term of (1.2) can now be recognized as the denominator of (1.3). In (1.3) the evidence represents a normalization factor. It therefore equals the integral:

$$P(D|H_q) = \int_{-\infty}^{+\infty} p(D|\boldsymbol{\theta}_q, H_q)p(\boldsymbol{\theta}_q|H_q)d\boldsymbol{\theta}_q \tag{1.4}$$

Equation (1.4) describes the interrelationship between the model validation/identification step and the calibration step; one should not take place without the other, and if one of the two posterior distributions is found, the other is found automatically too. Generally the calibration is performed first, after which the validation or comparison is performed.

Estimation & prediction

The final step of the data assimilation process is the estimation or prediction step. Define $\boldsymbol{y}_q$ to be the vector of outcomes of a model q. The prediction step is described by the conditional probability $p(\boldsymbol{y}_q|\boldsymbol{\theta}_q, D, H_q)$. Using the distributions of the parameters, the outcome of such a prediction is thus not a single value, but a distribution of values. For this last step, Bayes' rule can be used once more:

$$p(\boldsymbol{y}_q|\boldsymbol{\theta}_q, D, H_q) = \frac{p(\boldsymbol{\theta}_q|\boldsymbol{y}_q, D, H_q)p(\boldsymbol{y}_q|D, H_q)}{p(\boldsymbol{\theta}_q|D, H_q)} \tag{1.5}$$

If a likelihood function is assumed for this prediction step (a distribution of the data that is used for the prediction step), as well as a prior distribution, then the output distribution is known: the denominator of (1.5) equals the posterior of (1.3). This marks the interrelationship between calibration and estimation/prediction.

Expressing each step in the data assimilation process in probabilistic terms, a framework appears. This framework functions as a three-step procedure, where each step is interrelated with the previous step. If an expression is obtained for a posterior distribution in any of the steps, then this solution can be used to solve the step 'above' or 'below'. In Figure 1.3 these interrelationships are shown schematically. In general, the process starts with a calibration procedure, after which the validation/comparison and the estimation/prediction steps follow automatically.

As has been stated before, the goal of this thesis is to find a unified methodology for data assimilation for a wide range of traffic models. The Bayesian inference framework that has been described is hypothesized to be able to be just that. Throughout this thesis, the framework will be applied to different models that are used in traffic science, ranging from models describing the individual driving behavior to models describing traffic as a whole. In the core chapters of this thesis, the framework will therefore be put to the test.

1.6 Outline of this thesis

This thesis consists of six edited versions of papers that have been submitted to international journals and conferences and have all been peer reviewed. Two of the papers are at the time of printing this thesis still under review. At each chapter an abstract will be given such that the reader has a quick overview of the contents of that chapter. Furthermore, the reference to the original version of the journal or conference paper will be included.

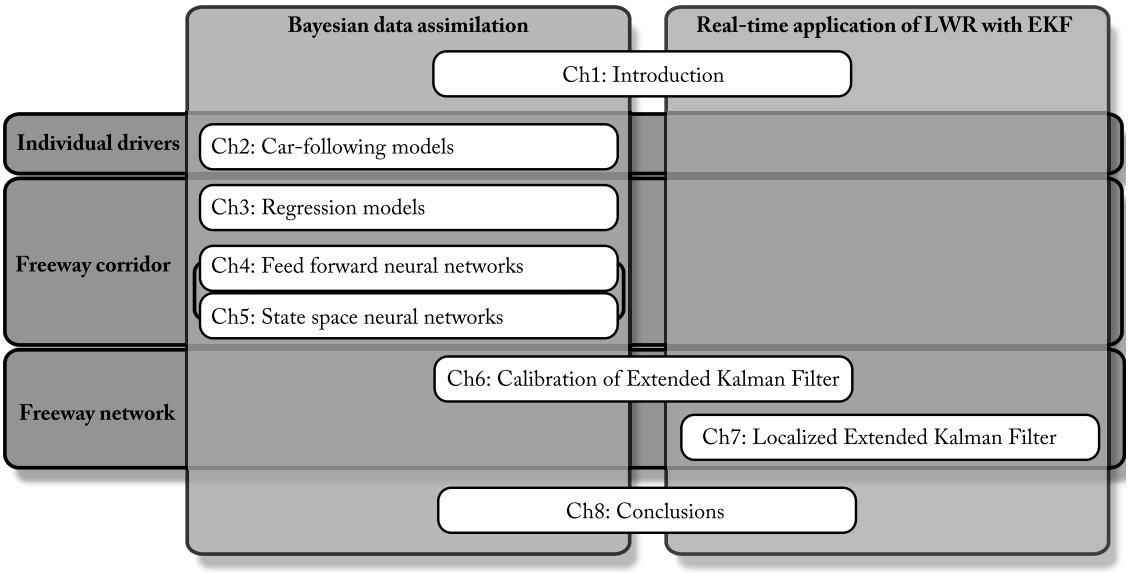


Figure 1.4: The structure of this thesis' chapters

Figure 1.4 shows the eight chapters of this thesis. Chapter 2 - Chapter 7 consist of edited versions of articles. The first five of these chapters are all based on the same Bayesian framework that has been presented in 1.5. The different applications of the framework are presented in order of increasing complexity of the models. In Chapter 2 the Bayesian framework is applied to models that predict car-following behavior that operate at the individual level. These models contain only a few parameters. Because there exist many different models for the description of car-following behavior and because of heterogeneity in car-following behavior the framework is specifically used to compare and choose between different models for each individual driver.

In Chapters 3, 4 and 5 the same framework is applied to the prediction of travel times. These travel times are predicted on a freeway corridor. In Chapter 3 the Bayesian theories are first applied to two relatively simple regression models. Next, in Chapter 4 the theories are used with Feed-Forward Neural Networks, and in Chapter 5 with the more complex State-Space Neural Networks. The papers that are the basis for 4 and 5 are based on nearly the same theories and originally contain about 30% overlap. This overlap has been removed from Chapter 5 so that Chapter 4 and 5 should be read together and have little overlap.

The last level of application is on a network-wide scale. One commonly used model to describe traffic on a network level is the macroscopic dynamic LWR model. One commonly applied tool for data assimilation with the LWR model is the Extended Kalman Filter (EKF). In this thesis two chapters deal with problems that need to be overcome before the EKF can be applied to the first order model for large scale traffic state estimation and prediction. First of all the EKF contains parameters itself that need to be

calibrated from data. Chapter 6 applies the same Bayesian framework that was applied to the problems of car-following and travel time prediction to the Extended Kalman Filter. Finally, current implementations of the EKF are too slow, a problem that is dealt with in Chapter 7. The Localized Extended Kalman Filter is designed in this chapter as an alternative method to compute the posterior distributions of the Bayesian data assimilation framework, which is shown to be much faster than current methods.

Finally, Chapter 8 describes the conclusions and synthesizes the different chapters. Also, in the same chapter recommendations for future research are proposed.

1.7 Contributions of this thesis

This section describes the contributions that this thesis has made. Two types of contributions are distinguished: scientific/methodological, and practical.

1.7.1 Scientific and methodological contributions

Scientific and methodological contributions are contributions that answer the question: "*what new knowledge has been gained by the research presented in this thesis, and what new methods have been developed in order to obtain this knowledge?*". Below, for the three types of applications to which the framework is applied, and for all applications as a whole, these contributions are summed up.

Car-following behavior

- This research has shown that the Bayesian framework for data assimilation is able to quantify inter-driver differences. It can compare any set of car-following models of any type.

- Recent studies have suggested that there may be large inter-driver differences in car-following behavior. In a case study on 500 m of Dutch highway, these inter-driver differences have indeed been confirmed and quantified.

Travel time prediction

- In this research the Bayesian 'evidence' measure has extensively been used to create committees of networks using different combination strategies, such as Winner Takes It All or the Weighted Linear Combination. Experiments have shown that both strategies lead to small improvements of the results, and that there appears to be little difference in performance between these strategies.

- A heuristic has been proposed to deal with models being calibrated by datasets of unequal size by normalizing the likelihood.
- As the evidence is a predictor of the true generalization ability, in this research it is used for the first time as a stopping criterion during training of neural networks. In the thesis it is shown that this early stopping reduces computation time and at the cost of only a small loss in accuracy.
- Experiments have revealed that the evidence is not a perfect predictor of the true generalization ability, because of imperfections of the models themselves, noise in the data and approximations that need to be made to quantify the evidence.
- Experiments have nonetheless revealed that the use of the evidence to form a committee generally leads to an improvement of accuracy of the predictions and to an improvement of accuracy of error bars compared to the performance of all individual models.
- In this thesis the exact Jacobian and Hessian for a recurrent neural network have been derived, that allow for more accurate training of recurrent neural networks using gradient-based methods, although at the cost of much higher computation times.

Network-wide state estimation

- Using static parameters of the Extended Kalman Filter, this thesis shows that there is a clear optimum in the parameter settings, where a shift of the settings to one side causes the corrections to be too weak and the data not to be used to its full potential, while a shift of the settings to the other side of the optimum leads to too much correction where noise is copied into the model.
- In this thesis a new method is proposed to set the parameters of an Extended Kalman Filter dynamically. This method is derived in the same way as the Extended Kalman Filter was derived itself. In one experiment, the dynamic parameter settings achieve approximately the same level of accuracy as the optimal static settings, independent of the starting point that was used.
- Experiments have revealed that the traditional Global Extended Kalman Filter (G-EKF) makes many negligible corrections to the traffic state as the cross-correlation of the states at two locations in the traffic network that are far apart are generally almost zero.
- Experiments have revealed that with networks larger then a few hundred (measured) cells, the G-EKF becomes too slow to perform in real-time on a normal PC.

- In this thesis a new methodology is proposed termed the Localized Extended Kalman Filter (L-EKF) that approximates the G-EKF by making many sequential corrections to the traffic state, where each sequential correction only corrects the traffic state in the vicinity of a measurement (termed the *radius* of the filter). Experiments have empirically validated that the L-EKF achieves the same level of accuracy with much lower computation times, and have shown that the L-EKF scales much more beneficial with the size of the network and with the number of measurements.

- Experiments have revealed that an increasing radius of the L-EKF leads to a relatively small increase in computation time, so that the radius can safely taken quite large. This leads to the level of accuracy of the L-EKF being equal to that of the G-EKF.

All applications combined

- This research shows that the Bayesian framework is applicable to a variety of problems in the field of road traffic.

- In each application, it has been shown that the framework has benefits that are specific to the problem at hand, such as the quantification of inter-driver differences in car-following modeling.

1.7.2 Practical contributions

Practical contributions are contributions that answer the question: "*what can be done based on the research presented in this thesis that couldn't be done before?*".

Car-following behavior

- Using the Bayesian data assimilation framework this thesis has paved the road for a heterogeneous microscopic simulation, where multiple car-following models are used in a single simulation environment. The framework can be used to quantify the distribution of optimal[3] models for a given group of drivers.

- This thesis shows that the Bayesian framework for data assimilation can be used to find the optimal car-following model(s) for a single driver, so that for that driver the vehicle position can be accurately predicted with error bars, which can be useful for vehicle-to-vehicle or vehicle-to-roadside infrastructures.

[3] What is optimal depends on the error function that the user chooses.

Travel time prediction

- In this contribution it has been shown that the Bayesian evidence can be used to choose between a set of available models. Therefore, users no longer have to choose based on experience or gut-feeling, but based on a numerical measure that expresses the user's belief in the model.
- It has been shown that instead of choosing between models, the evidence can also be used to form a committee of models so that multiple models' predictions are made in parallel and are combined into one single prediction.
- This research shows that the uncertainty due to the simplifications made by the models, due to noise in the data and possibly due to disagreement between different models can be quantified in the form of error bars on the predictions (prediction intervals).

Network-wide state estimation

- Using the L-EKF that has been developed in this thesis, large scale networks can be simulated on a single computer. This allows for accurate, possibly nation-wide state estimation and prediction.
- In this thesis a method is developed to automize the process of calibrating the parameters of the Extended Kalman Filter.

All applications combined

- In this research the applicability of the same Bayesian framework for data assimilation is shown for a large variety of models describing a myriad of phenomena observed in road traffic, such as car-following models, travel time prediction and network-wide traffic state estimation.

Chapter 2

Bayesian calibration and comparison of car-following models

This chapter is an edited version of van Hinsbergen, C. P. I., van Lint, J. W. C., Hoogendoorn, S., and van Zuylen, H. J. (2010f). A unified framework for calibration and comparison of car-following models. Submitted for publication to Transportmetrica.

Recent research has revealed that there exists large heterogeneity in car-following behavior such that different car-following models best describe different drivers' behavior. A literature review reveals that current approaches to calibrate and compare different models for one driver do not take the complexity of the model into account or are only able to compare a specific set of models. This contribution applies Bayesian techniques to the calibration of car-following model. The resulting evidence measure can be used to quantitatively assess any set of models and describes how well different models explain the car-following behavior of a single driver. When considered over multiple drivers the evidence can be used to describe the heterogeneity of the driving population. In a test case on actual data the Bayesian evidence indeed reveals heterogeneity and it is shown how these differences can quantitatively be assessed with the Bayesian framework.

2.1 Introduction

The longitudinal driver behavior of drivers in a traffic stream determines for a large part the dynamics of the flow of traffic. This important role of longitudinal driver behavior has resulted in a multitude of mathematical models to predict the longitudinal driving behavior of individual drivers, such as the CHM model (Chandler et al., 1958), the IDM model (Treiber et al., 2000), the OVM model (Bando et al., 1995) and the models proposed by Helly (1959), Bexelius (1968), Gipps (1981), Addison and Low (1998), Lenz et al. (1999) and Tampère (2004). Brackstone and McDonald (1999) present a historical review of these and other car-following models.

In recent microscopic traffic modeling research, a number of studies have revealed that there are large inter-driver differences in car-following behavior, such that different car-following models may apply to different drivers (Brockfeld et al., 2004; Ossen et al., 2006; Hoogendoorn et al., 2007a). Additionally, intra-driver differences (the fact that individual drivers may change their behavior over the data collection period) can cause some car-following models to produce erroneous predictions during certain episodes of the driver's car-following behavior (Hoogendoorn and Ossen, 2005; Hamdar et al., 2008). The effects of such heterogeneity of car-following behavior on the macroscopic properties of traffic are important (Hoogendoorn et al., 2007b). One possible solution is to model traffic heterogeneously, i.e. using multiple car-following models in one simulation. To achieve such a heterogeneous microscopic simulation, from all available models the most likely best-performing models need to be identified. Ideally, this identification process should be performed based on data, that is also used for calibration of the models. This contribution describes both the calibration and identification process of car-following models.

2.1.1 State of the art in model calibration and comparison

First, an extensive literature study has been carried out to investigate current calibration and model selection methods. Four methods have been identified: using default parameters, using the calibration error, using the validation error or using the Likelihood-Ratio Test. Below, each of these methods is described.

Default parameter settings

One study was found where the default manufacturer parameters are used to evaluate the performance of different models (Panway and Dia, 2005), even though they recognize the importance of the parameter values on the performance of the model. It is clear that these default parameters cannot be used in every case, and that data should be used to calibrate and compare the models.

Calibration error approach

The second and most commonly used approach is to select models based on the outcomes of the calibration procedure (Aycin and Benekohal, 1999; Chakroborty and Kikuchi, 1999; Rakha and Crowther, 2003; Ranjitkar et al., 2004, 2005; Ossen and Hoogendoorn, 2005; Ossen et al., 2006; Punzo and Tripodi, 2007; Kesting and Treiber, 2008). In most cases these studies conclude that models containing more parameters perform better than simpler models. However, the calibration error approach does not take the model complexity into account. In many cases more complex models will not make better predictions, due to 'overfitting' of the complex models. The models with many parameters then start predicting the noise rather than the underlying system. The use of calibration error as a basis to select models should therefore be rejected.

Validation error approach

Instead of using the same data set to calibrate and compare the models, also a separate data set can be used to make a selection from a set of models. The validation set approach is a theoretically sound way to compare models in case the validation data set is representative for the phenomenon that one is trying to model. Interestingly, different studies confirm that more complex models do not always perform better (Wu et al., 2003; Brockfeld et al., 2003, 2004; Punzo and Simonelli, 2005). For example, in Brockfeld et al. (2003) the model with 20 parameters performs worse on the validation set compared to simpler models, a confirmation of the overfitting problem of overly complex car-following models.

Unfortunately, this approach requires two data sets to be available for one single driver. Such data can usually only be collected under controlled conditions, where the same drivers are asked in an experimental setup to perform the car-following task. This has two major drawbacks: the results of experiments in controlled conditions may not always be portable to a real-life' situation, and usually only a small data set is available because of the expenses that have to be made to equip vehicles and to attract participants. More data may be collected monitoring the regular' traffic system (using for example cameras or a helicopter), but in those cases usually the data set of each single driver is too small to be split in half. Therefore, a method that allows for all data to be used for calibration, while still preventing overfitted models to be selected, should in most cases be preferred.

Likelihood-ratio test

The Likelihood-Ratio-Test (LRT) is a method that allows all data to be used for calibration and model selection in parallel, while still preventing overfitted models to be selected (Hoogendoorn et al., 2006, 2007a,b). More complex models receive a penalty, while models that fit well on data are promoted. This balances the goodness-of-fit to the data with the model complexity.

However, this method is only valid when used to compare *hierarchically nested* models. This means that the simple model must be a special case of the more complex model by setting one or multiple parameters to zero. Therefore, Hoogendoorn et al. (2006) formulated first a general equation in which several car-following models could be fitted. However, as not all car-following models that have been developed over the years may be fitted into one general equation, this will not be possible when a modeler is interested in trying many different car-following models.

Therefore, in this chapter a novel method is proposed to calibrate and compare car-following models. It is an extension to the LRT method, and allows for the comparison of any car-following model.

2.1.2 Structure of this chapter

In the Methodology section a Bayesian approach to calibrate and compare car-following models is developed, after which it is applied to two relatively simple car-following models in order to show its workings: the CHM model and the linear Helly model. Next, the result of the Bayesian 'evidence' as a selection mechanism is shown, after which a discussion, a conclusion and recommendations are presented.

The Bayesian approach is a generalization of the LRT approach (Hoogendoorn et al., 2007a). Prior probabilities are transformed into posterior probabilities for each parameter in the car-following model, for which Bayes' rule is used. In the Methodology section the exact formulation of this new method for calibration and model selection will be presented and it will be shown that this approach has several advantages over existing mechanisms: (1) the most important feature is that it leads to a probabilistic approach to compare different models on the basis of posterior distributions of their parameters. This allows a modeler to select the model that most probably best describes a certain driver's behavior, taking into account both the calibration error as well as the model complexity. The main contribution of the Bayesian approach compared to the LRT approach is that any model can be used; (2) just as with the LRT approach prior information can be included when calibrating the parameters of car-following models to rule out unrealistic estimation results due to the fact that too little information is present on certain parameters within data; (3) the approach can be used to combine the predictions of several models in a so-called committee or ensemble of models in which different models predict the behavior of one single driver, which may lead to a decrease in the error due to intra-driver differences; (4) error bars (prediction intervals) can be constructed on the predictions of the car-following models.

In this chapter the focus will be on the methodology. To remain focussed, the workings of the proposed procedure are then demonstrated using two relatively simple models and using some simplifications. Although these simplifications do influence the results, they do not prevent the illustration of the benefits of the Bayesian framework itself.

2.2 Methodology

2.2.1 Bayesian Inference: from prior to posterior

For the Bayesian analysis, the interest is in finding the posterior probability density function of a parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)^T$ which contains all N parameters of a car-following model under investigation after having used some data set D for calibration. This data set contains for example positions (lateral and longitudinal) and speeds of different vehicles, from which car-following models can be calibrated. This posterior probability is denoted by $p(\boldsymbol{\theta}|D)$, e.g. the probability density function of the parameters $\boldsymbol{\theta}$ given the data set D. Bayes' rule can be applied to find an expression for this posterior:

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \tag{2.1}$$

where $p(D|\boldsymbol{\theta})$ represents the distribution of noise on the data and corresponds to the likelihood function, $p(\boldsymbol{\theta})$ is a prior probability of the parameters, which represents prior knowledge of possible values for each parameter in our model, and where $p(D)$ is a normalization factor.

Now define the prior probability as a multivariate Gaussian with mean $\overline{\boldsymbol{\theta}}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$p(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}})\right) \tag{2.2}$$

where N equals the number of parameters of the model. A Gaussian shape is chosen in this study because it simplifies the calculations and enables analytical expressions for the posterior distribution of the parameters. Note that this assumption can be relaxed and other distributions are possible.

If it is assumed that the noise of the data is Gaussian distributed as well with mean zero and standard deviation σ_l, the likelihood function $p(D|\boldsymbol{\theta})$ can be defined as a uni-variate Gaussian (Hoogendoorn et al., 2007b):

$$p(D|\boldsymbol{\theta}) = \frac{1}{(\sigma_l^2 2\pi)^{K/2}} \exp\left(-\frac{1}{2\sigma_l^2}\sum_{k=1}^{K}\left(v_{pred}(k,\boldsymbol{\theta}) - v_{obs}(k)\right)^2\right) \tag{2.3}$$

where $v_{pred}(k, \boldsymbol{\theta})$ is the predicted vehicle speed at time instant k with the parameter set $\boldsymbol{\theta}$, $v_{obs}(k)$ is the observed (measured) vehicle speed at time instant k, and where K equals the number of observations of vehicle speed and position. Note that in this study the models are calibrated on speeds alone, but that other likelihood functions which incorporate for example the predicted positions of the vehicles are also possible.

2.2.2 Description of posterior distribution of parameters

Substituting (2.2) and (2.3) into (2.1) results in an expression for the posterior distribution of the parameters:

$$p(\boldsymbol{\theta}|D) = \frac{1}{Z_p \sigma_l^K |\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2\sigma_l^2}\sum_{k=1}^{K}\left(v_{pred}(k,\boldsymbol{\theta}) - v_{obs}(k)\right)^2\right) \times \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\overline{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\overline{\boldsymbol{\theta}})\right) \tag{2.4}$$

where Z_p is a constant that originates from $p(D)$ and the '2π-constants' in (2.2) and (2.3). This posterior distribution of the parameters can be described by the most probable parameter vector $\boldsymbol{\theta}^{MP}$ (the maximum of the posterior), and its covariance matrix $\boldsymbol{\Theta}$ (the width of the posterior), with the knowledge that it has a Gaussian shape.

The maximum of the posterior is denoted by the vector $\boldsymbol{\theta}^{MP}$, and can be found by maximizing the logarithm of (2.4):

$$\begin{aligned}\boldsymbol{\theta}^{MP} &= \arg\max_{\boldsymbol{\theta}} \ln\left(p(\boldsymbol{\theta}|D)\right) \\ &= \arg\max_{\boldsymbol{\theta}} -E(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta})\end{aligned} \tag{2.5}$$

where $E(\boldsymbol{\theta})$ is defined as

$$E(\boldsymbol{\theta}) = K\ln(\sigma_l) + E_p(\boldsymbol{\theta}) + E_l(\boldsymbol{\theta}) \tag{2.6}$$

with E_l and E_p defined by:

$$E_p(\boldsymbol{\theta}) = \frac{1}{2}\left(\boldsymbol{\theta}-\overline{\boldsymbol{\theta}}\right)^T\boldsymbol{\Theta}^{-1}\left(\boldsymbol{\theta}-\overline{\boldsymbol{\theta}}\right) \tag{2.7}$$

$$E_l(\boldsymbol{\theta}) = \frac{1}{2\sigma_l^2}\sum_{k=1}^{K}\left(v_{pred}(k,\boldsymbol{\theta}) - v_{obs}(k)\right)^2 \tag{2.8}$$

Notice that in (2.6) the expressions resulting from Z_p and $|\boldsymbol{\Sigma}|^{1/2}$ have been omitted, as these do not influence the solution of (2.5) and becomes zero for the derivatives that are defined next. For the minimization of (2.6) (so to find $\boldsymbol{\theta}^{MP}$), there is the condition (Hoogendoorn et al., 2007a):

$$\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\theta}-\overline{\boldsymbol{\theta}}\right) + \nabla_{\boldsymbol{\theta}} E_l(\boldsymbol{\theta}) = 0 \tag{2.9}$$

which needs to be solved for the model under consideration.

The covariance matrix $\boldsymbol{\Theta}$ of the posterior distribution (not to be confused with the covariance matrix $\boldsymbol{\Sigma}$ of the prior) can be approximated using:

$$\boldsymbol{\Theta}(\boldsymbol{\theta}^{MP}) = -\left(\boldsymbol{A}(\boldsymbol{\theta}^{MP})\right)^{-1} \tag{2.10}$$

where $\boldsymbol{A}(\boldsymbol{\theta})$ is the Hessian, given by:

$$\boldsymbol{A}(\boldsymbol{\theta}) = \nabla^2_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{-1} + \nabla^2_{\boldsymbol{\theta}} E_l(\boldsymbol{\theta}) \tag{2.11}$$

Finally, for the description of the posterior a value for the standard deviation of the likelihood function σ_l needs to be found, for which the derivative $\partial E/\partial \sigma_l$ is set to zero. This leads to

$$\sigma_l^2 = \frac{1}{K} \sum_{k=1}^{K} \left(v_{pred}(k, \boldsymbol{\theta}) - v_{obs}(k)\right)^2 \tag{2.12}$$

2.2.3 Bayesian framework for model comparison

Consider a certain car-following model m with a set of assumptions H_m, and another model n with a different set of assumptions H_n. To compare these two models in how well they describe the car-following behavior of a certain driver, the posterior probability of a model $q \in (m, n)$ as a whole after it has been calibrated with data D for this driver, which is denoted by $P(H_q|D)$, can be derived by again applying Bayes' rule:

$$P(H_q|D) = \frac{p(D|H_q)P(H_q)}{p(D)} \tag{2.13}$$

The term $P(H_q)$ represents the prior probability of the model q. If a priori there is no preference of one type of model over the other (so there is belief that the assumptions H_m are as likely as H_n), then the prior $P(H_q)$ is equal for all q. As the denominator of (2.13) is independent of the models H_q, the posterior probabilities of the models m and n can in that case be compared by only investigating the term $p(D|H_q)$, which is termed the *evidence* for the model q (Mackay, 1995):

$$P(H_q|D) \sim p(D|H_q) \tag{2.14}$$

This evidence can be recognized as the denominator of (2.1) if the conditional dependence on the model assumptions H_q is made explicit. The expressions used for deriving the *posterior distribution for the parameters* can therefore be used to derive expressions for the *evidence for the entire model*. From (2.1) the evidence can be written in the form

$$p(D|H_q) = \int p(D|\boldsymbol{\theta}, H_q) p(\boldsymbol{\theta}|H_q) d\boldsymbol{\theta} \tag{2.15}$$

Because this term would require integration (marginalization) over the entire parameter space, calculating it analytically is only possible in case of very simple models, and even then requires elaborate calculations. Although a numerical approximation could be used, in this study an analytical approximation is chosen to be able to analytically describe the evidence. Assuming that the posterior distribution is sharply peaked around its maximum, the evidence is approximated as the value at this maximum times the width of the peak, which in the multivariate case leads to the expression (Mackay, 1995):

$$p(D|H_q) \approx p(D|\boldsymbol{\theta}^{MP}, H_q) \times \frac{p(\boldsymbol{\theta}^{MP}|H_q)}{\sqrt{\det\left(\boldsymbol{A}(\boldsymbol{\theta}^{MP})/2\pi\right)}} \tag{2.16}$$

Together with (2.2), (2.3) and (2.11) a solution (approximation) is now found for the evidence. Note that values for the prior covariance matrix $\boldsymbol{\Sigma}$ and the prior mean $\overline{\boldsymbol{\theta}}$ are needed for this; the way the prior is defined will be treated later.

The evidence of (2.16) can be interpreted as consisting of two elements:

$$\text{Evidence} = \text{Best-fit likelihood} \times \text{Occam factor} \tag{2.17}$$

A higher best fit likelihood favors models that can explain the data well, i.e. that have a low prediction error $\sum(v_{pred} - v_{obs})^2$. However, if only this would be investigated the overfitting problem would occur as when the calibration error is used for model selection. Therefore, the model's performance is penalized by the Occam factor, which is always smaller than 1 and is named after Occam's Razor (Blumer et al., 1987). A model that has more parameters, so which is more complicated, has a lower Occam factor and therefore receives lower evidence. The evidence thus naturally reflects the trade-off between a good fit and overfitting. Extensive literature is available on the importance of this trade-off and other features of the evidence (Thodberg, 1993; Mackay, 1995; Bishop, 1995; Sivia, 1996; Mackay, 2003; van Hinsbergen et al., 2008a,d).

In the remainder of this contribution, the evidence is used to rank different car-following models for individual drivers. This is achieved by determining the evidence after the posterior distribution of its parameters has been found, after which a conclusion can be drawn to which model probably describes which driver's behavior best. The Bayesian analysis will be applied here to two simple car-following models, for which the evidence can be derived analytically.

2.2.4 Evidence for CHM model

To illustrate the derivation of the evidence for a car-following model, consider the CHM model (Chandler et al., 1958). This stimulus-response model describes the delayed accel-

eration of a vehicle as a function of the relative speed with respect to its leading vehicle:

$$a(t+\tau, \boldsymbol{\theta}) = \gamma \Delta v(t) \tag{2.18}$$

where $a(t+\tau, \boldsymbol{\theta})$ is the acceleration of the following vehicle at time $t+\tau$ given the parameter set $\boldsymbol{\theta}$ and $\Delta v(t)$ the speed difference between the leader and the follower at time t. In this study, one-step-ahead predictions are made, where the observed speeds of the follower and its leader in the previous time step are used in the calculations. An explicit time stepping scheme is used to solve the model, resulting in the following numerical scheme for the speed at time t:

$$v_{pred}(t, \boldsymbol{\theta}) = v_{obs}(t - \Delta t) + a(t - \Delta t, \boldsymbol{\theta}) \Delta t \tag{2.19}$$

with $v_{obs}(t-\Delta t)$ the observed speed at time $t-\Delta t$, and Δt the size of the time step which should be sufficiently small. The acceleration is in this scheme determined by:

$$a(t - \Delta t, \boldsymbol{\theta}) = \gamma \Delta v_{obs}(t - \Delta t - \tau) \tag{2.20}$$

The model has only one parameter that needs to be calibrated with data:

$$\gamma \qquad \text{response parameter (1/s)} \tag{2.21}$$

For this model, the parameter vector is denoted as $\boldsymbol{\theta} = \gamma$. For the sake of this example, the reaction time τ is chosen to be a constant with a value of $\tau = 1s$, and not as a parameter. This heavy simplification is made to keep the discussion focussed on illustrating the Bayesian framework and its benefits; the (complex) derivation of the derivatives to τ is not required to show the workings of the framework. In a real world application, the reaction time τ does need to be calibrated with data, and derivatives for it would be needed.

To analytically derive the evidence for the CHM model, first the gradient of (2.9) needs to be computed:

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \gamma} = \frac{1}{\sigma_{prior}^2} \left(\gamma - \overline{\gamma}_{prior} \right) + \frac{\Delta t}{\sigma_l^2} \sum_{k=1}^{K} v_q (v_p + \gamma v_q \Delta t - v_s) \tag{2.22}$$

where $\overline{\gamma}_{prior}$ is the mean of the prior distribution and σ_{prior}^2 is the prior variance (previously $\overline{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}$, but now for the one-dimensional case because the model only contains one parameter), and where v_q, v_p and v_s are all observations at different time steps, defined

by:

$$v_s = v_{obs}(k) \tag{2.23}$$
$$v_p = v_{obs}(k - \Delta t) \tag{2.24}$$
$$v_q = v_{obs}(k - \Delta t - \tau) \tag{2.25}$$

The Hessian of (2.11) is given by:

$$\frac{\partial^2 E_l(\boldsymbol{\theta})}{\partial \gamma^2} = \frac{1}{\sigma^2_{prior}} + \frac{\Delta t^2}{\sigma_l^2} \sum_{k=1}^{K} v_q^2 \tag{2.26}$$

To calculate the evidence, the most probable parameter γ^{MP} is required, for which (2.9) needs to be solved. This is done numerically using standard Matlab optimization tools as the analytical solution becomes rather complex. Then σ_l^{MP} is calculated using (2.12), γ^{MP} and σ_l^{MP} are substituted in (2.2), (2.3) and (2.11), and the resulting equations into (2.16) together with $\overline{\gamma}_{prior}$ and σ^2_{prior}, resulting in the evidence for the model.

2.2.5 Evidence for Helly model

As a second example of the derivation of the evidence for a car-following model, consider the Helly model (Helly, 1959), another stimulus-response model with a higher complexity. It is defined by:

$$a(t + \tau, \boldsymbol{\theta}) = \alpha \Delta v(t) + \beta \left(\Delta x(t) - \Delta x^{des}(v(t)) \right) \tag{2.27}$$
$$\Delta x^{des}(v) = x_0 + Tv \tag{2.28}$$

where $a(t + \tau, \boldsymbol{\theta})$ is the acceleration of the following vehicle at time $t + \tau$ given the parameter set $\boldsymbol{\theta}$, $\Delta v(t)$ the speed difference between the leader and the follower at time t, $\Delta x(t)$ the distance headway between the leader and the follower at time t and $\Delta x^{des}(v(t))$ the desired distance headway of the follower when driving at speed $v(t)$, the speed of the follower at time t. Again, one-step-ahead predictions are made, where the observed speeds and distances of the follower and its leader in the previous time step are used in the calculations. The same numerical scheme as in (2.19) is used, but with the acceleration now determined by:

$$a(t - \Delta t, \boldsymbol{\theta}) = \alpha \Delta v_{obs}(\kappa) + \beta \left(\Delta x_{obs}(\kappa) - \Delta x^{des}(v_{obs}(\kappa)) \right) \tag{2.29}$$
$$\kappa = t - \Delta t - \tau \tag{2.30}$$

The model has the following four parameters that need to be estimated from data:

α	response parameter $(1/s)$	(2.31)
β	response parameter $(1/s^2)$	(2.32)
x_0	stopping distance (m)	(2.33)
T	minimum time headway (s)	(2.34)

For this model, the parameter vector is denoted as $\boldsymbol{\theta} = (\alpha, \beta, x_0, T)$. Again, as with the CHM model, the reaction time is chosen to be a constant with a value of $\tau = 1s$, and not as a parameter.

The gradient and Hessian for the Helly model are derived analytically again, the result of which will be omitted here as it involves quite lengthy equations. The most probable parameter vector $\boldsymbol{\theta}^{MP}$ is estimated numerically using standard numerical tools in the Matlab software package, as the condition (2.9) is not easily solvable analytically. Then, the same procedure as with the CHM model is used to calculate the evidence.

Prior distribution for CHM model parameter

The original work of Chandler, Herman and Montroll showed high variations between subjects for the constant γ, between $0.17s^{-1}$ and $0.74s^{-1}$ with a mean of $0.37s^{-1}$ (Chandler et al., 1958; Brackstone and McDonald, 1999). A benchmarking study by Ossen et al. (2006) conducted on a Dutch motorway using helicopter data showed the distribution of parameter values for the CHM model as shown in Figure 2.1, more or less confirming the spread of the original study of Chandler, Herman and Montroll. From the results of these studies, a prior distribution $N(\overline{\gamma}_{prior}, \sigma^2_{prior}) = N(0.3, 0.04)$ is chosen.

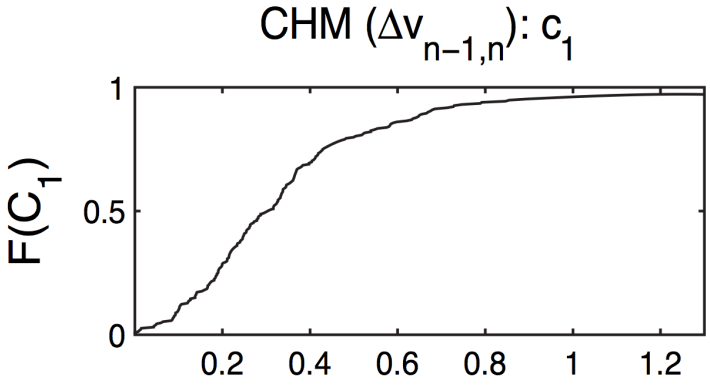


Figure 2.1: The cumulative distribution of the parameter γ (denoted by c_1 in the figure) given by Ossen et al. (2006)

Prior distribution for Helly model parameters

Helly in his original work (Helly, 1959) estimated the mean parameter values $\alpha = 0.5s^{-1}, \beta = 0.125s^{-2}, x_0 = 20m$ and $T = 1s$. The earlier mentioned benchmarking study (Ossen et al., 2006) only presents CDFs for the parameters α and β as shown in Figure 2.2, and not for x_0 and T. Taking both these studies into account, the following prior mean and covariance matrix are chosen (not taking into account covariance between the different parameter):

$$\begin{aligned} \overline{\boldsymbol{\theta}} &= (\overline{\alpha}, \overline{\beta}, \overline{x}_0, \overline{T})^T = (0.25, 0.075, 20, 1)^T \\ \boldsymbol{\Sigma} &= \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 40.0 & 0 \\ 0 & 0 & 0 & 0.4 \end{bmatrix} \end{aligned} \tag{2.35}$$

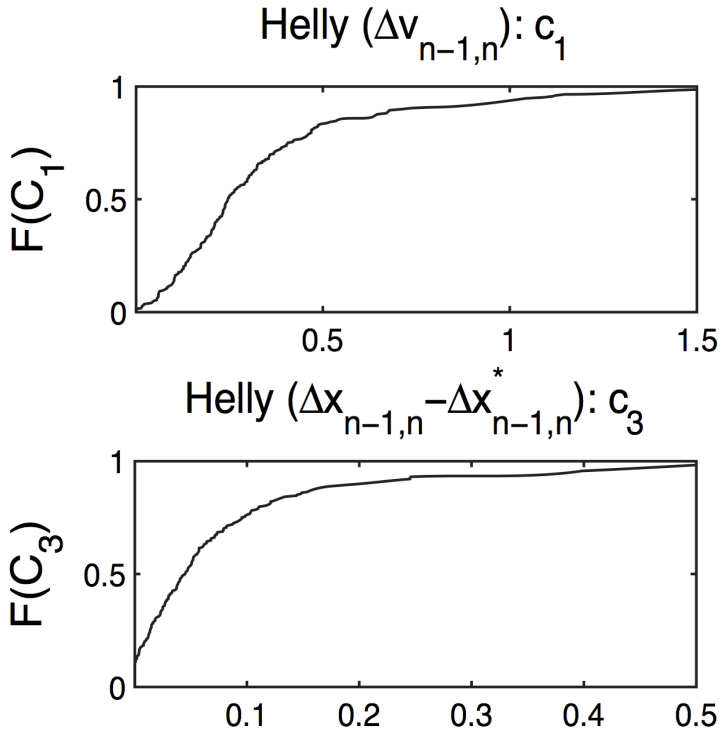


Figure 2.2: The cumulative distribution functions of α (c_1) and β (c_3) given by Ossen et al. (2006)

Large variances are taken for x_0 and T to reflect the fact that there is no reference study available for estimates of the variance of these two parameters. However, the variances are chosen in such a way that it is ensured that most of the mass (at least 95%) of the

CDF is for values > 0, which is sensible in the light of the physical meaning of these two parameters.

2.3 Experiment

To illustrate the workings of the Bayesian evidence the two models described in the methodology section are applied to a vehicle trajectory data set of the A2 motorway in the Netherlands, near the city of Utrecht, which was collected using helicopter data (Hoogendoorn et al., 2003). The traffic state at the data collection period was congested so that the drivers were mainly in car-following mode. The data covers approximately $500m$ of motorway stretch; the data interval is $0.1s$.

A selection was made in the dataset of drivers who were following one leader without any lane changes of either follower or leader (229 drivers in total). The posterior distributions of the parameters of the two models were then found after which the evidence was calculated for each model for each driver. Note that the natural logarithm of the evidence is used, as the denominator of (2.3) is taken to the power of K, which means that the likelihood becomes very large if $\sigma_l < 1$ and very small if $\sigma_l > 1$ in case $K \gg 1$. Given that the number of measurements and predictions is in the order of 100 to 400 for each driver, the log of the evidence is used to prevent numerical errors in the computations.

2.4 Results

Figure 2.3 shows the log evidence for the two models for 9 of the 229 drivers. As can be seen, the evidence assigns a preference over different models for different drivers: for some, the Helly model is preferred, while for others the CHM model is preferred. To illustrate why this happens, consider drivers 47 and 48. Figure 2.4 shows the actual speeds versus the predicted speeds that followed from the calibration by both the CHM model and the Helly model for these two drivers. The figure nicely illustrates the mechanism of the Bayesian framework. After calibration for driver 47, the Root Mean Square Error (RMSE) of the estimated versus the measured speed was $0.140m/s$ for the CHM model, while it was $0.085m/s$ for the Helly model. The larger calibration error of the CHM model is dominant over its lower complexity. The log of the evidence for the CHM model was therefore lower this case, 11.4 versus 44.0 for the Helly model.

In the case of driver 48, the two models perform almost equally well. Both the CHM model and the Helly model had an RMSE of $0.096m/s$. The evidence in this case prefers the simpler model over the more complex model, and assigns a log evidence of 118.5 to the CHM model and 115.0 to the Helly model. The Helly model in this case is punished for its higher complexity: the extra parameters do not lead to a lower calibration error.

The Bayesian evidence is a tool for ranking models for each individual driver. The

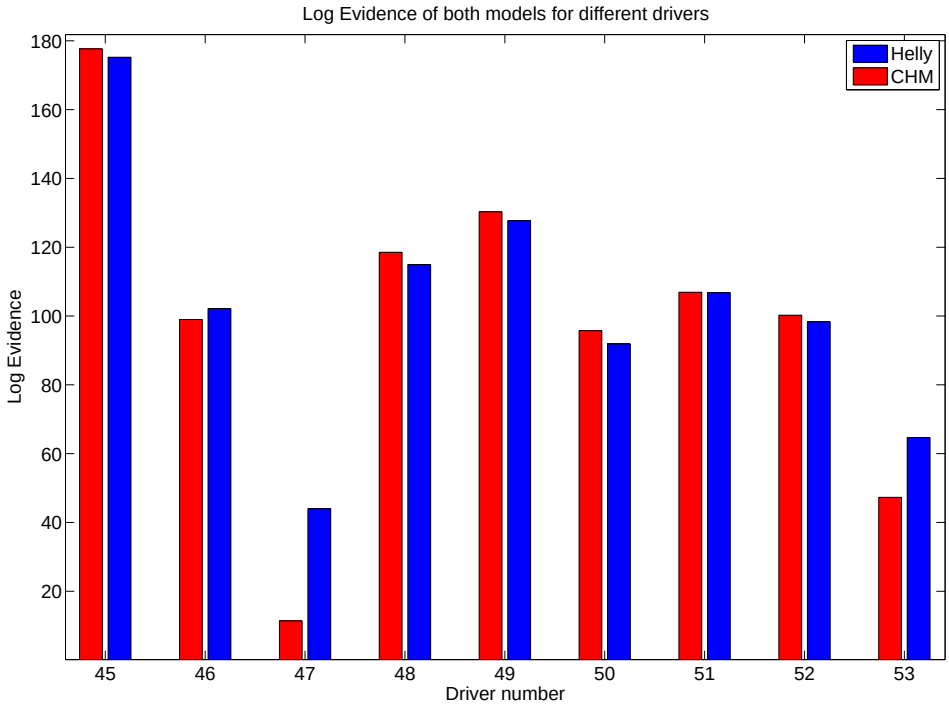


Figure 2.3: The natural logarithm of the evidence for the two models for 9 of the 229 drivers

posterior probability of the entire model can also be expressed. Two assumptions then need to be made. First, equal priors are assumed for both models $q \in (chm, helly)$, $P(H_{chm}) = P(H_{helly}) = P(H)$. Second, a closed world is assumed, i.e. the CHM and Helly models are considered to be the only two possible models for explaining car-following behavior, such that $P(\emptyset) = 0$. The normalization factor $P(D)$ in (2.13) can then be expressed as:

$$P(D) = P(H)\left(P(D|H_{chm}) + P(D|H_{helly})\right) \tag{2.36}$$

The probability of one model q then equals:

$$P(H_q|D) = \frac{P(D|H_q)}{P(D|H_{chm}) + P(D|H_{helly})} \tag{2.37}$$

Aggregated over all drivers, this mechanism can be used to see how well models perform relative to the other models for a group of drivers. By taking the mean of (2.37) over all individual drivers, an expression is found for the probability of a model compared to the probability of both used models. In Table 2.1 such aggregate results are presented. Note

that the closed world assumption is not very realistic in this case, because from literature it is known that there exist many more car-following models that are not used in this study. As Mackay (1992a) notes, inference is normally open ended: in the scientific process new models will be tested or developed to account for the data that have been gathered. Nevertheless, the closed world assumption here aids to express posterior probabilities for each of the two models, which are meaningful in comparison to each other. For a more detailed discussion on this assumption, see 1.4.

Table 2.1: Probabilities of the CHM and Helly model averaged over all 229 drivers

Model	$P(H_q\|D)$
CHM	31.0%
Helly	69.0%

2.5 Discussion and conclusion

The Bayesian evidence that has been developed for the car-following models in this contribution is shown to be useful as a tool for quantitatively analyzing inter-driver differences. It can be used to find a distribution of model parameters, as well as to compare models based on how well they fit and the relative complexity of the models.

As can be seen from the experiment the inter-driver differences are confirmed: for some of the drivers the CHM model suffices and the additional parameters of the Helly model do not contribute to explaining their car-following behavior, in which case the Helly model is penalized for its higher complexity. For others the additional parameters do lead to a better explanation of the car-following behavior in which case the Helly model is rewarded for this. The Bayesian evidence thus acts as a natural selection mechanism when choosing between different car-following models. Note that for the two models chosen in this study the Likelihood Ratio Test could also be applied, but that the evidence is favorable over the LRT in the general case, because the evidence can be used for any model, while the LRT can only be applied to hierarchically nested models.

The evidence, when normalized, represents a a probability of a certain model q's probability to describe one driver i's behavior, as expressed in (2.37). If this probability is averaged over all drivers i of a certain dataset, an approximation of the best performing models for an entire population of drivers can be made, as is indicated in Table 2.1. Such probabilities can serve as a basis for a heterogeneous microscopic simulation: first a model is drawn based on the posterior probabilities of the models, after which parameters are drawn from the posterior distribution of the parameters of the drawn model. The trajectories of the car are then predicted with the chosen model with the chosen parameter set. Future study will need to reveal if such a heterogeneous microsimulation better describes reality.

Other benefits of the Bayesian approach that have not been illustrated in this study are the possibility to use the evidence to create a committee, and to construct prediction intervals. A committee may improve the description of individual behavior (because it may deal with intra-driver differences), while the prediction intervals may become useful when predicting the trajectory of a single driver, in for example vehicle-to-vehicle or vehicle-to-roadside architectures. Future studies will need to investigate these benefits of the Bayesian calibration framework.

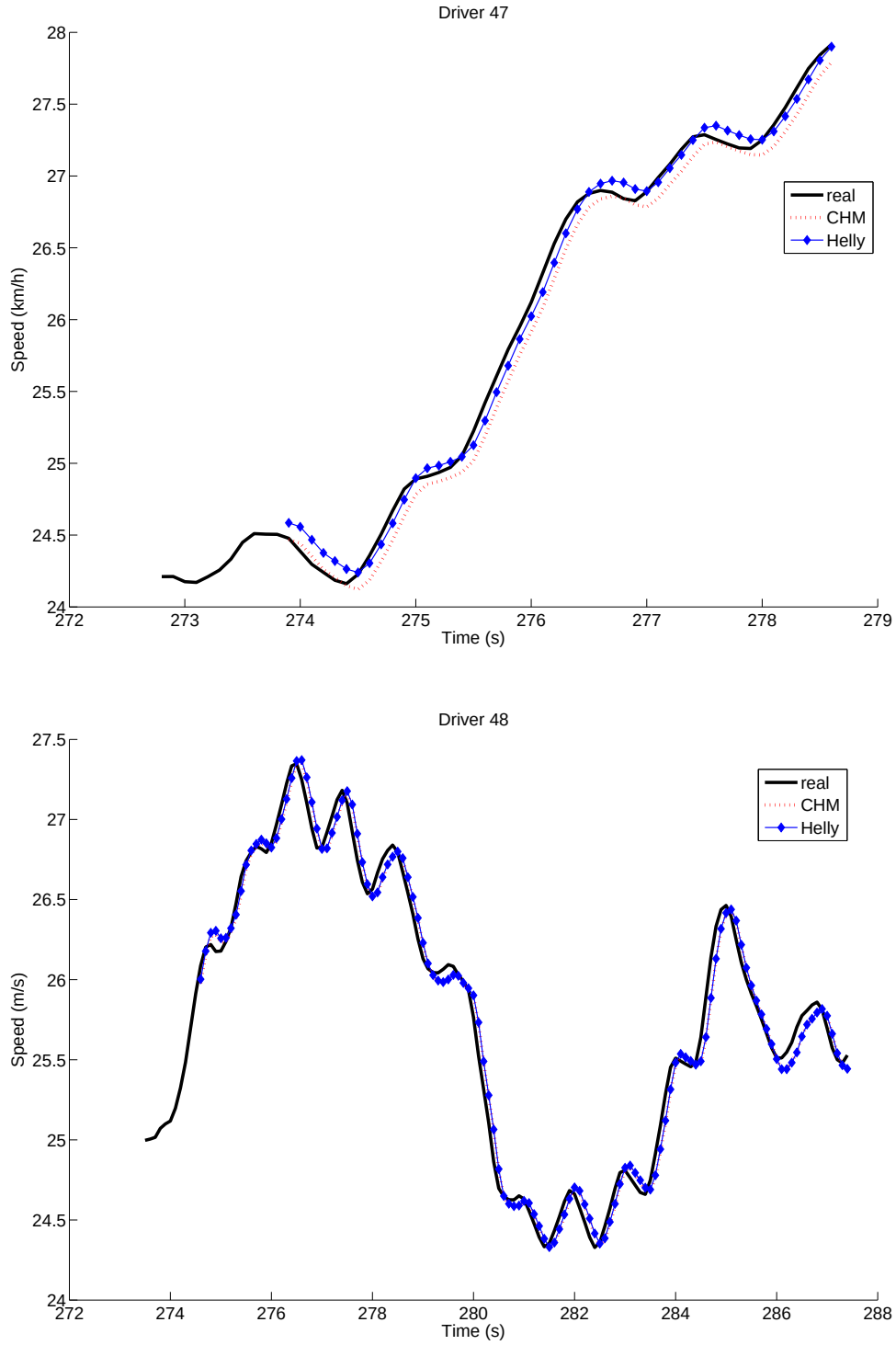


Figure 2.4: The actual versus predicted speed for driver 47 and 48

Chapter 3

Bayesian committee of regression models to predict travel times

This chapter is an edited version of van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008a). Bayesian combination of travel time prediction models. *Transportation Research Record: Journal of the Transportation Research Board*, 2064:73–80.

Short-term prediction of travel time is a central topic in contemporary intelligent transportation system (ITS) research and practice. Given the vast number of options, selecting the most reliable and accurate prediction model for one particular scientific or commercial application is far from a trivial task. One possible way to address this problem is to develop a generic framework that can automatically combine multiple models running in parallel. Existing combination frameworks use the error in the previous time steps. However, this method is not feasible in online applications because travel times are available only after they are realized; it implies that errors on previous predictions are unknown. A Bayesian combination framework is proposed instead. The method assesses whether a model is likely to produce good results from the current inputs given the data with which it was calibrated. A powerful feature of this method is that it automatically balances a good model fit with model complexity. With the use of two simple linear regression models as a showcase, this Bayesian combination is shown to improve prediction accuracy for real-time applications, but the method is sensitive in the event that all models are biased in a similar way. It is therefore recommended to increase the number and the diversity of the prediction models to be combined.

3.1 Introduction

Advanced Traffic Information Systems (ATIS) are widely acknowledged to have the potential to increase the reliability of road networks and to alleviate congestion and its negative environmental and societal side effects. However, for these beneficial collective effects to occur, reliable and accurate traffic information is required (van Lint et al., 2005). One valuable and objective piece of traffic information is travel time. Real-time travel time predictions can be used in dynamic traffic management (DTM) applications and in commercial applications for pre-trip planning or en-route navigation. Reliably and accurately predicting travel time for ATIS is a complex task that has been the subject of many research efforts over the past few decades.

In the international literature, many studies have focused on short-term travel time prediction. In van Hinsbergen et al. (2007) an overview of prediction methods is given. Many types of prediction models can be distinguished. However, every prediction model q has some set of assumptions (H_q) that can be physical, mathematical or statistical and a parameter vector $\boldsymbol{\theta}$ that must be determined from a certain data set D; a travel time prediction of model q can be written as $y_q = G(\boldsymbol{x}, \boldsymbol{\theta})|H_q$, where $\boldsymbol{x}$ represents the current input(s).

Given the myriad of prediction models and the complexity of travel time prediction, it is a far-from-trivial task to select the prediction model that is most reliable, most accurate, or both for a particular application. One possible way to approach this problem is to develop a generic framework that can automatically combine multiple models that run in parallel.

Prior attempts to combine multiple prediction models all use the error the models made in the previous time interval(s) (Petridis et al., 2001; Kuchipudi and Chien, 2003; Zheng et al., 2006). However, predicting travel time in real time, a necessity for most DTM and commercial applications, has one major complication: it takes time (the travel time, in fact) for the actual travel times to realize. Therefore, the actual travel time of the previous time step often is not yet revealed, especially in congested situations in which travel time prediction is most valuable. Using the error in the previous interval(s) when combining travel time prediction models hence must be considered a theoretical exercise and inapplicable to most real-time applications. Another approach is needed.

The goal of this study is to develop an alternative approach to the online combination of travel time predictions. A model's prediction and the probability that a model predicts the travel time correctly are used for this approach (Figure 3.2.1). For real-time applications, the probability must be calculated without looking at the errors in previous time intervals; only 'internal' information about the models can be used, with errors that have been revealed. In the model layer, multiple models simultaneously predict travel times, and the probability that a model is right about its prediction is computed for each model. The revealed prior prediction errors are stored in the data layer, which also provides in-

formation about current traffic conditions. In the combination layer, Bayes' theorem is used to combine the predictions, using the output of the model layer and of the data layer.

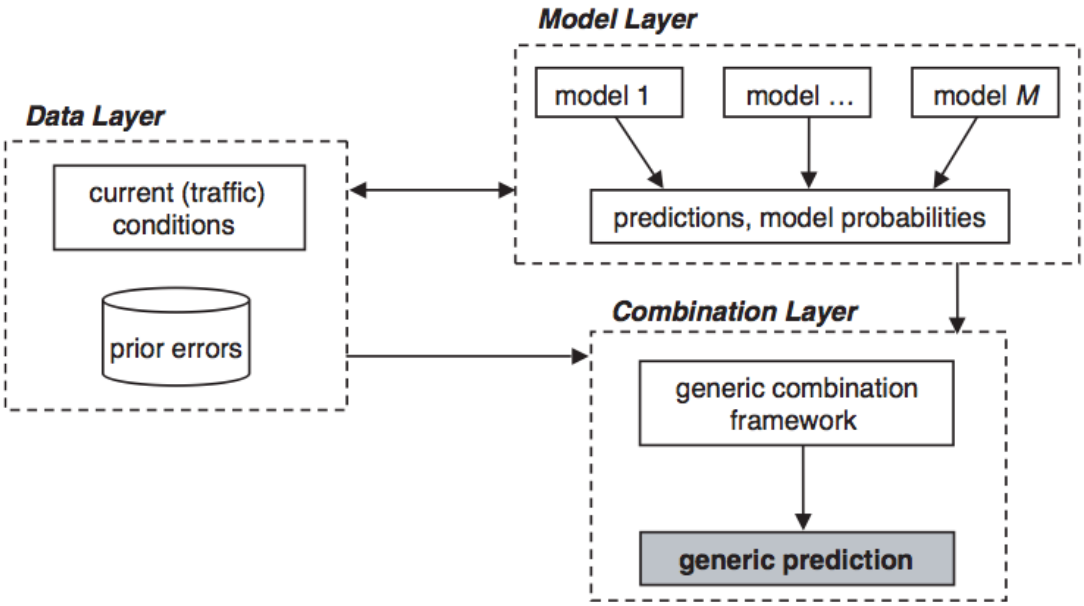


Figure 3.1: Framework for combining prediction models

3.2 Methodology

In this section it is shown how to combine multiple prediction models using the framework presented in Figure 3.2.1. These efforts are based on the Bayesian framework for model fitting and model comparison presented by Mackay (1995), providing a principled formalism that allows the ranking of the appropriateness (likelihood) of the model predictions, given the current inputs $\boldsymbol{x}$, the parameters $\boldsymbol{\theta}$, the data D with which these were calibrated, and all the other assumptions H underlying the model. The framework is the same framework that has been described in Chapter 2. Its derivation will be repeated here so that this chapter is individually readable.

3.2.1 Bayesian framework for model fitting and comparison

Assume M models in which a model $q \in M$ has underlying assumptions H_q (e.g., linear and contains 2 parameters, or nonlinear and contains 10 parameters) and is calibrated on a particular data set D. Using Bayes' theorem, the posterior probability that a model is

correct can be written as (Mackay, 1995, 2003):

$$P(H_q|D) = \frac{P(D|H_q)P(H_q)}{P(D)} \tag{3.1}$$

$$P(D) = \sum_{q \in M} P(D|H_q)P(H_q) \tag{3.2}$$

The numerator of (3.1) can be interpreted as the evidence for model H_q (more formally, $P(D|H_q)$ denotes the probability of generating data D using model H_q) times the prior probability of model q, $P(H_q)$, which could be based on belief or expert knowledge or statistics. Because the normalization constant in the denominator of (3.1) will be equal for all hypotheses tested, one can compare the posterior probabilities of different hypotheses on the basis of the numerator only. If it is further assumed that a priori each model is equally probable ($P(H_q)$ is equal for all q), then the different models can be evaluated and ranked on the basis of the evidence $P(D|H_q)$ only. The question is, how does one calculate this evidence?

Recall that most (travel time) prediction models are parameterized by means of some parameter vector $\boldsymbol{\theta}$, which is calibrated on some data set D, which in turn is assumed to be representative of the problem at hand. This process of model fitting usually entails minimizing some cost or objective function E:

$$\boldsymbol{\theta}_q^{MP} = \arg\min_{\boldsymbol{\theta}_q} \left(E(\boldsymbol{\theta}_q, D, H_q)\right) \tag{3.3}$$

which reflects the sum of squared errors on the calibration data D, for example, or some other goodness-of-fit measure. Mackay convincingly argues that model fitting should be viewed as probabilistic inference, in which the task is to find the maximum probable parameter vector $\boldsymbol{\theta}_q^{MP}$, given the available data D and all our other assumptions H_q (Mackay, 1995, 2003). That is,

$$\boldsymbol{\theta}_q^{MP} = \arg\max_{\boldsymbol{\theta}_q} \left(P(\boldsymbol{\theta}_q|D, H_q)\right) \tag{3.4}$$

In this case, Bayes' theorem yields

$$P(\boldsymbol{\theta}_q|D, H_q) = \frac{P(D|\boldsymbol{\theta}_q, H_q)P(\boldsymbol{\theta}_q|H_q)}{P(D|H_q)} \tag{3.5}$$

in which the denominator

$$P(D|H_q) = \int P(D|\boldsymbol{\theta}_q, H_q)P(\boldsymbol{\theta}_q|H_q)d\boldsymbol{\theta}_q \tag{3.6}$$

now equals the evidence from (3.1). Because this term would require integration

(marginalization) over the entire parameter space, calculating it analytically is possible only in simple models and even then requires elaborate calculations. In practice, it must be approximated.

To this end, first note that the model evidence (the denominator of (3.5)) does not depend on the parameters; they are integrated out of the equation. As a result, the parameter-fitting problem reduces to

$$\boldsymbol{\theta}_q^{MP} = \arg\max_{\boldsymbol{\theta}_q} \left(P(D|\boldsymbol{\theta}_q, H_q) P(\boldsymbol{\theta}_q|H_q) \right) \tag{3.7}$$

The first term in (3.7), $P(D|\boldsymbol{\theta}_q, H_q)$, equals the likelihood of the data arising from a model H_q with parameters $\boldsymbol{\theta}_q$, whereas the second term, the prior $P(\boldsymbol{\theta}_q|H_q)$, can be viewed as a term that bounds the parameter space to certain regions reflecting the prior belief on (or the known or desirable statistics of) these parameters. For example, if parameters have physical meaning (e.g., capacities or critical speeds in traffic models), then the prior enables us to incorporate these restrictions. In the Bayesian framework, model fitting thus leads to a posterior distribution of the parameters $p(\boldsymbol{\theta}_q|D, H_q)$ with a maximum at $\boldsymbol{\theta}_q^{MP}$ and variance $\boldsymbol{\Theta}_q$, rather than one particular parameter vector $\boldsymbol{\theta}_q^{MP}$.

3.2.2 Approximating the model evidence

Bayesians rank models on the basis of the evidence $P(D|H_q)$ for a certain model q after observing data. Mackay (1995, 2003), Bishop (1995), Sivia (1996) and Minka (2001) put forward clever approximations to quantify this evidence on the basis of the quantities calculated in the model fitting phase. Figure 3.2 shows the concept for a simple model with a one-dimensional parameter space. Let $p(\boldsymbol{\theta}_q|H_q)$ be the prior accessible volume by the model before fitting to the data (solid line). After the model is fitted to the data, the accessible volume collapses to $p(\boldsymbol{\theta}_q|D, H_q)$ (dashed line). The evidence of (3.6) equals the integral under this posterior $p(\boldsymbol{\theta}_q|D, H_q)$. Because this posterior usually is sharply peaked at $\boldsymbol{\theta}_q^{MP}$, this integral can be approximated by the height of its peak (at $\boldsymbol{\theta}_q^{MP}$) times its width, $\sigma_{\boldsymbol{\theta}_q|D}$, marked by the gray area on the right side:

$$P(D|H_q) \approx P(D|\boldsymbol{\theta}_q^{MP}, H_q) \times P(\boldsymbol{\theta}_q^{MP}|H_q)\sigma_{\boldsymbol{\theta}_q|D} \tag{3.8}$$

If the prior is (approximately) uniformly distributed (meaning each value of $\boldsymbol{\theta}$ within a particular range is equally probable), then (3.8) reduces to

$$P(\boldsymbol{\theta}_q^{MP}|H_q) \approx \frac{1}{\sigma_{\boldsymbol{\theta}_q|D}} \rightarrow P(D|H_q) \approx P(D|\boldsymbol{\theta}_q^{MP}, H_q) \times \frac{\sigma_{\boldsymbol{\theta}_q|D}}{\sigma_{\boldsymbol{\theta}_q}} \tag{3.9}$$

The first term of the right side of (3.9) can be interpreted as the best-fit likelihood of the model, which equals the height of peak of the dashed line in Figure 3.2. A model with

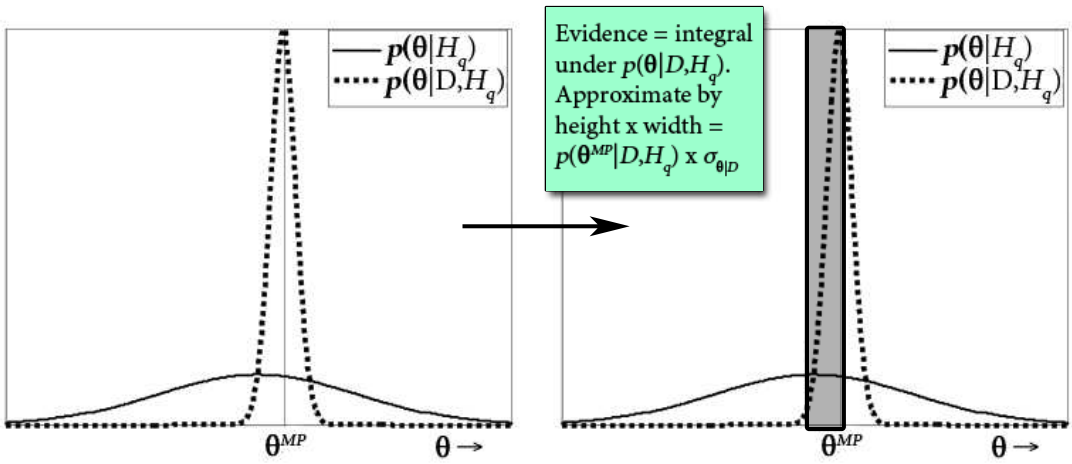


Figure 3.2: The evidence for the one-dimensional case. Adapted from (Mackay, 1995)

a higher peak is 'secure' about its fit. A higher best-fit likelihood therefore promotes models with a good fit. The second term of Equation (3.9) penalizes models that either have a large prior or a small posterior; it is called the Occam factor. Occam's razor is the concept of preferring the simpler model over the complex model if they predict the data equally well. Overly complex models with many parameters that are allowed to vary over a large parameter space tend to overfit the data and generalize poorly. The second term automatically penalizes a model that is overly complex, overfits the data, or both. Equation (3.9) therefore can be written as

$$\text{Evidence} = \text{Best-fit Likelihood} \times \text{Occam factor} \tag{3.10}$$

The evidence automatically balances a good fit on the data, overfitting, and model complexity. Recall that this same result was obtained in equation (2.17).

3.2.3 Normalizing the likelihood

In case the models that are to be compared do not all use the entire data set D, a problem possibly occurs: how can two models be compared if the data sets D in $P(H_q|D)$ are not equal? Of course, when the data sets are completely different, such a comparison will not be possible. However, when one model uses only a subset of the data set D while another uses the entire data set, then a heuristic can be applied and the comparison can still be made. If the models do not have equal numbers of data points which are used for calibration, then this discrepancy can be corrected by taking the log likelihood and

dividing over the number of data points. The corrected log likelihood for model q equals

$$\tilde{L}_q = \frac{\ln L_q}{N_q} \tag{3.11}$$

The corrected log likelihood then can be converted back to a likelihood in the range of $[0, 1]$. The evidence of (3.10) alters to the normalized likelihood times the Occam factor O_q:

$$Ev_q \propto \exp(\tilde{L}_q) \times O_q \tag{3.12}$$

Note that this heuristic will only work if all data comes from the same data set D but when one of the models uses not all data from D while another does.

3.2.4 The Occam factor for the multidimensional case

If a model has more than one parameter and the posterior distribution of the parameters can locally be approximated by a (multi-dimensional) Gaussian distribution, then the generalized variance (the determinant of the covariance matrix $\mathbf{\Sigma}_q$) can be used to describe the 'width' of the distribution (Mackay, 1995). Note that $\mathbf{\Sigma}_q$ is the covariance of the likelihood function, and not the covariance $\mathbf{\Theta}_q$ of the posterior distribution.The Occam factor O_q of a model q then equals

$$O_q = P(\boldsymbol{\theta}_q^{MP}|H_q)\det{}^{1/2}(\mathbf{\Sigma}_q) \tag{3.13}$$

If the optimal parameters are found by using a cost function that is efficient (i.e., that minimizes error variance such as the maximum likelihood estimate (MLE)), then the covariance matrix can be estimated as the inverse of the negative Hessian $\boldsymbol{A}_q$ of the log likelihood function $\ln L_q$ (Greene, 2000):

$$\mathbf{\Sigma}_q = (-\boldsymbol{A}_q)^{-1} = \left(-\frac{\partial^2 \ln L_q}{\partial \boldsymbol{\theta}_q^2} \right)^{-1} \tag{3.14}$$

3.2.5 Combination strategies

The models can be combined using their evidence. Two combination strategies are proposed:

- Winner Takes It All (WTIA)
  Mackay proposes to evaluate the evidence to rank the models: the model with the highest Evidence is chosen as the predictor (Mackay, 1995):

$$y(t) = y_q(t) \text{ where } q = \arg\max_q \left(Ev_q(t)\right) \tag{3.15}$$

- Weighted Linear Combination (WLC)
  It was investigated whether the evidence can be used as a weight. All M models' predictions are used but are multiplied by factors that add up to 1. If two models have an equal probability, then the truth intuitively will lie between the two predictions. A weighted linear combination is proposed in which evidence is normalized and used as a weight for a model's prediction:

$$y(t) = \sum_{q \in M} w_q(t) y_q(t) \tag{3.16}$$

$$w_q(t) = \frac{Ev_q(t)}{\sum_{r \in M} Ev_r(t)} \tag{3.17}$$

3.3 Proof of concept: two simple models

To demonstrate how the above theory can be applied in practice, two linear regression models where the evidence can be analytically solved are chosen: a linear regression (LR) model and a locally weighted linear regression (LWLR) model. For the sake of simplicity, a prediction horizon of 0 min ahead is chosen; predictions are made for the travel time on a route for vehicles that leave in the current time window.

Rice and van Zwet (2004) demonstrate that there is an approximate linear relationship between instantaneous travel time (ITT) and predicted travel time at a time interval t. Both models use the ITT, here denoted by the symbol ζ, in which traffic conditions (and therefore the speeds reported by loop detectors) are assumed to be constant for the whole trip. The vehicle speed is considered linearly increasing or decreasing between two detectors:

$$\zeta(t) = \frac{d_1}{u_1(t)} + \sum_{x=2}^{X} \left(\frac{2(d_x - d_{x-1})}{u_x(t) + u_{x-1}(t)} \right) + \frac{L - d_X}{u_X(t)} \tag{3.18}$$

where

$$X = \text{number of loop detectors on the route,} \tag{3.19}$$

$$d_x = \text{distance from loop detector to the beginning of the route,} \tag{3.20}$$

$$u_x = \text{speed reported by loop detector } x\text{, and} \tag{3.21}$$

$$L = \text{length of the route} \tag{3.22}$$

The first and last terms of (3.18) calculate the travel time from the beginning of the route to the first detector and the travel time from the last detector to the end of the route. Using the ITT, a simple linear model of the travel time equals

$$y(t) = \alpha + \beta\zeta(t) + \varepsilon(t) \tag{3.23}$$

where

$$y(t) = \text{travel time prediction at time window } t, \tag{3.24}$$
$$\zeta(t) = \text{instantaneous travel time at time window } t, \tag{3.25}$$
$$\alpha, \beta = \text{parameters to be estimated from data, and} \tag{3.26}$$
$$\varepsilon(t) = \text{random error that is normally distributed, } \varepsilon = N(0, \sigma^2). \tag{3.27}$$

MLEs of the parameters α, β and σ^2 are desired. Both linear regression models use a two-dimensional parameter vector, with parameters α and β. If these parameters are chosen to be drawn from the same parameter space, then the prior probability $P(\boldsymbol{\theta}_q|H_q)$ is equal for both models. Therefore, the Occam factor O_q of (3.13) reduces to

$$O_q \propto \det{}^{-1/2}(-\boldsymbol{A}_q) \tag{3.28}$$

For this two-dimensional case, the Hessian $\boldsymbol{A}_q$ of the log likelihood function (so not the Hessian of the posterior) equals

$$\boldsymbol{A}_q = \begin{bmatrix} \frac{\partial^2 \ln L_q}{\partial \alpha^2} & \frac{\partial^2 \ln L_q}{\partial \alpha \partial \beta} \\ \frac{\partial^2 \ln L_q}{\partial \beta \partial \alpha} & \frac{\partial^2 \ln L_q}{\partial \beta^2} \end{bmatrix} \tag{3.29}$$

where L is the likelihood function, which is Gaussian because the parameters are assumed to be normally distributed.

One assumption for the Bayesian framework is that the models are unbiased (i.e., the error distribution has 0 mean). To achieve this, the biases of the two models of prior predictions are incorporated into the predictions. It can be seen as the connection between the data layer and the combination layer in Figure 3.2.1: knowledge of prior errors is included in the Bayesian framework by subtracting the mean of the (revealed) prior error distribution of a certain model from the model's prediction.

3.3.1 Model 1: Linear Regression

Assume that a prediction is to be made at time step τ. Rice and van Zwet (2004) perform a linear regression on a data set of ITTs and actual travel times that were measured before time τ. Of the data set, only those points in history that have the same time of day are used for regression. For example, if one wishes to predict the travel time at 8 a.m., the model is fitted to a data set consisting of all pairs of TT and ITT at 8 a.m. in the data set.

Best fit likelihood

For a linear regression, the least squares estimate equals the MLE and satisfies the objective function

$$\min \sum_{t=1}^{\tau} \left(y(t) - \alpha - \beta\zeta(t)\right)^2 \tag{3.30}$$

where τ is the current data interval. In total, between $t = 1$ and $t = \tau$ there are N data points that are used for fitting the linear function; in this case, only those points are used of the same time of day. As stated before, the joint PDF of $y(t)$ is the product of the marginal PDFs (Casella and Berger, 1990). Because the parameters are assumed to be Gaussian distributed, the likelihood function equals

$$L_{LR}(\tau) = \prod_{t=1}^{\tau} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y(t) - \alpha - \beta\zeta(t)\right)^2}{2\sigma^2}\right) \tag{3.31}$$

The most probable parameters are estimated by the following equations (Casella and Berger, 1990):

$$\hat{\beta}(\tau) = \frac{\sum_{t=1}^{\tau}(\zeta(t) - \bar{\zeta})(y(t) - \bar{y})}{\sum_{t=1}^{\tau}(\zeta(t) - \bar{\zeta})^2} \tag{3.32}$$

$$\hat{\alpha}(\tau) = \bar{y} - \hat{\beta}\bar{\zeta} \tag{3.33}$$

$$\hat{\sigma}^2(\tau) = \frac{1}{N}\sum_{t=1}^{\tau}\left(y(t) - \hat{\alpha} - \hat{\beta}\zeta(t)\right)^2 \tag{3.34}$$

where $\bar{\zeta}$ is the mean ζ and $\bar{y}$ is the mean y. The best-fit likelihood is found by substituting the estimated parameters in the likelihood function of (3.31).

Occam factor

First, the log-likelihood function must be obtained from (3.31):

$$\ln L_{LR}(\tau) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{\sum_{t=1}^{\tau}\left(y(t) - \alpha - \beta\zeta(t)\right)^2}{2\sigma^2} \tag{3.35}$$

The Hessian $\boldsymbol{A}(\tau)$ is the second derivative of this log likelihood to the parameters (see (3.29)). The negative value of the Hessian equals

$$-\boldsymbol{A}_{LR}(\tau) = \begin{bmatrix} -\frac{N}{\sigma^2} & -\frac{\sum\zeta}{\sigma^2} \\ -\frac{\sum\zeta}{\sigma^2} & -\frac{\sum\zeta^2}{\sigma^2} \end{bmatrix} \tag{3.36}$$

The variance σ^2 can be obtained from (3.32). Substituting equation (3.36) in (3.28) gives the Occam factor:

$$O_{LR}(\tau) \propto \left(\frac{N \sum_{t=1}^{\tau} \zeta(t)^2 - \left(\sum_{t=1}^{\tau} \zeta(t)\right)^2}{\sigma^4} \right)^{-1/2} \tag{3.37}$$

3.3.2 Model 2: Locally Weighted Linear Regression

Locally weighted linear regression (LWLR) is a 'memory based' method, where model fitting is postponed until the moment of prediction (Atkeson et al., 1997; Zhong et al., 2005; Nikovski et al., 2005). The input vector, consisting of all measurements before the current point, is weighted by the proximity to the current measurements. This way, points in history that are close to the current situation are weighted more heavily in the regression than points farther away (Figure 3.3). ITT is used as input for the LWLR, analogously to the LR model described earlier.

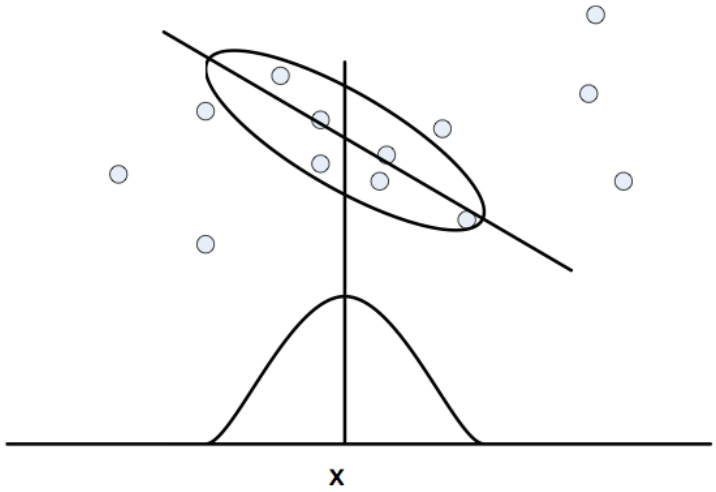


Figure 3.3: In locally weighted regression points are weighted by the proximity to the current point (x)

Best fit likelihood

The same linear model as in (3.23) is used, but the parameters are found by optimizing a weighted least squares (Atkeson et al., 1997):

$$\min \sum_{t=1}^{\tau} \mu(t) \left(y(t) - \alpha\zeta(t) - \beta\right)^2 \tag{3.38}$$

with τ the current time interval. In contrast to the LR model, not only points at the same time of day but all points in history for which the ITT and travel time are known are used for regression. The term $\mu(t,\tau)$ is a weight that is put on the input vector. Although no substantial empirical evidence prefers a certain weighting function over all others, a Gaussian kernel is often used (Atkeson et al., 1997; Zhong et al., 2005):

$$\mu(t,\tau) = \exp(-\delta(t,\tau)^2) \tag{3.39}$$

$$\delta(t,\tau) = \frac{|\zeta(t) - \zeta(\tau)|}{K} \tag{3.40}$$

where

$$\zeta(t) = \text{an instantaneous travel time at time } t \text{ some time in history,} \tag{3.41}$$

$$\zeta(\tau) = \text{current instantaneous travel time at time } \tau\text{, and} \tag{3.42}$$

$$K = \text{kernel width, which determines how quickly weights decline} \tag{3.43}$$

The prediction performance of the LWLR model was insensitive to the K value. The cost function of (3.38) was flat, with values of K between 50 and 150. Therefore, it was decided not to vary K in optimizing the objective function but to set it as a fixed value of 100s $(K = 100)$. Using the objective function of (3.38), the likelihood function becomes

$$L_{LWLR}(\tau) = \prod_{t=1}^{\tau} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\mu(t,\tau)(y(t) - \alpha - \beta\zeta(t))^2}{2\sigma^2}\right) \tag{3.44}$$

and the most likely parameter estimates become

$$\hat{\beta}(\tau) = \frac{\sum_{t=1}^{\tau} \mu(t,\tau)(\zeta(t) - \bar{\zeta})(y(t) - \bar{y})}{\sum_{t=1}^{\tau} \mu(t,\tau)(\zeta(t) - \bar{\zeta})^2} \tag{3.45}$$

$$\hat{\alpha}(\tau) = \bar{y} - \hat{\beta}\bar{\zeta} \tag{3.46}$$

$$\hat{\sigma}^2(\tau) = \frac{1}{N}\sum_{t=1}^{\tau} \mu(t,\tau)\left(y(t) - \hat{\alpha} - \hat{\beta}\zeta(t)\right)^2 \tag{3.47}$$

Occam factor

The Occam factor can be determined by substituting the log likelihood in (3.29) and substituting the result in (3.28), resulting in

$$O_{LWLR}(\tau) \propto \left(\frac{\sum_{t=1}^{\tau} \mu(t,\tau) \sum_{t=1}^{\tau} (\mu(t,\tau)\zeta(t)^2) - (\sum_{t=1}^{\tau} \mu(t,\tau)\zeta(t))^2}{\sigma^4}\right)^{-1/2} \tag{3.48}$$

3.4 Results

For this study, two data sets were available: one from license plate recognition cameras at the beginning and end of the route, and one from 19 double loop detectors along the whole route. The layout of the network is shown in Figure 3.4.

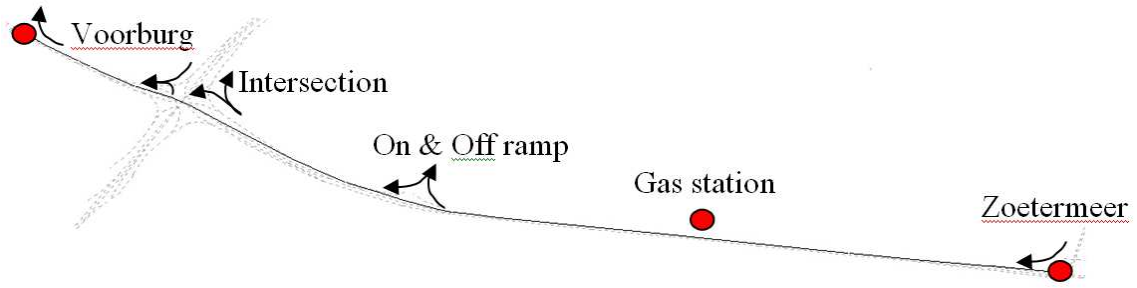


Figure 3.4: The A12 network from Zoetermeer to Voorburg, the Netherlands

The selected route is a 8.5-km (5.3-mi) stretch on the A12 motorway in the Netherlands, from an on ramp (Zoetermeer) to an off ramp (Voorburg). License plate cameras were placed at both ramps to record vehicle license plates, but only four of six characters were recorded because of privacy legislation. Individual travel times based on the four-character matches were recorded for 95 days in the winter and spring of 2007. The data were filtered for outliers, which are considerable, mainly because of coincidental matches between the four recorded license plate characters. After the data were filtered and visually inspected, the travel times of the vehicles leaving in the same 5-min period were aggregated. The selected motorway has a considerable morning peak but rarely an evening peak. Therefore, only the morning peaks between 7:00 and 10:00 a.m. were selected from the data sets.

For the same periods, double loop detector data were available that allow for the calculation of ITTs, used in both models. The loop detector data were available in 1-min arithmetic mean speeds for all vehicles that were recorded (i.e., time mean speeds). To smooth out large variances, five ITTs were calculated for each 5-min period using (3.18) and aggregated using the arithmetic mean.

The data sets were split in two: a training set of 80% (76 days) and a test set of 20% (19 days). To correct for bias, for the second half of the training set (38 days), predictions were made by the two models and the prediction errors were stored by model. Predictions were then made for each 5-min period in the test set, and the mean of the prior error distribution was subtracted from each prediction. For each period, the predictions also were combined using the two Bayesian combination strategies described earlier. This way, 681 predictions of 5 min were made. The models and the Bayesian framework were programmed in Microsoft Visual J#. The 681 predictions took a total time of 410 s per combination strategy on an Intel Pentium M 1.60 GHz, which included reading data files

of previously stored ITTs and actual travel times and storing the results. Of this time, only 1.0 s was spent on calculating the evidence.

Table 3.1 lists the prediction results and three performance indicators: the root-mean-square error (RMSE), which indicates the overall error; the bias, which shows a structural difference between the actual and predicted travel time and the root residual error (RRE), which is the remaining error after correcting for bias. Notice that

$$RMSE^2 = Bias^2 + RRE^2 \tag{3.49}$$

Table 3.1: Results of the different prediction models

Prediction model	RMSE (s)	Bias (s)	RRE (s)
LR	50.5	+1.6	50.5
LWLR	48.5	-2.5	48.4
Bayesian WTIA	48.0	-2.3	48.0
Bayesian WLC	47.9	-1.5	47.9

Table 3.1 shows that the Bayesian combined model has the best performance overall and that the WLC strategy is slightly better then the WTIA strategy. Using only internal information, calculated while calibrating the models, the combined predictions show a lower RRE and lower bias. However, on one day (Friday, April 13), the two models largely underestimated travel time. It is hypothesized that an accident occurred on that day, but no data are available to validate this theory. Nevertheless, this type of congestion apparently was not present in the training set, causing both models to perform badly. April 13 data therefore were deleted from the results, because the Bayesian combined model will never be able to create a good prediction from two bad predictions.

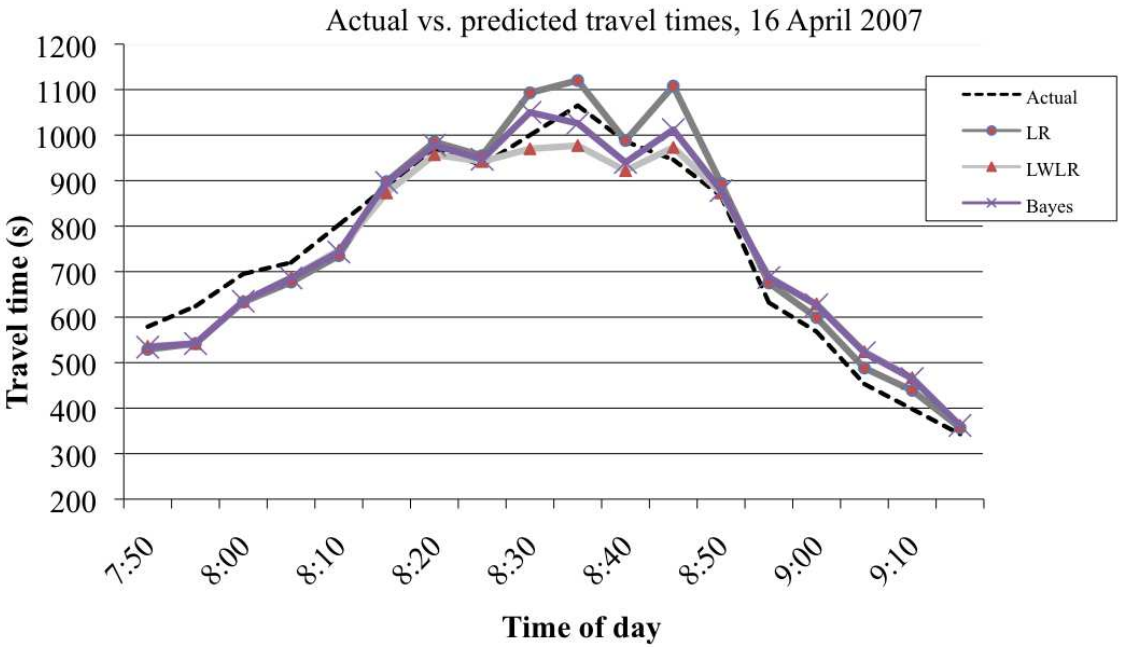
As an example of how the Bayesian framework combines the models, April 16 is considered (Figure 3.5). Especially at the peak of congestion, the Bayesian model can minimize errors. For most periods of this day, the evidence factor follows the model with the lowest prediction error. It especially can be seen at the peak of congestion, between 8:25 and 8:50 a.m. The single models show large peaks in the absolute error, but the combined model shows a more flat error, cutting off those peaks. Between 8:55 and 9:15 a.m., the Bayesian framework is sensitive to bias; the congestion dissolves rapidly. Both models lag behind and overestimate the travel time (i.e., both have a positive bias). The Bayesian combined model is bounded by the two prediction lines and therefore also shows a positive bias. Moreover, the evidence wrongly approves the LWLR model over the LR model. Although the lines of the two single models are close to each other, because of the steep descent of the travel times the vertical distance between the prediction and the actual travel time (the prediction error) is large. The effect of pointing out the wrong model is therefore large on these occasions.

3.5 Discussion and conclusion

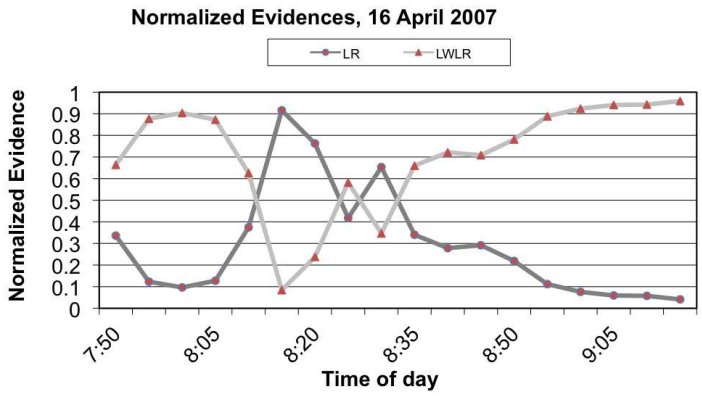
Using the Bayesian framework, the LR and LWLR prediction models were successfully combined, slightly improving prediction accuracy and reliability. Even though some simplifications were made, such as the errors and the posterior of the parameters being normally distributed, using the evidence as a ranking mechanism or as a weight improved results with limited extra effort. Additional research is needed to determine which combination strategy generally is preferable. The Bayesian framework for combining prediction models can be used for online DTM or commercial applications, because the often unrevealed error in previous interval(s) is not needed. Only 'internal' information on the models' probabilities is needed, and it can be calculated after the parameters of the models have been fitted to the data. This result is promising for both scientists and practitioners and encourages future research in this direction.

As the results show, the evidence was not always right about which model performed best, having consequences on performance. One way of overcoming this problem could be to introduce prior knowledge about the models' performance under certain conditions. It was assumed that every model is equally probably a priori ($P(H_q)$ is equal for all q); however, from prior predictions of the models, one may know which model performs better under which conditions. For example, the LR model may generally outperform the LWLR model in dissolving congestion. The Bayesian framework then balances model fit, overfitting, model complexity, and prior model performance.

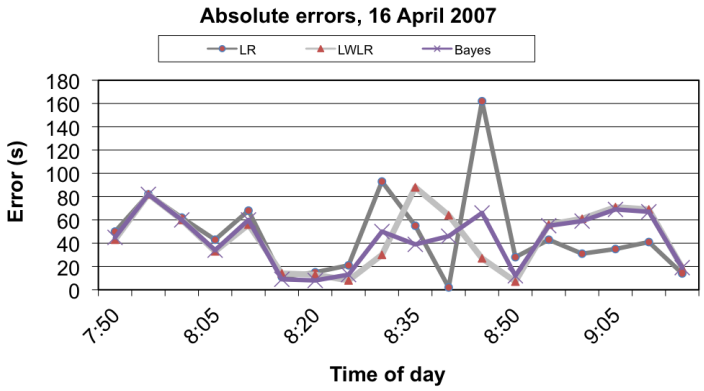
As the results indicate, the Bayesian combined model is sensitive to bias. It is bounded by the minimum and maximum predictions of all models. If all models have a bias with the same sign, then the Bayesian framework will have a larger prediction error than the best of the single models. Therefore, it is recommended to increase the number and diversity of the models inside the model layer of the framework. Doing so will decrease the chance of all models having the same bias. More advanced prediction models, such as neural networks or dynamic traffic assignment models, have shown promising prediction results in different circumstances (van Hinsbergen et al., 2007). Adding these models to the Bayesian framework can be expected to improve results.



(a)



(b)



(c)

Figure 3.5: Data from April 16, 2007: (a) actual versus predicted travel times for the two single models and the WLC combined model, (b) normalized evidence, and (c) absolute errors of the two models and the combined model

Chapter 4

Bayesian committee of neural networks to predict travel times

This chapter is an edited version of van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009d). Bayesian committee of neural networks to predict travel times with confidence intervals. *Transportation Research Part C: Emerging Technologies*, 17:498–509.

Short-term prediction of travel time is one of the central topics in current ITS research and practice. Among the more successful travel time prediction approaches are neural networks and combined prediction models (a 'committee'). However, both approaches have disadvantages. Usually many candidate neural networks are trained and the best performing one is selected. However, it is difficult to select the optimal network. In committee approaches a principled and mathematically sound framework to combine travel time predictions is lacking. This contribution overcomes the drawbacks of both approaches by combining neural networks in a committee using Bayesian inference theory. An 'evidence' factor can be calculated for each model, which can be used as a stopping criterion during training, and as a tool to select and combine different neural networks. Along with higher prediction accuracy, this approach allows for accurate estimation of prediction intervals. When comparing the committee predictions to single neural network predictions on the A12 motorway in the Netherlands it is concluded that the approach indeed leads to improved travel time prediction accuracy.

4.1 Introduction

The widely acknowledge potential of traffic information to alleviate congestion and to decrease negative environmental and societal side effects has led to a surge of research into reliable and accurate traffic and travel time prediction models in the past few decades (van Lint et al., 2005).

Among the most applied types of traffic prediction models are ARIMA-like time series approaches (Nihan, 1980; Lee and Fambro, 1999), Kalman filtering (Okutani and Stephanedes, 1984), local weighted regression (Sun et al., 2003; Zhong et al., 2005; van Hinsbergen et al., 2008a) (see also Chapter 3), nearest neighbor techniques (Smith and Demetsky, 1996; Clark, 2003), neural networks (Dougherty and Cobbett, 1997; Zhang, 2000; Dharia and Adeli, 2003; van Lint et al., 2005; Innamaa, 2005) and so called committee or ensemble approaches, in which multiple model-predictions are combined (Petridis et al., 2001; Kuchipudi and Chien, 2003; Zheng et al., 2006; van Hinsbergen et al., 2008a). The last two approaches, neural networks and committees, have shown a high accuracy for prediction of traffic conditions (van Hinsbergen et al., 2007). However, these two approaches exhibit some imperfections when applied in real-time applications, as will be shown in sections 4.1.1 and 4.1.2.

One valuable and objective piece of traffic information is the travel time. Real-time travel time predictions can be used in dynamic traffic management applications and in commercial applications, such as pre-trip planning or en-route navigation. In Chapter 3 two simple regression models were applied for the task of travel time prediction, and their predictions were combined using the Bayesian data assimilation framework. This contribution presents a neural network-based committee approach as an alternative for online travel time prediction, because, based on literature, their predictions are expected to be more accurate. The same Bayesian framework that has been applied in Chapters 2 and 3 will also be applied to these neural networks.

4.1.1 Committees of prediction models

One way of improving prediction accuracy and reliability is to combine multiple prediction models in a committee, where the outcomes are a weighted combination of the outcomes of its members. Previous attempts to combine traffic prediction models typically use the errors the models make in the previous time intervals (Petridis et al., 2001; Kuchipudi and Chien, 2003; Zheng et al., 2006). However, when applied to predicting travel time, one major complication occurs: it takes time (in fact the travel time) for the actual trip to be realized and consequently for a travel time to become available. Therefore, in most practical situations the actual travel time is not available within one discrete time step, especially in congested situations where accurate travel time prediction is most valuable. Using the error in the previous intervals to combine travel time prediction models must thus be considered a theoretical exercise and inapplicable to most real-time

applications (van Lint, 2008).

In Chapter 3 an alternative committee approach using Bayesian inference theory was applied to the travel time prediction problem. In this theory, a model's prediction as well as the probability that a model predicts the travel time correctly (the *evidence* for a model) is used. The relative probabilities of the models are then used to combine their predictions. This approach does not involve evaluating the prediction error of the last prediction(s) made, which makes it appropriate for online applications. In the chapter it is demonstrated that prediction accuracy can be improved using this approach.

4.1.2 Artificial neural networks

It is common practice in the application of (artificial) neural networks for travel time prediction to train many different candidate networks and then to select the best, based on the performance on an independent validation set, to make predictions. Although this might intuitively make sense, there are a number of serious drawbacks to this approach. In the first place, this implies that much effort involved in training networks is wasted. More seriously, the fact that one neural network model outperforms all other models on one particular validation data set does not guarantee that this neural network model indeed contains the optimal weights and structure, nor that this model has the best generalization capabilities. This completely depends on the statistical properties of the training and validation set (e.g. the amount of noise in the data), the complexity of the problem at hand and most importantly on the degree to which the training and validation set are representative for the true underlying process which is modeled. The network performing best on the validation set may therefore not be the one with the best performance on new data (Bishop, 1995).

These drawbacks can be overcome by combining all (or a representative selection of) trained neural network models in a committee. The Bayesian framework that is applied in Chapter 3 can be used for this purpose. The theory of Bayesian inference to train and combine a committee of feed-forward neural networks has been described in Bishop (1995) and Mackay (1992b, 1995) and has been applied in various fields of study (Thodberg, 1993; Mackay, 1994; Penny, 1999; Baesens et al., 2002; Chua and Goh, 2003; Lisboa et al., 2003). To the authors' knowledge this approach has not yet been applied to travel time prediction or traffic prediction in general.

4.1.3 Objective of this study

In this study the Bayesian approach for neural network based travel time prediction will be used and its workings will be demonstrated on real data from the A12 motorway in the Netherlands. In this approach two intrinsic and informative quantities are calculated, which allow for real time model comparison and combination. First, during training, the

so-called model-evidence is calculated, which ranks the models on the basis of the fit on the training data taking into account the degree of over-fitting (inducing variance) or under-fitting. Second, in actual operation the approach also allows the estimation of error bars on each prediction, which indicate the degree in which the currently presented input pattern matches with the input patterns seen during training. The committee approach is compared to individual neural networks to show that the committee provides a more accurate prediction of travel times and has better generalization performance.

As traffic systems are highly dynamic, it is expected that in order to make highly accurate travel time predictions, neural networks that are able to incorporate these dynamics are needed, such as recurrent neural networks or state-space neural networks (van Lint et al., 2005). However, to maintain focus on the workings and powerful properties of the Bayesian framework, relatively simple feed-forward neural networks are used in this study.

4.2 Methodology

In this section first the general approach to Bayesian model fitting is presented. Subsequently, the construction of a committee of neural networks and the derivation of error bars on each committee member's predictions are discussed.

4.2.1 Feed forward neural networks for travel time prediction

Figure 4.1 shows a typical feed-forward neural network topology with an input layer, a hidden layer and an output layer. The input layer consists of d input elements, the hidden layer of M hidden nodes and the output layer of c outputs.

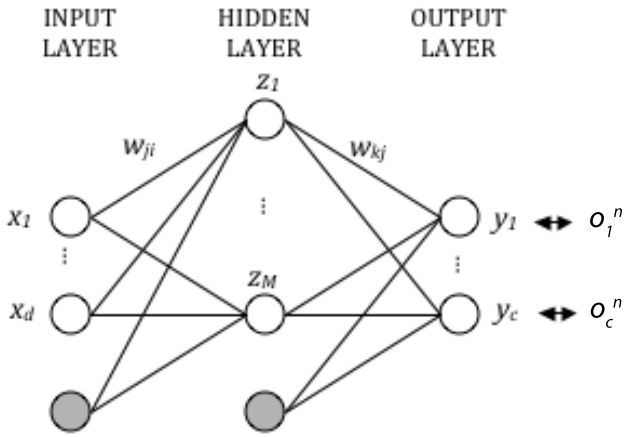


Figure 4.1: A neural network with d input elements, one hidden layer with M hidden nodes and c outputs, where the biases are represented as an extra node (in gray)

Mathematical description of a neural network

An output y_k, $k = (1, \ldots, c)$ can be described by the following equations:

$$y_k(\boldsymbol{x}) = f_2\left(\sum_{j=1}^{M+1} \theta_{kj} z_j\right) \tag{4.1}$$

$$z_j = f_1\left(\sum_{i=1}^{d+1} \theta_{jl} x_i\right) \tag{4.2}$$

where θ_{ji} and θ_{kj} are called *weights* which are adjustable and whose values need to be estimated from data. The *bias weights* (biases) are represented by an extra node in a layer to the left (the gray nodes in Figure 4.1) which have a constant output of 1, so $x_{d+1} = 1$ and $z_{M+1} = 1$. The functions f_1 and f_2 are called *activation functions* and apply transformations to the weighted sum of the output of the units to the left. Common forms of the activation of the hidden nodes are the *logistic sigmoid* and the *hyperbolic tangent* functions. In practice, the latter is found to give rise to faster convergence (Bishop, 1995). A linear activation function is commonly used for the output units.

$$f_1(a) = \tanh(a) \tag{4.3}$$

$$f_2(a) = a \tag{4.4}$$

The weights and biases together form a weight vector $\boldsymbol{\theta}$ with a total of W weights (parameters). The input vector $\boldsymbol{x}^n = (x_1^n, \ldots, x_d^n)$ is drawn from a data set $X = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N)$ of N data points. The output values of the network $\boldsymbol{y}(\boldsymbol{x}^n) = (y_1(\boldsymbol{x}^n), \ldots, y_c(\boldsymbol{x}^n))$ can be compared to target values $\boldsymbol{o}^n = (o_1, \ldots, o_c)$, drawn from a target data set $D = (\boldsymbol{o}^1, \ldots, \boldsymbol{o}^N)$. Only networks with a single output, $c = 1$, are considered in this study, so the index k will be omitted from now on.

Neural network training (model fitting)

The values of the weight vector $\boldsymbol{\theta}$ of the network need to be learned from data, which is usually referred to as neural network training. Typically this learning mechanism is based on a maximum likelihood approach, equivalent to the minimization of an error function such as the sum of squared error:

$$E_D = \frac{1}{2}\sum_{n=1}^{N} \left(y(\boldsymbol{x}^n, \boldsymbol{\theta}) - o^n\right)^2 \tag{4.5}$$

Preferably, a regularizer term is added to 4.5 to avoid overfitting of the networks to the training data. A commonly used regularizer is the *partitioned weight decay* error term

which has empirically been found to improve network generalization (Krogh and Herts, 1995) and is invariant to transformations to the input or output data (Bishop, 1995). Let us briefly explain this regularizer. Define V groups of weights $\boldsymbol{\theta}_v$, e.g. one for each layer and one for the biases, and define the regularizer by:

$$E_W = \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{4.6}$$

$$E_{W,v} = \frac{1}{2} \sum_{\theta \in \boldsymbol{\theta}_v} \theta^2 \tag{4.7}$$

where the parameters α_v control the extent to which the regularizer influences the solution. The regularized performance (error) function then becomes

$$E(\boldsymbol{\theta}) = E_D + E_W \tag{4.8}$$

The minimum of this performance function can be found by regular *back-propagation* or one of its many variations such as gradient descent (Rumelhart et al., 1986) or the (scaled) conjugate gradient algorithm (Williams, 1991; Johansson et al., 1991; Møller, 1993; Press et al., 2007). In the current study the scaled version of the latter algorithm is used.

In the conjugate gradient algorithm, a series of search directions $\boldsymbol{d}_j$ through weight space is constructed using the negative gradient $-\boldsymbol{g} = -\nabla E(\boldsymbol{\theta})$, which can be found by back propagating the errors (Hecht-Nielsen, 1989). A new search direction is set to always be *conjugate* to or *non-interfering* with all previous search directions, which ensures fast convergence to a minimum. After having found a search direction, the length of the step is determined using the Hessian $\boldsymbol{A} = \nabla\nabla E(\boldsymbol{\theta})$, which can be exactly evaluated by a back propagation approach (Bishop, 1992). In the scaled version of the conjugate gradient algorithm, a Levenberg-Marquardt technique is added to ensure that the quadratic error approximation that is used in the approach is valid for the step under consideration.

However, instead of using maximum likelihood techniques, neural network training can be viewed from a Bayesian inference perspective (Bishop, 1995; Mackay, 1995). This has some major advantages in the application of the neural networks. First, error bars can be assigned to the predictions of a network. Second, an automatic procedure for weighing the two error parts E_D and E_W of the error function can be derived; the values of these weights can be inferred simultaneously from the training data without the need of a separate validation data set. Because all data is used for training, better models will result. Third, the *evidence* measure emerging from the Bayesian analysis can be used as an early stopping criterion in the training procedure. Finally, different networks can be selected and combined in a committee approach using this evidence measure.

4.2.2 Bayesian trained neural networks for travel time prediction

From a Bayesian inference perspective, the parameters in a neural network (or any model for that matter) should not be conceived as single values, but as a *distribution* of values representing various degrees of belief. The goal is then to find the posterior probability distribution for the weights after observing the dataset D, denoted by $p(\boldsymbol{\theta}|D)$.

Neural network training formulated as Bayesian inference

This posterior can be found using Bayes' Theorem:

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \tag{4.9}$$

where $p(D)$ is the normalization factor, $p(D|\boldsymbol{\theta})$ represents a noise model on the target data and corresponds to the likelihood function, and $p(\boldsymbol{\theta})$ is the prior probability of the weights. Although many forms of the prior and the likelihood function are possible, often Gaussian forms are chosen to simplify further analyses:

$$p(\boldsymbol{\theta}) = \frac{1}{Z_w(\boldsymbol{\alpha})} \exp\left(-\sum_{v=1}^{V} \alpha_v E_{W,v}\right) \tag{4.10}$$

$$p(D|\boldsymbol{\theta}) = \frac{1}{Z_D(\beta)} \exp\left(-\frac{\beta}{2}\sum_{n=1}^{N} \left(y(\boldsymbol{x}^n, \boldsymbol{\theta}) - o^n\right)^2\right) \tag{4.11}$$

where Z_W and Z_D are normalizing functions and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_V)$ and β are called *hyperparameters* as they control the distributions of other parameters, the weights w of the network. The prior has zero mean and variances $1/\alpha_v$ for every group of weights, the likelihood function has zero mean and variance $1/\beta$. It can be seen that the exponents in 4.10 and 4.11 take the form of the error functions E_W and E_D already introduced in 4.8. Substituting 4.10 and 4.11 in 4.9 results in an expression for the posterior:

$$p(\boldsymbol{\theta}|D) = \frac{1}{Z_S(\boldsymbol{\alpha}, \beta)} \exp\left(-E(\boldsymbol{\theta})\right) \tag{4.12}$$

$$E(\boldsymbol{\theta}) = \beta E_D + \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{4.13}$$

where $Z_S(\boldsymbol{\alpha}, \beta)$ is a normalizing function. Consider now the maximum of the posterior distribution, $\boldsymbol{\theta}^{MP}$ (the most probable value of the weight vector). This can be found by minimizing the negative logarithm of 4.12, which is equivalent to minimizing 4.13. Because this equation is similar to 4.8 (except for an overall multiplicative factor), the maximum of the posterior $p(\boldsymbol{\theta}|D)$ can be found by simple and well-established back-

propagation techniques (see section 4.2.1).

Approximation of the posterior distribution of the weights

Although the most probable values for the weights (the peak of the posterior distribution) can be found using normal back-propagation, the entire posterior distribution needs to be evaluated to generate for example error bars on the predictions or to construct a committee of networks, as will be shown later. A complication here is that the normalizing coefficient $Z_S(\boldsymbol{\alpha}, \beta)$ of 4.12 in most cases cannot be evaluated analytically. Therefore, the posterior needs to be approximated, for example by a Taylor expansion (Mackay, 1992b), which results in the posterior

$$p(\boldsymbol{\theta}|D) = \frac{1}{Z_S} \exp\left(-E(\boldsymbol{\theta}^{MP}) - \frac{1}{2}\Delta\boldsymbol{\theta}^T \boldsymbol{A} \Delta\boldsymbol{\theta}\right) \tag{4.14}$$

where $\boldsymbol{A}$ is the Hessian given by

$$\boldsymbol{A} = \nabla\nabla E(\boldsymbol{\theta}) = \beta\nabla\nabla E_D + \sum_{v=1}^{V} \alpha_v \boldsymbol{I}_v \tag{4.15}$$

where E_D is the error function of equation 4.5 and $\boldsymbol{I}_v$ is a matrix with all elements zero except for the elements $\boldsymbol{I}_{ii} = 1$ where i corresponds to a weight from a group v. This estimation of the posterior distribution of the weights can be used to construct error bars and to create a committee of networks.

Approximation of the posterior distribution of the hyperparameters

In order to evaluate 4.13, the values (distributions) of the hyperparameters β and $\boldsymbol{\alpha}$ in 4.13 need to be found. These can be approximated by the same Bayesian inference framework that is used to approximate the posterior distributions of the weights. The posterior distribution of $\boldsymbol{\alpha}$ and β given the data D is given by:

$$p(\boldsymbol{\alpha}, \beta|D) = \frac{p(D|\boldsymbol{\alpha}, \beta)p(\boldsymbol{\alpha}, \beta)}{p(D)} \tag{4.16}$$

It can be shown (Gull, 1989; Mackay, 1992a; Bishop, 1995) that the maximum of this posterior can be approximated with the following values for $\boldsymbol{\alpha}$ and β:

$$\boldsymbol{\alpha}_v^{MP} = \frac{\gamma_v}{2E_{W,v}} \tag{4.17}$$

$$\beta^{MP} = \frac{N - \gamma}{2E_D} \tag{4.18}$$

where $\gamma = \sum_{v=1}^{V} \gamma_v$ is the so-called number of well-determined parameters, the elements of which are given by:

$$\gamma_v = \sum_{j=1}^{W} \left(\frac{\eta_j}{\eta_j + \alpha_j} \left(\boldsymbol{V}^T \boldsymbol{I}_v \boldsymbol{V} \right)_{jj} \right) \tag{4.19}$$

where η_j is the jth eigenvalue of the Hessian $\boldsymbol{A}$, $\boldsymbol{V}$ is the matrix of eigenvectors of the Hessian $\boldsymbol{A}$ and $\boldsymbol{I}_v$ was defined when explaining equation 4.15. In this summation negative eigenvalues are omitted (Thodberg, 1993).

In practice, the optimal values for $\boldsymbol{\alpha}$ and β as well as the optimal weight vector $\boldsymbol{\theta}^{MP}$ need to be found simultaneously. A simple heuristic is to use a standard iterative training algorithm (i.e. the scaled conjugate gradient algorithm) to find $\boldsymbol{\theta}^{MP}$ while periodically re-estimating the values of $\boldsymbol{\alpha}$ and β using 4.17 and 4.18.

The initial values of the hyperparameters depend on the typical values of the input (e.g. speeds, flows) and outputs (e.g. travel times). The data are transformed to ensure that all of the input and target variables are of order unity, in which case it is expected that the network weights also are of order unity, and thus the hyperparameters can be initialized to one. If the variables are treated as independent, they can be transformed by

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i} \tag{4.20}$$

where $\bar{x}_i$ is the mean of the ith variable and σ_i its standard deviation.

4.2.3 The evidence framework for committees of neural networks

In the next sections the Bayesian evidence framework for neural network training and model comparison will be discussed.

Calculating the evidence for a single neural network

Consider a certain neural network q with a set of assumptions H_q, such as the number of layers and the number of hidden units. The posterior probability of this model given the training data D, $P(H_q|D)$, can be determined using Bayes rule:

$$P(H_q|D) = \frac{p(D|H_q)P(H_q)}{p(D)} \tag{4.21}$$

where $P(H_q)$ is the prior probability of model H_q and $p(D|H_q)$ is called the *evidence* for model q. The evidence is a measure which intuitively and consistently combines a model's ability to fit the data with its complexity (Mackay, 1992a). It naturally embodies *Occam's Razor*, which states to prefer a simpler model over a more complex one given

it predicts the data sufficiently well and can be used to for example compare different models after they are trained. The evidence equals the denominator of 4.16 if the prior $P(H_q)$ is taken equal for all models and the conditional dependence on the model H_q are made explicit. Therefore, the evidence can be found using

$$p(D|H_q) = \int\int p(D|\boldsymbol{\alpha}, \beta, H_q) p(\boldsymbol{\alpha}, \beta|H_q) d\boldsymbol{\alpha} d\beta \tag{4.22}$$

If the same Gaussian approximation introduced in deriving 4.16 is assumed and the symmetries of neural networks with equal structures but different initial weights, corresponding to for example exchanges of weights or sign-flip' symmetries, are accounted for, the following logarithm of the evidence for a two-layer neural network model H_q emerges:

$$\begin{aligned} \ln p(D|H_q) = \sum_{v=1}^{V} \left(\frac{W_v}{2} \ln \alpha_v^{MP} + \frac{1}{2} \ln \frac{2}{\gamma_v} - \alpha_v^{MP} E_{W,v}^{MP} \right) - \beta^{MP} E_D^{MP} \\ -\frac{1}{2} \ln |\boldsymbol{A}| + \frac{N}{2} \ln \beta^{MP} + \ln M! + 2 \ln M + \frac{1}{2} \ln \frac{2}{N - \gamma} \end{aligned} \tag{4.23}$$

where terms which are equal for all models H_q are omitted, as only the relative values of the log evidence of the different models are of interest as will be shown later. For the exact derivation of this equation, the reader is referred to (Thodberg, 1993; Bishop, 1995).

As the determinant of the Hessian $\boldsymbol{A}$ in equation (18) is a product of the eigenvalues it is sensitive to errors in small values of the eigenvalues. Therefore, eigenvalues smaller than a certain cutoff value ϵ should be excluded when determining $|\boldsymbol{A}|$ to avoid numerical problems (Bishop, 1995).

Using the evidence as a stopping criterion

The evidence can be used as a stopping criterion or as a guide for pruning, due to its abilities to balance between model fit and model complexity (Thodberg, 1993). In this study, the development of the evidence is monitored during training. It is found by looking at many examples that the evidence flattens around the point when there is little to be gained in the generalization performance.

Figure 4.2 shows an example of this behavior for a case of predicting travel times where a dataset of 59 days was randomly split in two parts: 80% was assigned to a training set and the remaining 20% was used as a set to test the generalization performance. The log evidence, calculated using 4.23, of a network with 12 inputs and 15 hidden nodes hardly increases after epoch 100. Around the same time, the error of the network on the test set does not decrease anymore, although the training error does decrease if training is continued. The training can therefore be stopped once the increase in the evidence falls below a certain threshold value ς.

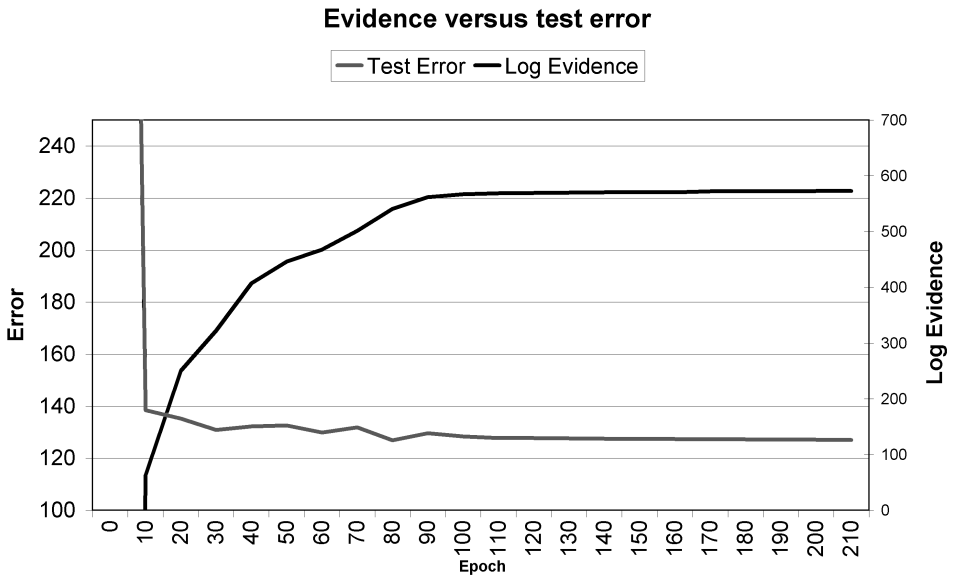


Figure 4.2: Evidence and test error during training of a network with 12 inputs and 15 hidden units

Constructing a committee on the basis of the evidence

The evidence that was derived in section 4.2.3 can also be used to select promising networks and to construct a committee. In a committee, the predictions of multiple models are combined. It has been shown that committees can lead to improved generalization (Thodberg, 1993; Bishop, 1995). In this study, neural networks with different structures and different weight distributions are combined.

Consider a generalized committee given by a weighted combination of predictions of its L members of the form (Perrone, 1994):

$$y_{GEN}(\boldsymbol{x}) = \sum_{q=1}^{L} \mu_q y_q(\boldsymbol{x}) \tag{4.24}$$

The best L committee members may be selected based on their evidence. Different types of weights μ_q are possible, but in this study a simple average over all committee members

is considered, $\mu_q = \frac{1}{L} \forall q$ (Thodberg, 1993; Mackay, 1994):

$$y_{GEN}(\boldsymbol{x}) = \frac{1}{L} \sum_{q=1}^{L} y_q(\boldsymbol{x}) \tag{4.25}$$

Note that in Chapter 3 the evidence was used as a weighting factor of the committee members. Initial experiments that were performed indicate that this did not result in different outcomes. Therefore, a simple average was used here.

4.2.4 Error bars on each committee member's predictions

If it is assumed that the output distribution arises from Gaussian noise on the output variables, that the distributions on the weights are Gaussian, and that the posterior distribution of the weights is sufficiently narrow so that it can be approximated by its linear expansion around $\boldsymbol{\theta}^{MP}$, then the output distribution of a single neural network is given by $N(y^{MP}, \sigma_t)$ where y^{MP} is the output of the network with the parameters set to $\boldsymbol{\theta}^{MP}$, and the standard deviation σ_t can be found by (Bishop, 1995):

$$\sigma_t^2 = \sigma_D^2 + \sigma_W^2 = \frac{1}{\beta} + \boldsymbol{k}^T \boldsymbol{A}^{-1} \boldsymbol{k} \tag{4.26}$$

where $\boldsymbol{A}$ is the Hessian and $\boldsymbol{k}$ is defined by:

$$\boldsymbol{k} \equiv \nabla_{\boldsymbol{\theta}} y|_{\boldsymbol{\theta}^{MP}} \tag{4.27}$$

This standard deviation 4.26 has two contributions: the first term reflects the spread (the uncertainty) in the target data, whereas the second term reflects the width of the posterior distribution of (and thus the uncertainty in) the network weights. The standard deviation can be used to construct error bars, for example 95% prediction intervals (twice the standard deviation).

A third and additional source of output variance is in the spread of the predictions between members of a committee. If the committee members' predictions are combined using the simple average of 4.25, it can be shown that the combined error bar for a prediction becomes (Thodberg, 1993):

$$\sigma_{total}^2 = \bar{\sigma}_D^2 + \bar{\sigma}_W^2 + \sigma_C^2 \tag{4.28}$$

where $\bar{\sigma}_D^2$ is the average over all σ_D^2, $\bar{\sigma}_W^2$ the average over all σ_W^2 and σ_C^2 is the committee

variance (the disagreement among the networks) given by:

$$\sigma_C^2 = \frac{1}{L}\sum_{q=1}^{L}\left(y_{GEN}(\boldsymbol{x}) - y_q(\boldsymbol{x})\right)^2 \tag{4.29}$$

In the next section all ingredients discussed so far are summarized and presented in a step-by-step description of the Bayesian committee approach.

4.2.5 Step-by-step procedure: committee of neural networks

To summarize all key concepts, below a step-by-step procedure is presented for making the committee predictions.

1. Construct many different neural networks with different numbers of hidden units and with different initial weight values.
2. For a model, draw initial weight values for the hyperparameters from their priors.
3. Train the networks by the scaled conjugate gradient algorithm.
4. Every step of the algorithm, re-estimate values for $\boldsymbol{\alpha}$ and β using 4.17 and 4.18.
5. Calculate the evidences for each network every few epochs. If the increase in the evidence relative to the previous epoch it was calculated falls below a certain threshold ς, stop, otherwise go to step 3 and repeat the procedures.
6. After all networks are trained, choose a selection of the better networks on the basis of their final evidences and construct a committee using 4.25.
7. Combine the error bars using 4.28 and draw 95% prediction intervals by adding and subtracting twice the standard deviation from the committee predictions.

4.3 Experiment

The theory of a committee of neural networks to predict travel times is applied to an 8.5 km (5.3 mi) long route of the A12 motorway in the Netherlands, from an on ramp (Zoetermeer) to an off ramp (Voorburg) (see Figure 4.3). This is the same network on which the regression models were tested in Chapter 3. On this route, 84 neural networks with the number of hidden nodes varying from 3 to 14 and with different initial weight values were trained, after which the networks with the highest evidence were selected and combined. To investigate the effects of early stopping discussed in section 4.2.3, the networks were also trained using a fixed number of 400 epochs, using the same structure and initial weight values as when trained with early stopping.

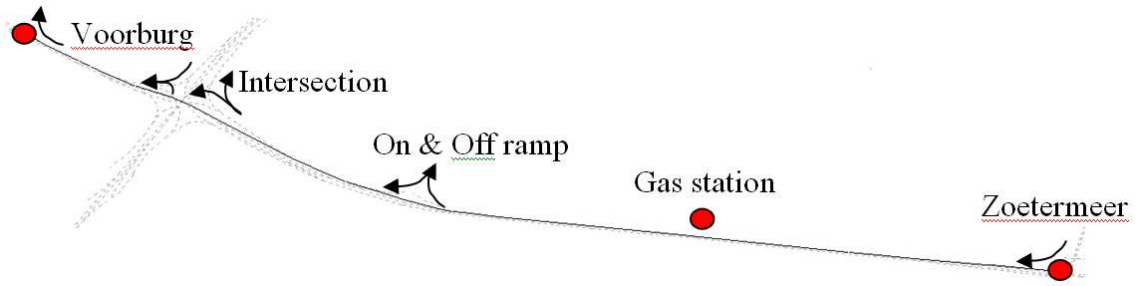


Figure 4.3: The A12 motorway from Zoetermeer to The Hague

4.3.1 Data

At both the on ramp and the off ramp license plate cameras are placed that record each vehicles' license plate. Individual travel times based on matches of license plates were available for 95 days in the winter and spring of 2007 (as in Chapter 3). The data were filtered for outliers, which were a considerable number, mainly due to the fact that only four characters out of six are recorded due to privacy legislations. After filtering the data and inspecting them visually, the travel times of the vehicles leaving in the same 5-minute time period were averaged. A total of 47 peak periods of about 3.5 hours each were selected from the data set. These peak periods were randomly split over two subsets: 37 peak periods with which the networks were initially trained and 10 peak periods on which the performance of the individual networks and of the committee was validated.

As input to the neural networks, 12 double loop detectors, evenly spread over the route, are available, reporting speeds and flows every minute. The speed data are available in one minute arithmetic mean speeds of all vehicles that are recorded (i.e. time mean speeds). Due to the inherent bias in time mean speeds when used as a proxy for space mean speeds, the speeds were corrected to space mean speeds using an estimate for the variance of the speeds in the one minute interval described in van Lint (2004); van Hinsbergen et al. (2008a).

4.3.2 Parameters

Initial values for the hyperparameters (section 4.2.2) were set to $\alpha_v = 1 \forall v$ and $\beta = 1$. The cutoff value when calculating the determinant of the Hessian (section 4.2.3) was set to $\epsilon = 10^{-10}$. The early stopping criterion ς (section 4.2.3) was set to 1%, where the evidence was evaluated every 10 epochs.

4.4 Results

Figure 4.4 shows the log evidence versus the test errors. A negative trend can be seen from this graph: the lower the error, the higher the log evidence; the fitted linear line has an R^2 of 0.52. As the R^2 deviates from zero, the graph shows that the evidence is informative about the accuracy of the predictions on a new data set, although the correlation does show imperfections. Figure 4.5 shows the effect of varying the size of the committee on the prediction error of the combined models. It shows that the optimal size is 4 for this case, and that after that point there is no gain in increasing the size of the committee.

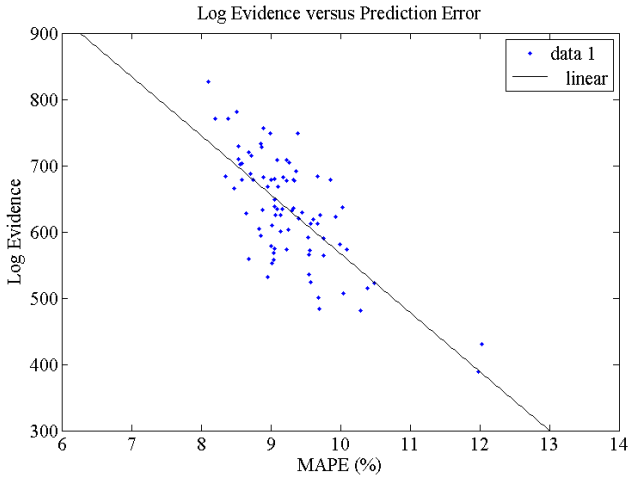


Figure 4.4: The log evidence versus MAPE on the test set for 84 different neural networks shows a negative trend

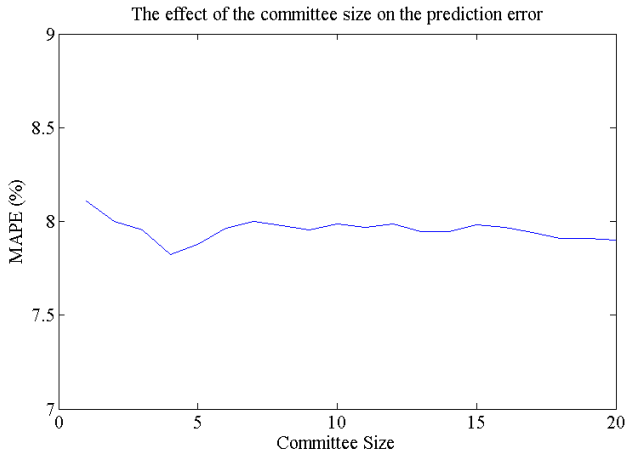


Figure 4.5: The MAPE versus the committee size shows an optimal size of 4

Table 4.1 shows the Mean Absolute Percentage Error (MAPE) of the committee of 4 networks compared to the 4 individual networks' predictions on the test set. It can be seen that the committee leads to a small gain in accuracy: a decrease of almost 0.3% in the error, compared to selecting the single network with the highest evidence, is achieved by retaining multiple networks and combining their predictions.

Table 4.2 presents the effects of the early stopping criterion discussed in section 4.2.3 on the training time for all 84 neural networks and the mean committee prediction error. The number of epochs when stopping early varied between 50 and 300, with a mean of 134 epochs. It can be seen that the total training time of the networks is much lower when using the early stopping criterion, at the cost of only a small decrease in performance.

Table 4.1: The performance of the individual networks compared to the committee prediction

Predictor	Log evidence	MAPE
#1	827.3	8.11%
#2	781.0	8.51%
#3	771.3	8.39%
#4	771.0	8.89%
Committee	-	7.82%

Table 4.2: The effect of early stopping on training and on the prediction results for 84 networks

Stopping criterion	Training time (min)	Mean epochs	Optimal committee size	Committee MAPE
$<$ 1% evidence increase	475	134	4	7.82%
400 epochs	1415	400	21	7.72%

Figure 4.6 shows a particular day where the error bars are plotted together with the committee predictions. The error bars are larger in the peak of the day, where the predictions are indeed deviating more from the actual travel times. It was found that 97.4% of the actual travel times fell within the calculated 95% committee prediction intervals. However, the prediction intervals are found to be too pessimistic on occasions, where the first factor of 4.26, $1/\beta$, appears to be dominant. This is due to the fact that the error term E_D is relatively large for all networks, as they show oscillating behavior around the actual travel times in some peak periods of the training days. This can be explained by the fact that relatively simple neural network architectures, which are not capable of capturing all traffic dynamics, are chosen in this study, as was already noted in the introduction (section 4.1.3).

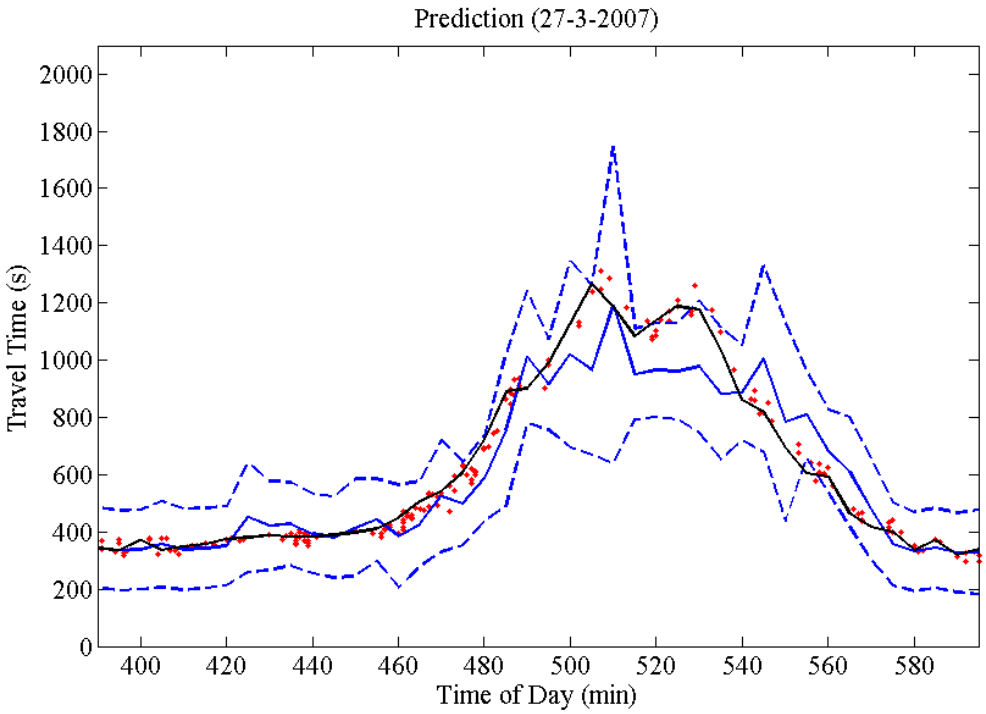


Figure 4.6: Prediction of travel time with prediction intervals. In congested situations, the error bars are larger than in free flow situations

4.5 Discussion

As is shown in Figure 4.4, the evidence is found to be informative on the generalization performance of the neural networks. It can therefore be used to select high performance neural networks from all models that are trained, without having to split the training set in two and using a part to test the generalization performance. The evidence framework provides a convenient and simple way to select high performance networks, leaving all training data to be used to train the networks. The correlation between the evidence and the test error does show imperfections, as is reported by other authors as well (Mackay, 1992b; Thodberg, 1993; Bishop, 1995). Apart from the fact that the calculation of the evidence involves several simplifications and assumptions, Mackay (1992b) notes that a poor correlation between evidence and generalization error may be an indicator for the limitations of the models. The neural networks used in this study all have one hidden layer and use only one time period of flows and speeds as input; in other words, the predictive power of these networks is limited due to their relatively simple input structures. It is expected that if some of these limitations are overcome, for example by using recurrent or state space networks (van Lint et al., 2005), the correlation between the evidence and the generalization error can become stronger. Furthermore, the test error is measured on a

finite data set and therefore is a noisy quantity, causing part of the scatter in Figure 4.4. It is expected that the correlation becomes stronger when the networks are tested on a larger data set.

The error of the committee is 0.3% lower than that of the individual neural network with the highest evidence. This means that the effort in training many candidate networks is not lost, but can be used to improve predictions. Besides this being positive for the modeler, the gain in prediction accuracy will benefit the road user, as they will have more accurate information available about the travel time they will experience. This may be beneficial to alleviate congestion and to decrease negative effects on the environment and the society.

The prediction interval provides a convenient way to inform the road user about the uncertainty of the predictions. It is desirable to avoid giving the road user a false sense of certainty when in fact the travel time proves hard to be predicted (by the selected prediction models). The users' trust of the information is an important factor for the impact of ATIS applications (Kantowitz et al., 1997), as providing inaccurate traffic information causes drivers to distrust the information and the possible beneficial effects of ATIS to decrease. The estimation of the error bars appeared to be too pessimistic on occasions, due to oscillating behavior of the models causing relatively large errors on training data. When more powerful models are used, the data error term E_D is expected to decrease, and as $\beta \sim \frac{1}{E_D}$, from equation 4.26 it follows that the prediction intervals will decrease as a result.

4.6 Conclusion

In this study two successful approaches to traffic prediction have been fused: combined prediction and neural networks. The Bayesian framework for neural networks, which is applied to traffic prediction for the first time, introduces a way of dealing with noisy input data when training neural networks and naturally leads to prediction intervals. A new stopping criterion using the evidence factor calculated for each neural network was introduced in the contribution. Furthermore, the evidence proved to be useful as a measure to select high performance networks and to form a committee of travel time predictors.

The predictions of the committee with the selected high-evidence networks proved to be more accurate than those of the individual networks. This leaves the modeler with a procedure to construct a more accurate prediction with very little additional effort, but more importantly, the end user with more accurate information. Together with the error bars that follow from the Bayesian analysis, the end user does not only receive more accurate traffic information, but also receives information on the reliability of the information and of the traffic conditions. This leads to more useful information for commercial as well as dynamic traffic management applications.

Future research will focus on the application of the theory on other traffic variables,

such as traffic flow, which can serve as an input to Dynamic Traffic Assignment (DTA) models. The DTA models can then be used to predict traffic conditions on entire road networks or as a dynamic traffic management tool.

If the Bayesian learning of the network weights is applied to networks with more powerful structures, such as recurrent networks, or with for example Bayesian pruning, and if the analysis is applied to larger training and test sets, it is expected that the evidence becomes more informative and the error bars become more accurate.

Chapter 5

Bayesian committee of state space neural networks to predict travel times

This chapter is an edited version of van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009e). Bayesian training and committees of state space neural networks for online travel time prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 2105:118–126.

This chapter presents the Bayesian framework that enables a unified way of constructing committees of an arbitrary number of models. The main contribution is that this framework is expanded for recurrent neural networks, which involves deriving the gradient and the Hessian of the network. State Space Neural Networks (SSNN), a special type of recurrent neural networks, are compared to Feed Forward Neural Networks (FFNN) and the effect of the Bayesian framework on both types is investigated on a freeway in the Netherlands. From a cross-validation procedure it can be concluded that for a short time horizon, both Bayesian training and recurrence do not lead to improvements, but that for a longer horizon both techniques are beneficial. It is shown that the use of a committee leads to improved performance and the correlation between the evidence factor, which follows from Bayesian model-fitting, with the generalization performance is compared versus the training error and the generalization performance. It is found that the evidence has lower correlation, which is an indication that (1) the dataset may be too small, (2) the used models require improvement and (3) the approximation of the evidence is imperfect. Future research will need to resolve these issues. However, the Bayesian framework will already be beneficial to more complex problems, and leads to estimations of error bars on the predictions, which may be useful for many applications.

5.1 Introduction

In Chapter 4 neural networks were applied for the task of online travel time prediction. A Bayesian framework was used to train and combine neural networks of different structures and sizes with a large data set. It was shown that this framework not only leads to better selection between models, but that the *evidence* measure that follows from the framework can also be used for early stopping, that accurate error bars can be constructed using the Bayesian analysis and that multiple models could be combined in a so-called *committee* leading to lower errors.

The neural networks used in Chapter 4 are *feed forward neural networks*. However, the traffic processes are highly dynamic. Therefore, it is expected that travel time predictions models which incorporate a dynamic component could further improve the prediction accuracy. Good candidate models are Elman Networks or State-Space Neural Networks (SSNN) (van Lint et al., 2005), which have been shown to produce good results in numerous studies (Yun et al., 1998; Dia, 2001; van Lint et al., 2002; Alecsandru, 2003; Ishak et al., 2003) and has been applied to freeways as well as urban streets (van Lint, 2004; Liu et al., 2005). Therefore, in this chapter the framework of Chapter 4 is applied to SSNN. To the authors best knowledge, the evidence theory for Bayesian model fitting and comparison, as described in (Mackay, 1992b; Thodberg, 1993; Mackay, 1995; Bishop, 1995) has so far only been applied to Feed Forward Neural Networks (FFNN).

The main contribution of this chapter is the application of the previously used Bayesian theory to recurrent neural networks, which has so far only been applied to feed-forward neural networks. The SSNNs are then compared to FFNNs and the effect of using the Bayesian theory is investigated.

5.2 Methodology

In this section first a brief general description of the SSNN is given. For a more elaborate description of the mathematics, see Chapter 4. Next, the Bayesian approach to fitting the parameters of the SSNN is described. It is then shown that this yields an automatic procedure for ranking and combining the SSNNs in a committee.

5.2.1 State Space Neural Networks for travel time prediction

Figure 5.1 shows a State Space Neural Network (SSNN) topology. It consists of an input layer, a hidden layer, a context layer and an output layer. The input layer consists of d input elements, the hidden layer and context layer of M hidden nodes and the output layer of c outputs. The inputs are grouped by road section; every hidden node represents one road section of the route under consideration and can be connected to a few or all inputs of the route under consideration. The context layer, which effectively represents a short

term memory of the internal states of the network, is fully connected to the hidden layer to allow the model to learn the different (upstream and downstream) dynamics of traffic.

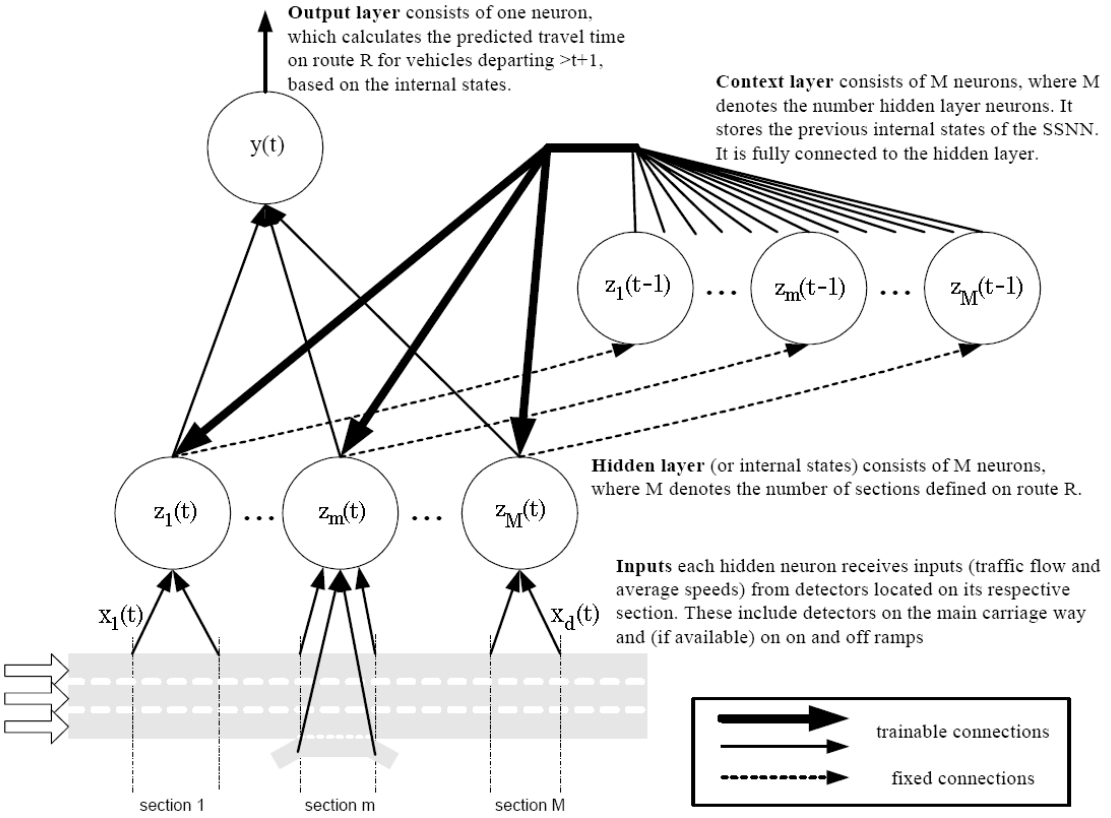


Figure 5.1: Topology of a state space neural networks, obtained from van Lint et al. (2005)

Mathematical description of the SSNN

An output $y_k, k = (1, \ldots, c)$ can be described by the following equations:

$$y_{k,t}(\boldsymbol{x}) = f_2(a_{k,t}) = f_2\left(\sum_{j=1}^{M+1} \theta_{kj} z_{j,t}\right) \tag{5.1}$$

$$z_{j,t} = f_1(a_{j,t}) = f_1\left(\sum_{i=1}^{d+1} \theta_{ji} x_i + \sum_{l=1}^{M} + \theta_{jl} z_{l,t-1}\right) \tag{5.2}$$

where θ_{ji} are the weights from the inputs to the hidden layer, θ_{jl} are the weights from the context layer to the hidden layer and θ_{kj} are the weights from the hidden layer to the output layer. Note that (5.1) is equal to (4.1) but that (5.2) now contains an additional term

compared to (4.2). Also note that now the time step of the data t is explicitly included as an index to all variables, because values of both t as well as $t-1$ are used in the computations. In the SSNN, the input layer may not be fully connected to the hidden layer but the context layer is fully connected to the hidden layer. Also note that in the first time step, $t = 1$, $z_{l,t-1} = z_{l,0}$ will not exist, and that a constant value C for $z_{l,0}$ will be chosen to initialize the context units with. The bias weights (biases) are represented by an extra node in the input layer and hidden layer which have a constant output of 1, so $x_{d+1} = 1$ and $z_{M+1} = 1$. The functions $f_1(a)$ and $f_2(a)$ are called activation functions and apply transformations to the weighted sum of the output of the connected units. A logistic sigmoid is used for the hidden layers and a linear activation function is used for the output units, just as in (4.3) and (4.4):

$$f_1(a) = \tanh(a) \tag{5.3}$$
$$f_2(a) = a \tag{5.4}$$

All weights (parameters) together form a weight vector $\boldsymbol{\theta}$ of size W. The same definitions for the input vector $\boldsymbol{x}^n$, output vector $\boldsymbol{y}(\boldsymbol{x}^n)$ and target values $\boldsymbol{o}^n$ as in Chapter 4 is kept. Note that in order for the context layer to remain consistent, the data set needs to consist of a continuously chronological set of values, in which case the time step index k is equal to the data index n.

In matrix notation, the SSNN can be written in a state space form, hence the name State Space Neural Network (van Lint et al., 2005):

$$\begin{aligned} \boldsymbol{y}_t &= f_2(\boldsymbol{\theta}_k \boldsymbol{z}_t) \\ \boldsymbol{z}_t &= f_1(\boldsymbol{\theta}_j \boldsymbol{x}_t + \boldsymbol{\theta}_l \boldsymbol{z}_{t-1}) \end{aligned} \tag{5.5}$$

The vectors $\boldsymbol{\theta}_j$, $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_k$ contain the weights from input to hidden, context to hidden and hidden to output layers respectively.

Neural network training (model fitting)

The same training algorithm as in 4.2.1 is used for the SSNN. A slightly different definition will be used for the data error E_D:

$$E_D = \frac{1}{2} \sum_{t=1}^{N} \sum_{k=1}^{C} \left(y_{k,t} - o_{k,t}\right)^2 \tag{5.6}$$

The same regularizer is used:

$$E_W = \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{5.7}$$

$$E_{W,v} = \frac{1}{2} \sum_{w \in \boldsymbol{\theta}_v} w^2 \tag{5.8}$$

where the hyperparameters α_v control the extent to which the regularizer influences the solution. Adding also a hyperparameter β for the data error, the regularized performance (error) function then becomes

$$E(\boldsymbol{\theta}) = \beta E_D + E_W \tag{5.9}$$

The minimum of this performance function is found using the scaled conjugate gradient algorithm (Williams, 1991; Johansson et al., 1991; Møller, 1993; Press et al., 2007).

5.2.2 Neural network training formulated as Bayesian inference

Just as in 4.2.2 the training can also be viewed from a Bayesian inference perspective. The parameters in the SSNN are no longer conceived as single values, but as a distribution of values representing various degrees of belief. The posterior distribution of the parameters is given by

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \tag{5.10}$$

Again, Gaussian forms are chosen for the prior and for the likelihood:

$$p(\boldsymbol{\theta}) = \frac{1}{Z_w(\boldsymbol{\alpha})} \exp\left(-\sum_{v=1}^{V} \alpha_v E_{W,v}\right) \tag{5.11}$$

$$p(D|\boldsymbol{\theta}) = \frac{1}{Z_D(\beta)} \exp\left(-\frac{\beta}{2} \sum_{t=1}^{N} \sum_{k=1}^{c} \left(y_{k,t} - o_{k,t}\right)^2\right) \tag{5.12}$$

Substituting 5.11 and 5.10 into 5.10 results in an expression for the posterior:

$$p(\boldsymbol{\theta}|D) = \frac{1}{Z_S(\boldsymbol{\alpha}, \beta)} \exp\left(-E(\boldsymbol{\theta})\right) \tag{5.13}$$

$$E(\boldsymbol{\theta}) = \beta E_D + \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{5.14}$$

The maximum of the posterior distribution $\boldsymbol{\theta}^{MP}$ is found by minimizing the negative logarithm of 5.13, which is equivalent to minimizing 5.14. As in Chapter 4 simple and well-established back-propagation techniques are used for this. Finally, the same approximation for the hyperparameters as in 4.17 and 4.18 is used and the input and output data is transformed using 4.20.

5.2.3 Determination of the gradient

For training and for the Bayesian framework, the gradient of the error function towards the weights needs to be known. In contrast to the feed-forward neural networks, where well-established algorithms exist for the exact calculation of the gradient (Rumelhart et al., 1986) and Hessian (Bishop, 1992) through back-propagation there exist no exact definitions for recurrent neural networks for these first and second derivatives yet. In this section, the gradient is determined for each weight in the State Space Neural Network; in the next section the same will be done for the Hessian.

To determine the derivative of the error function to the weights at a certain epoch, consider the data error E_D and the regularizer errors $E_{W,v}$ separately, so:

$$\nabla E(\boldsymbol{\theta}) = \beta \nabla E_D + \nabla \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{5.15}$$

The derivative of the second term is straightforward:

$$\nabla \sum_{v=1}^{V} \alpha_v E_{W,v} = \sum_{v=1}^{V} \alpha_v \boldsymbol{I}_v \boldsymbol{\theta}_v \tag{5.16}$$

where $\boldsymbol{I}_v$ is a matrix with all elements zero except for some diagonal elements $\boldsymbol{I}_{ii} = 1$ where i is the index in the weight vector $\boldsymbol{\theta}$ of a weight belonging to a group v.

The gradient of E_D is more complex. Because this term is a sum over all N input patterns, the gradient of the error over one pattern n (which is equivalent to the error at time step t as noted before) will be considered first, which is defined as $E_{D,t} = \frac{1}{2}\sum_k (y_{k,t} - o_{k,t})^2$, and later be summed over all patterns N to obtain the full gradient. Define the part of the error from one output k as $E_{D,k,t} = \frac{1}{2}(y_{k,t} - o_{k,t})^2$. For an arbitrary weight θ_q in any layer of the network, it holds that it only influences $E_{D,t}$ through the outputs y_k, so the chain rule for partial derivatives can be applied:

$$\begin{aligned} \frac{\partial E_{D,t}}{\partial \theta_q} &= \sum_k \frac{\partial E_{D,k,t}}{\partial y_k} \frac{\partial y_k}{\partial \theta_q} \\ &= \sum_k \delta_{k,t} \frac{\partial y_k}{\partial \theta_q} \end{aligned} \tag{5.17}$$

where $\delta_{k,t} = \partial E_{D,k,t}/\partial y_k = (y_{k,t} - o_{k,t})$. Substituting 5.1, 5.2, 5.3 and 5.4 in 5.17, the derivatives for the weights in each layer are found. The exact derivation is given in Appendix A; here, only the resulting equations are given:

$$\frac{\partial E_{D,t}}{\partial \theta_{kj}} = \delta_{k,t} f_2'(a_{k,t}) z_{j,t} \tag{5.18}$$

$$\frac{\partial E_{D,t}}{\partial \theta_{ji}} = \sum_k \delta_{k,t} f_2'(a_{k,t}) h_{kji,t} \tag{5.19}$$

$$\frac{\partial E_{D,t}}{\partial \theta_{jl}} = \sum_k \delta_{k,t} f_2'(a_{k,t}) g_{kjl,t} \tag{5.20}$$

where θ_{kj} is a weight in the output layer, θ_{ji} a weight in the hidden layer and θ_{jl} a weight in the context layer, $\delta_{k,t} = (y_{k,t} - o_{k,t})$ and where the auxiliary variables $h_{kji,t}$ and $g_{kjl,t}$ are defined by:

$$h_{kji,t} = \sum_{j'} \theta_{kj'} f_1'(a_{j',t}) \omega_{j'ji,t} \tag{5.21}$$

$$\omega_{j'ji,t} = \Delta_{j'j} x_{i,t} + \sum_l \theta_{j'l} f_1'(a_{l,t-1}) \omega_{lji,t-1} \tag{5.22}$$

$$g_{kjl,t} = \sum_{j'} \theta_{kj'} f_1'(a_{j',t}) \eta_{j'jl,t} \tag{5.23}$$

$$\eta_{j'jl,t} = \Delta_{j'j} z_{l,t-1} + \sum_{l'} \theta_{j'l'} f_1'(a_{l',t-1}) \eta_{l'jl,t-1} \tag{5.24}$$

where Δ is the Kronecker delta symbol. Finally, the starting conditions $\eta_{jjl,1} = \Delta_{jj} C \forall j, j, l$ and $\omega_{jji,1} = \Delta_{jj} x_{i,1} \forall j, j, i$ hold. What should be noted is that the recursive variables (of time $t-1$) in an actual application can be kept in memory for the next iteration, and can be overwritten at the end of each time step to be used later.

The total gradient of the error term E_D can now be obtained by concatenating all values into a vector of size W (the total number of weights in the network), summing over all t and multiplying the resulting vector by β. The gradient term of 5.16 is then added to obtain the entire gradient for a certain epoch.

5.2.4 Determination of the Hessian

To determine the step size in the conjugate gradient algorithm and to calculate the Bayesian evidence, the Hessian $\boldsymbol{A}$ is required for the State Space Neural Network. As with the gradient, no procedure for finding the exact $\boldsymbol{A}$ exists yet; here it will be derived.

Again, two error parts E_D and $E_{W,v}$ are considered separately:

$$\boldsymbol{A} = \nabla^2 E(\boldsymbol{\theta}) = \beta \nabla^2 E_D + \nabla^2 \sum_{v=1}^{V} E_{W,v} \tag{5.25}$$

The second term again is straightforward:

$$\nabla^2 \sum_{v=1}^{V} \alpha_v E_{W,v} = \sum_{v=1}^{V} \alpha_v \boldsymbol{I}_v \tag{5.26}$$

The first term, the error part E_D is first considered per pattern n (time step t), $E_{D,t}$, and later summed over all n to obtain the full value. If two arbitrary weights θ_q and θ_r of any two layers are considered, the previously derived first derivatives (see 5.2.3) can be used:

$$\frac{\partial^2 E_{D,t}}{\partial \theta_q \partial \theta_r} = \frac{\partial}{\partial \theta_q} \left(\frac{\partial E_{D,t}}{\partial \theta_r} \right) \tag{5.27}$$

The appropriate expression for $\partial E_{D,t} / \partial \theta_r$ can then be substituted and the second derivatives can be constructed from the result. As this procedure is very lengthy but only involves straightforward algebra, the complex-looking outcomes are omitted but the results again only contain recursive variables from one time step back, $t-1$, which in an application can be kept in memory and overwritten at the end of each time step. The final Hessian $\boldsymbol{A}$ is obtained by concatenating all the values into a matrix of size W by W, by summing over all t, multiplying the obtained matrix by β and by adding the part of equation 5.26. The exact derivation is given in Appendix A.

However, in a final application, the above procedure becomes very slow, especially due to the presence of three recurrent variables which require 6-fold loops for each input vector. Therefore, an approximation of the Hessian is useful to speed up calculations per iteration. If the sum-of-squares error function is considered, the elements of the Hessian can be written in the form (Bishop, 1995):

$$\frac{\partial^2 E_D}{\partial \theta_q \partial \theta_r} = \sum_{t=1}^{N} \sum_{k=1}^{c} \frac{\partial y_{k,t}}{\partial \theta_q} \frac{\partial y_{k,t}}{\partial \theta_r} + \sum_{t=1}^{N} \sum_{k=1}^{c} (y_{k,t} - o_{k,t}) \frac{\partial^2 y_{k,y}}{\partial \theta_q \partial \theta_r} \tag{5.28}$$

As the quantity $(y_{k,t} - o_{k,t})$ is a random variable with zero mean (if the biases in the network are well-trained), uncorrelated with the value of the second derivative term, this whole term will tend to average to zero in the summation over t (Hassibi and Stork, 1993). This term can therefore be neglected, resulting in the so-called *outer-product approximation*:

$$\frac{\partial^2 E_D}{\partial \theta_q \partial \theta_r} \approx \sum_{t=1}^{N} \sum_{k=1}^{c} \frac{\partial y_{k,t}}{\partial \theta_q} \frac{\partial y_{k,t}}{\partial \theta_r} \tag{5.29}$$

As this term only involves first derivatives of the outputs to the weights, which were already derived when solving equation 5.17 for the different layers in the network, the evaluation is much easier and faster than the exact procedure. Extensive tests on various SSNN topologies with exact and approximate Hessians show that the use of the approximate Hessian leads to similar prediction accuracy, and that the approximate procedure is much faster. The use of the outer product approximation is therefore preferred over the exact Hessian procedure in this study.

5.3 Experiment

The same 8.5 km (5.3 mi) long route of the A12 motorway in the Netherlands, from an on ramp (Zoetermeer) to an off ramp (Voorburg) was investigated for this study (see Figure 5.2)), the same network that was used in Chapter 3 and Chapter 4. Four different types of neural networks were trained to predict travel times on this route: Bayesian SSNN, Bayesian FFNN, non-Bayesian SSNN and non-Bayesian FFNN. For the non-Bayesian procedure, constant hyperparameter values were chosen (see equation 5.9) for the entire training procedure. Extensive experiments were carried out to find optimal values for the fixed hyperparameters for these networks, resulting in $\beta = 1.5$ and $\alpha_v = 1 \forall v$. In total 70 FFNN and 70 SSNN, with different structures (varying from 4 to 10 hidden nodes) and in the case of SSNN some networks having fully and others partially connected input layers, were trained on a small random training set. After testing all networks on the rest of the dataset, 5 FFNN and 5 SSNN were selected that showed low error. These promising networks were then used for further comparison.

The SSNN needs to learn weights for the context layer from the data. In that light, each whole day only represents one data point to the memory layer. Therefore, the training set needs to be as large as possible in order to obtain good results from the SSNN. The dataset was therefore randomly split in 33 days for training and 6 days for performance testing. To ensure that the results do not heavily depend on the random component in dividing the dataset, a cross-validation approach was used. The procedure of splitting the dataset, training and testing was repeated 5 times to investigate the generalization ability of the different types of neural networks.

Then, the networks were ranked to form a committee. In case of the Bayesian networks, the evidence was used as a ranking mechanism; for the non-Bayesian networks, the data error ED was used. The highest ranking networks were then combined using equation 4.25 to produce a committee prediction for different committee sizes. Two prediction horizons were used: a 5-minute-ahead (one step) and a 15-minute-ahead (3 step) prediction, to investigate the different types of networks in different applications.

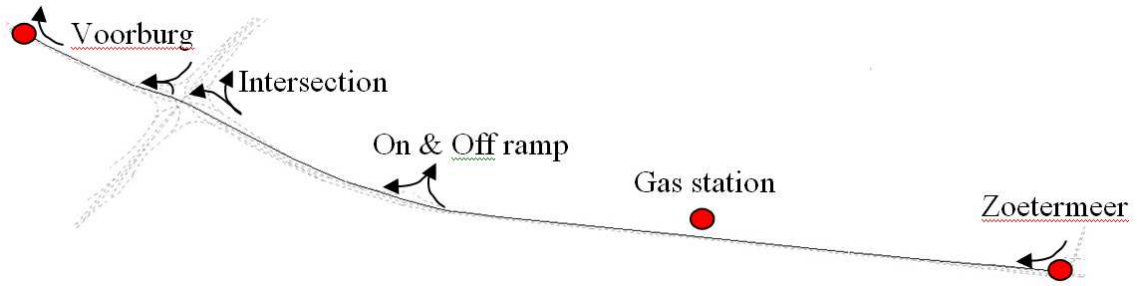


Figure 5.2: The A12 motorway from Zoetermeer to The Hague

5.3.1 Data

At both the on ramp and the off ramp license plate cameras are placed that record each vehicles license plate. Individual travel times based on matches of license plates were available for 95 days in the winter and spring of 2007, the same data set that was also used in Chapter 3 and Chapter 4. After filtering the data for outliers the travel times were aggregated to 5-minute time periods. A total of 39 morning peak periods of 3.5 hours each were selected from the data set, discarding those days that either were absent of congestion or contained failing loop detectors.

As input to the neural networks, 19 double loop detectors are available, reporting arithmetic mean speeds (i.e. time mean speeds) and total flows every minute. Due to the inherent bias in time mean speeds when used as a proxy for space mean speeds, these were corrected to space mean speeds using an estimate for the variance of the speeds in the one minute interval, as described in (van Lint, 2004; van Hinsbergen et al., 2008a). The input data were then aggregated to 5-minute periods.

To warrant a proper functioning of the context layer, the context layer in the SSNNs was reset to its initial values, $C = 0$, whenever a new day started in the data set, and it was ensured that the last data points of a previous day as well as the first points in a new day were all in free flow conditions.

5.3.2 Stopping criterion

When training the SSNNs, a stopping criterion needs to be formulated. For this purpose, the evidence can be used as was discussed in 4.2.3. However, to be able to make a fair comparison between the Bayesian and the non-Bayesian networks, a stopping criterion based on the training error E_D was used. After conducting extensive experiments, it was found as a rule of thumb that if the decrease of the data error part E_D relative to the starting value of E_D with random initial weights in 10 epochs dropped below 0.2%, the network was close to its minimum test error (and has approximately the best possible generalization performance).

5.4 Results

Table 5.1 shows several results for the two prediction horizons. The first column shows the average number of epochs. The total training time in the second column is the total time required to train all networks for all cross validations. It can be seen that the SSNN need considerably more training time due to the fact that they require more epochs to obtain a good fit and due to the more complex gradient and Hessian computations. The next two columns show the cross-correlation coefficients averaged over all 5 cross-validations between the test error (in RMSE) and the training error E_D (also in RMSE) and the test error and the Bayesian evidence. It can be seen that for the case under investigation the correlation between the training and test error is stronger than that between the evidence and the test error. The last two columns show the average Mean Average Percentage Error and Root Mean Square Error of the 5 networks over all cross validations. It can be seen that the prediction error for the 5-minute-ahead prediction is very similar for all types of networks. On such a short term prediction, the recurrent nature of the SSNN does not add to the prediction accuracy; even a slight decrease of accuracy can be seen, which can be explained by the fact that the SSNNs contain more parameters and therefore are harder to train. For the 15-minute-ahead prediction, the recurrent layer does have effect on the prediction accuracy, resulting in lower MAPE and lower RMSE.

Table 5.1: Results with 5-minute ahead prediction and 15-minute ahead prediction

Method	Mean epochs	Training time (s)	Correlation RMSE/E_D	Correlation RMSE/Ev.	Mean MAPE	Mean RMSE (S)
5 minute prediction horizon						
FFNN	43	1409	0.85	-	12.4%	104.3
SSNN	85	11499	0.79	-	12.5%	108.4
Bayes FFNN	44	4665	0.8	-0.59	12.7%	106.0
Bayes SSNN	80	21265	0.61	-0.42	12.7%	110.1
15 minute prediction horizon						
FFNN	47	2009	0.86	-	17.6%	141.6
SSNN	93	19598	0.46	-	16.7%	141.0
Bayes FFNN	50	6468	0.82	-0.50	17.6%	140.0
Bayes SSNN	86	29322	0.57	-0.53	16.4%	139.6

Table 5.2 shows the effect of forming a committee on the prediction accuracy. Two ranking mechanisms were used: the training error and the evidence. To investigate the effectiveness of the evidence versus that of the training error, a committee size of 3 networks was chosen. It can be seen that all committee errors are considerably lower than the average errors of the individual predictions (compare with Table 5.1). The ranking based on the training error performs slightly better than when the evidence is used as a

ranking mechanism. This is expected, because the correlation between training error and test error is higher than the evidence and test error.

Table 5.2: Committee errors

Method	*Committee ranked on E_D*		*Committee ranked on Evidence*	
	MAPE	RMSE (s)	MAPE	RMSE (s)
5 minute prediction horizon				
FFNN	10.8%	92.8	-	-
SSNN	10.6%	95.1	-	-
Bayes FFNN	11.2%	95.0	11.3%	96.6
Bayes SSNN	11.1%	97.4	11.3%	98.6
15 minute prediction horizon				
FFNN	16.2%	133.2	-	-
SSNN	14.8%	129.4	-	-
Bayes FFNN	16.1%	132.6	16.4%	135.0
Bayes SSNN	14.8%	126.8	14.7%	127.1

5.5 Discussion and conclusion

In this research, the Bayesian framework for neural networks has been adapted to recurrent neural networks, and State Space Neural Networks in particular. With this result, recurrent neural networks and feed-forward neural networks can now both be applied in the Bayesian framework. The Bayesian evidence that results from this framework equips the modeler with a natural way to select high-accuracy neural networks from a large pool of trained networks without the use of additional validation data sets. Moreover, on each prediction an error bar can be calculated which provides an estimation of the prediction uncertainty. With this approach, the networks can also be combined into a committee of prediction models, which results in lower test error (see Table 5.2) and more accurate error bars (Bishop, 1995).

The adaptation scheme of the hyperparameters, resulting from the Bayesian analysis, did not prove to be beneficial for the prediction results on the 1-step-ahead prediction, and did show a slight improvement in the 3-step-ahead prediction over the use of a fixed set of hyperparameters. This is an indication that the Bayesian adaptation of the hyperparameters becomes more important when the system under investigation becomes more complex. It is expected that on larger datasets that contain more diverse and complex circumstances, the effect of more advanced smoothing (by continuously adapting the hyperparameters) will be positive in terms of prediction accuracy.

It is clear from the results that there are still issues to be resolved in the Bayesian framework. Table 5.1 shows that the cross-correlation between the model-evidence and

the test error is reasonable, but worse than that of the training error. As noted by Bishop (1995), the weighting coefficient μ_q in equation 4.24 theoretically represents the posterior probability of the model q, which can be obtained through the evidence factors calculated for all models. Using this evidence-based weighting mechanism is only expected to improve results if the evidence and test error (a proxy for the generalization error) show high correlation. A second and related concern is that with a lower correlation between model evidence and test-error the optimal committee size is expected to increase, which leads to large calculation times. Clearly, improving the correlation between evidence and test error would lead to both better performance as well as smaller and hence more practical committees. Fortunately, there are various ways to improve the correlation between evidence and the generalization error, the most important being:

- Increasing the sizes of both the training and the test sets, which has two possible beneficial effects: it would increase the probability that the training and test set have identical statistical properties and that the test error is indicative of the true generalization error (the error on the entire population). Only 6 days were used for testing in the cross-validation approach.

- A moderate / weak correlation between evidence and generalization error is an indication of inconsistencies in the models. The evidence provides a quantitative tool to assess possible improvements, such as weight pruning to improve the models structure (Thodberg, 1993), different types of input data, other transfer functions or even completely different mathematical structures. These improvements are expected to not only improve the evidence/generalization error relationship but also to improve overall prediction results.

- In this study the outer product approximation was used for training and for evidence determination to speed up the calculations. The use of the exact Hessian may improve the estimation of the evidence.

Finally, error bars, which can prove to be very useful for various applications, can only be obtained through Bayesian analysis. Now that the Bayesian framework can be applied to both FFNN and SSNN, future studies will focus on the accuracy of the error bars and its value for the individual road user.

Other directions for future research include the application of Bayesian combined NNs on more complex situations, such as travel time prediction on shorter and longer freeway sections and in urban networks. More technically, the relevance of the exact versus the outer product approximation of the Hessian (needed to calculate model evidence) should be more thoroughly investigated. Furthermore, it will be worthwhile to implement other neural network structures or other types of prediction models into the Bayesian framework, to increase the heterogeneity of the committee, which is expected to benefit the generalization results (van Hinsbergen et al., 2008a).

Chapter 6

Bayesian calibration of the Extended Kalman Filter

This chapter is an edited version of van Hinsbergen, C. P. I., Schreiter, T., van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2010b). Online estimation of kalman filter parameters for traffic state estimation. In *Proceedings of the Seventh Triennial Symposium on Transportation Analysis (TRISTAN VII)*. Tromso, Norway.

Online traffic state estimation, which can be used to inform road users or as input to traffic state prediction or route guidance, has been the subject of study for many researchers in recent years, for various applications in the fields of advanced traffic information systems or dynamic traffic management. Online state estimation requires two components: traffic data, and a traffic simulation model. When combining these two, it is important to consider that both the data collection and the model are imperfect and thus contain noise. The Extended Kalman Filtering (EKF) provides a convenient formalism to use such traffic models to estimate (partially) unobserved state variables from observed sensor data. One of the difficulties in applying the EKF for this purpose is choosing appropriate (and possibly time varying) values for the noise parameters which govern how well the EKF / traffic model is able to track state variables from observed data. In this chapter a two-stage Bayesian framework is proposed which enables simultaneous recursive estimation of both state variables and the noise parameters. First, a posterior distribution of state variables is calculated using the Extended Kalman Filter equations. Second, optimal values for the measurement and process noise parameters are found using the results of the first step. In a small-scale simulation study the approach is verified. These preliminary results suggest that this approach potentially leads to superior state estimation results compared to ad hoc setting of the noise parameters.

6.1 Introduction

Online traffic simulation models have been subject of study for many researchers in the last years, for various applications in the field of advanced traffic information systems (ATIS) or dynamic traffic management (DTM) (Lebacque, 1996; Ben-Akiva et al., 2001; Mahmassani, 2001; Wang and Papageorgiou, 2005; Zuurbier et al., 2006; Barceló et al., 2007; van Hinsbergen et al., 2008f). One of these applications is online state estimation. In those applications, the current state of traffic is estimated using two components: traffic data from online data collection equipment at certain points in the network, and a traffic simulation model that estimates the state of traffic everywhere in the network based on these data and based on fundamental laws that describe how traffic progresses over the network. Online estimates of the traffic state can be used to inform road users about the current state to allow them to anticipate, or can be used as input to the prediction of the future traffic state or for route guidance applications such as in Zuurbier (2010).

To be able to accurately estimate the current state of traffic, a traffic model has to be chosen. In this study, the first order model is chosen as a modeling paradigm, as it has proven to perform well without introducing too many parameters, opposed to second- or higher-order models (Daganzo, 1995b). The first order traffic flow model is based on the kinematic wave theory of Lighthill and Whitham (1955) and Richards (1956) and mainly applies to modeling freeway networks, although extensions have been proposed to be able to apply it to urban networks (van Hinsbergen et al., 2009b). One common numerical solution of this model is to discretize the network into segments or cells and model the traffic in discretized time steps. The state estimation problem then consists of finding the state, which is uniquely described by the density in each cell, at each time step; other variables such as speed and flow in each cell can be obtained through a fundamental diagram. At each time step, a numerical scheme is used to calculate the fluxes between two cells, usually by applying the Godunov scheme (Lebacque, 1996), or as recently proposed, using Lagrangian coordinates (Leclerq et al., 2007). As the Godunov scheme is one of the most widely applied numerical solution, in this study it is also applied. Application of the ideas proposed in this contribution to other numerical schemes is straightforward.

One of the challenges in estimating the traffic state in an online setting is how to use traffic data to correct the estimates made by the model. One important point to consider is that both the data contains noise (the measurement equipment is imperfect) as well as the model (the model describes the traffic process imperfectly). Any solution for combining the model with measurements has to be able to deal with these noise processes. Specifically, there must be a balance between the trust placed on the model and the trust placed on the measurements for the estimates to be accurate and smooth. The Extended Kalman Filter (EKF) framework as described by Wang and Papageorgiou (2005) is an appropriate solution, as it is fast and results in smooth estimates of the state, taking into account both the error distribution of the measurements as well as the error distribution of the model.

The EKF has been successfully applied to the first order traffic model (Zuurbier et al., 2006; Tampère and Immers, 2007; van Hinsbergen et al., 2008f).

However, one of the great difficulties of applying the EKF is that choices have to be made for the noise parameters: the covariance of the measurement noise and of the process noise. The values of the noise parameters heavily influence the accuracy of the estimates produced by the model with the EKF. Although in some cases, through specifications of the manufacturer, the noise distribution produced by the measurement equipment may be known, the size of the process noise is never known beforehand. Up to now, assumptions are usually made based on trial and error or experience.

In this study a consistent methodology is proposed to set the size of the covariances of both noise models, using Bayesian inference theory. The work is based on a similarity between training neural networks (specifically the work of Bishop (1995) and Mackay (1995)) and Kalman filtering; for more information on these similarities, see Haykin (2001). To derive expressions for the noise parameters, first the Kalman Filter needs to be assessed from a Bayesian point of view, where it is shown to be equal to a Bayesian Maximum A Posteriori (MAP) approach. Next, a Bayesian choice can be made for the values of the covariance matrices using the outcomes of the MAP approach. This is the main contribution of this chapter. In a small-scale experiment it will then be shown that these choices lead to good performance when compared to choosing fixed values. Finally, the discussion and conclusion are presented.

6.2 Methodology: Bayesian estimation of noise parameters

Define the state-space equation that describes the state vector $\boldsymbol{x}[k]$ of size $N \times 1$ as a function of the previous state $\boldsymbol{x}[k-1]$ and a noise vector $\boldsymbol{w}[k]$:

$$\boldsymbol{x}[k] = f(\boldsymbol{x}[k-1]) + \boldsymbol{w}[k] \tag{6.1}$$

and the measurement (observation) equation that describes the measurement vector $\boldsymbol{z}[k]$ of size $M \times 1$ as a function of the state $\boldsymbol{x}[k]$ with measurement noise $\boldsymbol{v}[k]$

$$\boldsymbol{z}[k] = h(\boldsymbol{x}[k]) + \boldsymbol{v}[k] \tag{6.2}$$

In this contribution, the state vector $\boldsymbol{x}[k]$ equals a vector of all densities in all cells of the discretized model; the function f equals the first order model solved by the Godunov scheme; the measurement vector $\boldsymbol{z}[k]$ consists of speed and/or flow measurements that can be translated to densities (and vice-versa) using the fundamental diagram, which is represented by the function h. This is similar to the approach in (Wang and Papageorgiou, 2005), except that here the parameters of the fundamental diagram are not included in the

state but are taken a fixed value for the sake of simplicity.

The process noise $\boldsymbol{w}[k]$ and measurement noise $\boldsymbol{v}[k]$ are assumed zero mean white Gaussian noise with covariance matrix $\boldsymbol{Q}[k]$ and $\boldsymbol{R}[k]$ respectively and are assumed to be independent of each other. Furthermore, each element of the state or measurement vector is assumed to be coming from a single distribution with variance $1/\alpha[k]$ and $1/\beta[k]$ respectively, such that $\boldsymbol{Q}[k] = 1/\alpha[k]\boldsymbol{I}_N$ where $\boldsymbol{I}_N$ is the identity matrix of size $N \times N$, and $\boldsymbol{R}[k] = 1/\beta[k]\boldsymbol{I}_M$ where $\boldsymbol{I}_M$ is the identity matrix of size $M \times M$. These assumption are plausible in cases where the state is described everywhere in the same quantity (for example density in veh/km) as is the case in this contribution, and if all measurements are of the same quantity (for example speed in km/h) and are produced by the same type of measurement equipment.

If a traffic simulation model is operated in an online mode, data is obtained sequentially over time. Each time k when new data arrives, the state vector can be updated given the previous estimate of the state at $k-1$, denoted by $\hat{\boldsymbol{x}}[k-1]$, and the data obtained so far, denoted by $\boldsymbol{Z}[k] = (\boldsymbol{z}[k], \boldsymbol{z}[k-1], \ldots, \boldsymbol{z}_1)$. In further derivations, the matrix $\boldsymbol{Z}[k]$ is split into two parts: $\boldsymbol{Z}[k-1]$ to denote all data used up to time step $k-1$, and $\boldsymbol{z}[k]$ to denote the last vector of measurements. This is done to reflect the fact that $\boldsymbol{z}[k-1], \boldsymbol{z}[k-2], \ldots, \boldsymbol{z}[1]$ have already been used to estimate the state at $k-1$ and only the last data vector $\boldsymbol{z}[k]$ will be used to update the traffic state, as is reflected by (6.1) and (6.2).

This methodology section consists of two parts. In the second part, the main contribution of this chapter is presented: a methodology to choose values for the noise parameters of the EKF (i.e. the values of the covariance matrices of the process and measurement noise). To be able to do so, in the first part the Extended Kalman Filter (EKF) will be shown to be equal to a Bayesian Maximum A Posteriori (MAP) approach.

6.2.1 Bayesian derivation of the EKF

In (Ho and Lee, 1964; Chen, 2003) it is shown that the EKF can be given a Bayesian interpretation. In this section the Bayesian interpretation is briefly repeated, as it is a necessary starting point for the derivation of the Bayesian choice for the process and noise covariance matrices which is the main goal of this contribution.

Consider the state estimate at time k, denoted by $\hat{\boldsymbol{x}}[k]$, that is based on the first order model and the data vector $\boldsymbol{z}[k]$ that was retrieved at time k. In probabilistic terms, the interest is in finding the probability $p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1])$. This probability is called the posterior probability of the state vector, and describes the probability of the state $\boldsymbol{x}[k]$ given all data $\boldsymbol{z}[k]$, $\boldsymbol{Z}[k-1]$ that were obtained so far. In order to obtain this posterior, Bayes rule can be applied:

$$p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) = \frac{p(\boldsymbol{z}[k]|\boldsymbol{x}[k], \boldsymbol{Z}[k-1])p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1])}{p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1])} \tag{6.3}$$

The right hand side of (6.3) consists of three terms: a likelihood function $p(\boldsymbol{z}[k]|\boldsymbol{x}[k], \boldsymbol{Z}[k-1])$, a prior $p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1])$ and a normalization term $p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1])$. Note that all terms have been conditioned on all data $\boldsymbol{Z}[k-1]$ that have already been used in the sequential process prior to obtaining $\boldsymbol{z}[k]$. It will now be shown that by defining equations for the prior and the likelihood an expression for the posterior can be obtained.

Definition of the prior $p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1])$

As a starting point, consider the fact that at time $k-1$, the available knowledge about $\boldsymbol{x}[k-1]$ is the previous estimate of the state $\hat{\boldsymbol{x}}[k-1]$ and an estimate of its covariance matrix, denoted by $\hat{\boldsymbol{P}}[k-1]$. For both $\hat{\boldsymbol{x}}[k-1]$ and $\hat{\boldsymbol{P}}[k-1]$ equations will be determined later (note that at the first time step $k=0$, initial estimates $\hat{\boldsymbol{x}}[0]$ and $\hat{\boldsymbol{P}}[0]$ need to be defined; in the experiment section, it will show that the Bayesian procedure for finding amongst others $\hat{\boldsymbol{P}}[k]$ is insensitive to these initial values). Moving one time step forward, an estimate of $\boldsymbol{x}[k]$ without any new data available yet equals

$$\hat{\boldsymbol{x}}^{-}[k] = f(\hat{\boldsymbol{x}}[k-1]) \tag{6.4}$$

This is the prior estimate of the state. The prior of $\boldsymbol{x}[k]$ is defined to be a Gaussian, i.e.:

$$p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1]) = \frac{1}{Z_p}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}[k]-\hat{\boldsymbol{x}}^{-}[k]\right)^T\left(\hat{\boldsymbol{P}}^{-}[k]\right)^{-1}\left(\boldsymbol{x}[k]-\hat{\boldsymbol{x}}^{-}[k]\right)\right) \tag{6.5}$$

with Z_p given by Bishop (1995)

$$\begin{aligned} Z_p &= \int \exp\left(-\frac{1}{2}\left(\boldsymbol{x}[k]-\hat{\boldsymbol{x}}^{-}[k]\right)^T\left(\hat{\boldsymbol{P}}^{-}[k]\right)^{-1}\left(\boldsymbol{x}[k]-\hat{\boldsymbol{x}}^{-}[k]\right)\right) d\boldsymbol{x}[k] \\ &= (2\pi)^{\frac{N}{2}}\left|\hat{\boldsymbol{P}}^{-}[k]\right|^{\frac{1}{2}} \end{aligned} \tag{6.6}$$

where $\hat{\boldsymbol{P}}^{-}[k]$ is an estimate of the prior covariance, which can be approximated by linearizing $\boldsymbol{x}[k]$ around the mean of the prior $\hat{\boldsymbol{x}}^{-}[k]$ (Haykin, 2001), i.e.

$$\boldsymbol{x}[k] \approx \hat{\boldsymbol{x}}^{-}[k] + \boldsymbol{J}[k-1]\left(\boldsymbol{x}[k-1]-\hat{\boldsymbol{x}}[k-1]\right) + \boldsymbol{w}[k] \tag{6.7}$$

where $\boldsymbol{J}[k]$ is the Jacobian $\nabla_{\boldsymbol{x}[k]} f|_{\hat{\boldsymbol{x}}[k]}$. Substitution of (6.7) into (6.5) leads to an estimate for the prior covariance matrix (Chen, 2003):

$$\hat{\boldsymbol{P}}^{-}[k] = \boldsymbol{J}[k-1]\hat{\boldsymbol{P}}[k-1]\boldsymbol{J}[k-1]^T + \boldsymbol{Q}[k] \tag{6.8}$$

Derivation of the likelihood $p(\boldsymbol{z}[k]|\boldsymbol{x}[k], \boldsymbol{Z}[k-1])$

The likelihood essentially determines the overall measurement noise model. As the measurement noise $\boldsymbol{v}[k]$ has already been assumed to be Gaussian with mean 0 and covariance matrix $\boldsymbol{R}[k]$, the likelihood can be described by:

$$p(\boldsymbol{z}[k]|\boldsymbol{x}[k], \boldsymbol{Z}[k-1]) = \frac{1}{Z_l} \exp\left(-\frac{1}{2}\left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right)^T \left(\boldsymbol{R}[k]\right)^{-1} \left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right)\right) \tag{6.9}$$

with Z_l given by

$$\begin{aligned} Z_l &= \int \exp\left(-\frac{1}{2}\left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right)^T \left(\boldsymbol{R}[k]\right)^{-1} \left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right)\right) d\boldsymbol{x}[k] \\ &= (2\pi)^{\frac{M}{2}} \left|\boldsymbol{R}[k]\right|^{\frac{1}{2}} \end{aligned} \tag{6.10}$$

Derivation of the posterior $p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1])$

Substituting (6.5) and (6.9) into (6.3), the posterior can now be written as

$$p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) = \frac{1}{Z_s} \exp\left(-E(\boldsymbol{x}[k])\right) \tag{6.11}$$

with $E(\boldsymbol{x}[k])$ defined as

$$E(\boldsymbol{x}[k]) = E_p(\boldsymbol{x}[k]) + E_l(\boldsymbol{x}[k]) \tag{6.12}$$

$$E_p(\boldsymbol{x}[k]) = \frac{1}{2}\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right)^T \left(\hat{\boldsymbol{P}}^{-}[k]\right)^{-1} \left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right) \tag{6.13}$$

$$E_l(\boldsymbol{x}[k]) = \frac{1}{2}\left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right)^T \left(\boldsymbol{R}[k]\right)^{-1} \left(\boldsymbol{z}[k] - h(\boldsymbol{x}[k])\right) \tag{6.14}$$

and with Z_s given by

$$Z_s = \int \exp\left(-E(\boldsymbol{x}[k])\right) d\boldsymbol{x}[k] \tag{6.15}$$

The maximum of this posterior can be found by maximizing the logarithm of the posterior, i.e. a MAP approach (Chen, 2003):

$$\begin{aligned} \hat{\boldsymbol{x}}[k] &= \arg\max_{\boldsymbol{x}[k]} \ln\left(p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1])\right) \\ &= \arg\min_{\boldsymbol{x}[k]} E(\boldsymbol{x}[k]) \end{aligned} \tag{6.16}$$

Note that in (6.16), the normalizing constant Z_s has been omitted as it is independent of $\boldsymbol{x}[k]$ and therefore does not influence the solution. In order to find the minimum for

$E(\boldsymbol{x}[k])$, the condition

$$\nabla_{\boldsymbol{x}[k]} E(\boldsymbol{x}[k]) = 0 \tag{6.17}$$

needs to be solved. Substituting (6.12) into (6.17), and approximating the measurement equation by its linearization around the prior estimate $\hat{\boldsymbol{x}}^-[k]$, i.e.

$$h(\boldsymbol{x}[k]) \approx \hat{\boldsymbol{z}}^-[k] + \boldsymbol{H}[k]\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}^-[k]\right) + \boldsymbol{v}[k] \tag{6.18}$$

where $\boldsymbol{H}[k]$ is the Jacobian $\nabla_{\boldsymbol{x}[k]} h|_{\hat{\boldsymbol{x}}^-[k]}$ and $\hat{\boldsymbol{z}}^-[k] = h(\hat{\boldsymbol{x}}^-[k])$, and solving for $\boldsymbol{x}[k]$ leads to the expression (Chen, 2003):

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\left(\boldsymbol{z}[k] - \hat{\boldsymbol{z}}^-[k]\right) \tag{6.19}$$

$$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{H}[k]^T\left(\boldsymbol{R}[k] + \boldsymbol{H}[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{H}[k]^T\right)^{-1} \tag{6.20}$$

which is exactly the EKF (Kalman, 1960; Haykin, 2001); $\hat{\boldsymbol{x}}[k]$ equals the estimated mean of the posterior, while the estimate of the covariance matrix $\hat{\boldsymbol{P}}[k]$ of the posterior equals

$$\hat{\boldsymbol{P}}[k] = \left(\boldsymbol{I}_N - \boldsymbol{K}[k]\boldsymbol{H}[k]\right)\hat{\boldsymbol{P}}^-[k] \tag{6.21}$$

where $\boldsymbol{I}_N$ is the identity matrix of size $N \times N$. This equation can be obtained by substitution of (6.19) in the posterior (6.11) and rearranging the resulting terms such that the posterior covariance matrix appears (Ho and Lee, 1964; Chen et al., 2003).

Note that the posterior of time k is used to determine the prior at time $k+1$, in equations (6.4) and (6.8). This indicates the recursive (state-space) nature of the EKF procedure. Moreover, only the previous state vector and covariance matrix need to be retained in memory; all previous estimates can be discarded, which is a desirable property for any computational implementation.

6.2.2 Bayesian derivation of $\alpha[k]$ and $\beta[k]$

In common applications, suitable values for the noise levels $\boldsymbol{R}[k] = 1/\beta[k]\boldsymbol{I}_M$ and $\boldsymbol{Q}[k] = 1/\alpha[k]\boldsymbol{I}_N$ are chosen based on experience or trial and error. However, using the same Bayesian framework of the previous section, proper values for $\alpha[k]$ and $\beta[k]$ can be found. For this, the posterior distribution of $\boldsymbol{x}[k]$ is used. The noise parameters $\alpha[k]$ and $\beta[k]$ are sometimes called hyperparameters (Mackay, 1995; Bishop, 1995) or scales (Thodberg, 1993), as their distribution controls other distributions (those of the state).

The correct Bayesian treatment for parameters such as $\alpha[k]$ and $\beta[k]$, whose values are

unknown, is to make their dependency explicit and integrate them out of any predictions

$$\begin{aligned} p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) &= \int\int p(\boldsymbol{x}[k], \alpha[k], \beta[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) d\alpha[k] d\beta[k] \\ &= \int\int p(\boldsymbol{x}[k]|\alpha[k], \beta[k], \boldsymbol{z}[k], \boldsymbol{Z}[k-1]) \\ &\quad \times p(\alpha[k], \beta[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) d\alpha[k] d\beta[k] \end{aligned} \tag{6.22}$$

If it is assumed that the posterior probability distribution $p(\alpha[k], \beta[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1])$ is sharply peaked around its most probable values $\hat{\alpha}[k]$ and $\hat{\beta}[k]$, then (6.22) can also be written as (Mackay, 1995; Bishop, 1995):

$$\begin{aligned} p(\boldsymbol{x}[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) \approx & p(\boldsymbol{x}[k]|\hat{\alpha}[k], \hat{\beta}[k], \boldsymbol{z}[k], \boldsymbol{Z}[k-1]) \\ & \times \int\int p(\alpha[k], \beta[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) d\alpha[k] d\beta[k] \\ = & p(\boldsymbol{x}[k]|\hat{\alpha}[k], \hat{\beta}[k], \boldsymbol{z}[k], \boldsymbol{Z}[k-1]) \end{aligned} \tag{6.23}$$

Equation (6.23) states that the values $\hat{\alpha}[k]$ and $\hat{\beta}[k]$ should be found that maximize the posterior probability, and the remaining calculations should be performed with $\alpha[k]$ and $\beta[k]$ set to these values.

In order to find these most probable values $\hat{\alpha}[k]$ and $\hat{\beta}[k]$, the Bayesian MAP can be applied a second time at the level of the hyperparameters. For this, the posterior distribution of $\alpha[k]$ and $\beta[k]$ is evaluated. This posterior can be found using Bayes rule again:

$$p(\alpha[k], \beta[k]|\boldsymbol{z}[k], \boldsymbol{Z}[k-1]) = \frac{p(\boldsymbol{z}[k]|\alpha[k], \beta[k], \boldsymbol{Z}[k-1]) p(\alpha[k], \beta[k]|\boldsymbol{Z}[k-1])}{p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1])} \tag{6.24}$$

In (6.24), the likelihood term can be recognized as the denominator $p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1])$ of (6.3) conditioned on $\alpha[k]$ and $\beta[k]$. This is a very important feature, as it allows for the derivation of the values for $\alpha[k]$ and $\beta[k]$ using the current posterior distribution of the state at time k. In Bayesian inference, the denominator of (6.24) is called the evidence for $\alpha[k]$ and $\beta[k]$.

Considering the fact that there is very little knowledge of suitable values for $\alpha[k]$ and $\beta[k]$, a flat prior is chosen for $\alpha[k]$ and $\beta[k]$. Therefore only the evidence is used to assign a preference to alternative values for $\alpha[k]$ and $\beta[k]$ (Bishop, 1995; Mackay, 1995).

Derivation of the evidence for $\alpha[k]$ and $\beta[k]$

Using Bayesian inference, in this section it will be shown that the value for $\beta[k]$ can be updated at each step k by the expression

$$\hat{\beta}[k] = \frac{M}{\left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right)^T \left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right) + Tr(\boldsymbol{A}[k]^{-1}\boldsymbol{H}[k]^T\boldsymbol{H}[k])} \tag{6.25}$$

The value for $\alpha[k]$ can be chosen fixed, as (6.25) expresses $\beta[k]$ as a function of $\alpha[k]$ through the Hessian $\boldsymbol{A}$ (see (6.29)); $\hat{\beta}[k]$ will therefore be optimal for a given value of $\alpha[k]$. Because $\alpha[k]$ and $\beta[k]$ depend on each other, one of the two can be fixed while the other can be estimated using 6.25, which thus provides a solution for the problem of finding appropriate values for $\boldsymbol{Q}[k]$ and $\boldsymbol{R}[k]$.

Equation (6.25) can be interpreted as follows. The term $(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k]))$ represents a value for the difference between the measurement and the mapping of the posterior estimate of the state to the measurement. As the estimate $\hat{\boldsymbol{x}}[k]$ is optimal (in the sense of MAP) in case the model is linear (a Kalman Filter) and is the best estimate that be made in case the model is linearized (an Extended Kalman Filter), this difference can be seen as an estimate of the noise in the data. A larger value for $(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k]))$ leads to smaller $\beta[k]$ and thus larger values of the elements of $\boldsymbol{R}[k] = 1/\beta[k]\boldsymbol{I}_M$. A larger $\boldsymbol{R}[k]$ in turn causes the Kalman gain to become smaller, leading to smaller corrections. This is a very intuitive result: the higher the estimated noise on the data, to lower the trust in the data should be and the more trust should be placed on the model, and vice versa.

It will now be shown how the key result (6.25) was obtained. First, consider the fact that the evidence for $\alpha[k]$ and $\beta[k]$ equals the denominator of (6.3), and that the evidence can thus be written as

$$\begin{aligned} p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = \int & p(\boldsymbol{z}[k]|\boldsymbol{x}[k], \boldsymbol{Z}[k-1], \alpha[k], \beta[k]) \\ & \times p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) d\boldsymbol{x}[k] \end{aligned} \tag{6.26}$$

where the knowledge that the likelihood $p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \boldsymbol{x}[k])$ is independent of $\alpha[k]$ and that the prior $p(\boldsymbol{x}[k]|\boldsymbol{Z}[k-1])$ is independent of $\beta[k]$ is used. Using (6.6), (6.10) and (6.15) this is equal to

$$p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = \frac{Z_s}{Z_p Z_l} \tag{6.27}$$

Expressions for Z_p and Z_l were already found in (6.6) and (6.10); the integral Z_s still needs to be evaluated. As this cannot easily be evaluated analytically, an approximation

of the posterior distribution is made that will allow for a solution. Considering the Taylor expansion of $E(\boldsymbol{x}_k)$ around its minimum value $\hat{\boldsymbol{x}}[k]$ where terms up to second order are retained

$$E(\boldsymbol{x}[k]) \approx E(\hat{\boldsymbol{x}}) + \frac{1}{2}\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right)^T \boldsymbol{A}[k]\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right) \tag{6.28}$$

with $\boldsymbol{A}[k]$ the Hessian

$$\begin{aligned}\boldsymbol{A}[k] &= \nabla\nabla_{\boldsymbol{x}[k]} E(\boldsymbol{x}[k])|_{\hat{\boldsymbol{x}}[k]} \\ &= \nabla\nabla_{\boldsymbol{x}[k]} E_p(\boldsymbol{x}[k])|_{\hat{\boldsymbol{x}}[k]} + \nabla\nabla_{\boldsymbol{x}[k]} E_l(\boldsymbol{x}[k])|_{\hat{\boldsymbol{x}}[k]} \\ &= \left(\hat{\boldsymbol{P}}^-[k]\right)^{-1} + \beta[k]\boldsymbol{H}[k]^T\boldsymbol{H}[k]\end{aligned} \tag{6.29}$$

Substituting (6.28) into (6.15) leads to a Gaussian form, for which the integral can easily be evaluated (Bishop, 1995)

$$\begin{aligned}Z_s &\approx \int \exp\left(-E(\hat{\boldsymbol{x}}[k]) - \frac{1}{2}\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right)^T \boldsymbol{A}[k]\left(\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]\right)\right) d\boldsymbol{x}[k] \\ &= \exp\left(-E(\hat{\boldsymbol{x}}[k])\right)(2\pi)^{\frac{N}{2}}\left|\boldsymbol{A}[k]\right|^{-\frac{1}{2}}\end{aligned} \tag{6.30}$$

Now that expressions have been found for Z_p, Z_l and Z_s, an expression for the evidence is found

$$p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = \exp\left(-E(\hat{\boldsymbol{x}}[k])\right)(2\pi)^{\frac{S}{2}}\left|\boldsymbol{A}[k]\right|^{-\frac{1}{2}}\left|\frac{1}{\beta[k]}\boldsymbol{I}_M\right|^{-\frac{1}{2}}\left|\hat{\boldsymbol{P}}^-[k]\right|^{-\frac{1}{2}} \tag{6.31}$$

The evidence can be used to determine the most probable values $\hat{\alpha[k]}$ and $\hat{\beta[k]}$ by applying the MAP principle, i.e. by solving the conditions $\nabla_{\alpha[k]} \ln p(\boldsymbol{z}[k], \boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = 0$ and $\nabla_{\beta[k]} \ln p(\boldsymbol{z}[k], \boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = 0$ respectively. To do so, first the log of the evidence is evaluated:

$$\ln p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = -E(\hat{\boldsymbol{x}}[k]) - \frac{S}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\boldsymbol{A}[k]\right| + \frac{M}{2}\ln\beta[k] - \frac{1}{2}\ln\left|\hat{\boldsymbol{P}}^-[k]\right| \tag{6.32}$$

The derivatives of (6.32) for all parts dependent on $\beta[k]$ to $\beta[k]$ are

$$\nabla_{\beta[k]} E(\hat{\boldsymbol{x}}[k]) = \frac{1}{2}\left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right)^T\left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right) \tag{6.33}$$

$$\nabla_{\beta[k]} \ln\left|\boldsymbol{A}[k]\right| = Tr\left(\boldsymbol{A}[k]^{-1}\nabla_{\beta[k]}\boldsymbol{A}[k]\right) = Tr\left(\boldsymbol{A}[k]^{-1}\boldsymbol{H}[k]^T\boldsymbol{H}[k]\right) \tag{6.34}$$

$$\nabla_{\beta[k]} \frac{M}{2}\ln\beta[k] = \frac{M}{2\beta[k]} \tag{6.35}$$

where Tr equals the Trace operator. Putting everything together now yields

$$\nabla_{\beta[k]} \ln p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k]) = \frac{1}{2} \left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right)^T \left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right) - \frac{1}{2} Tr\left(\boldsymbol{A}[k]^{-1} \boldsymbol{H}[k]^T \boldsymbol{H}[k]\right) + \frac{M}{2\beta[k]} \quad (6.36)$$

Setting (6.36) to zero and solving for $\beta[k]$ results in

$$\hat{\beta}[k] = \frac{M}{\left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right)^T \left(\boldsymbol{z}[k] - h(\hat{\boldsymbol{x}}[k])\right) + Tr(\boldsymbol{A}[k]^{-1} \boldsymbol{H}[k]^T \boldsymbol{H}[k])} \quad (6.37)$$

which is the result of (6.25) that was presented earlier.

Finally, the choice of a value for $\alpha[k]$ needs to be considered. The derivative $\nabla_{\alpha[k]} \ln p(\boldsymbol{z}[k]|\boldsymbol{Z}[k-1], \alpha[k], \beta[k])$ cannot be solved for $\alpha[k]$ as was done for $\beta[k]$ because of the difference in nature in the appearance of the hyperparameters in the terms $\frac{1}{2} \ln |\hat{\boldsymbol{P}}^-[k]|$ and $\frac{M}{2} \ln \beta[k]$ respectively. However, the value of $\alpha[k]$ depends on $\beta[k]$ and vice versa. As a heuristic, a fixed value for $\alpha[k]$ is chosen and $\beta[k]$ is varied. As $\beta[k]$ is expressed as a function of $\alpha[k]$ (through the term of the Hessian $\boldsymbol{A}[k]$), over time, $\hat{\beta}[k]$ will thus become optimal given the chosen fixed value for $\alpha[k]$.

6.3 Experiment

To illustrate the impact of the Bayesian choice for the EKF parameters, a small-scale case study is performed. The traffic network as shown in Figure 6.1 is simulated with *JDSMART* with a time step of two seconds with link capacities as shown in Figure 6.1a. A total of 600 time steps are simulated, with four different demand levels at the two origins O_1 and O_2 and four different turn fractions at the node A. Each time step the speeds in all cells are stored as the ground truth. Then, the network is simulated again with the same demands and turn fraction, but with random changes applied to the capacities of the links as shown in Figure 6.1b; this represents the presence of process noise. The speeds at four different cells, indicated by the arrows in Figure 6.1a, are then used as measurements to correct the state in the altered network. Zero mean Gaussian noise is added to these measurements, representing measurement noise. The states in the noisy network are then corrected using the EKF every five time steps. The resulting speeds in all cells are compared to the cell speeds in the original network.

For all simulations $1/\alpha[k]$ was set to $4veh^2/km^2 \forall k$, while the initial value $1/\beta[0]$ was varied from 0.01 to $20km^2/u^2$, both with the Bayesian adaptation scheme as well as without. Figure 6.2 shows the resulting Mean Absolute Percentage Error (MAPE) that was calculated for all cell speeds for all time steps. It can be seen from Figure 6.2 that for constant $1/\beta[k]$, the error shows a clear minimum. Left of the minimum the noise of the

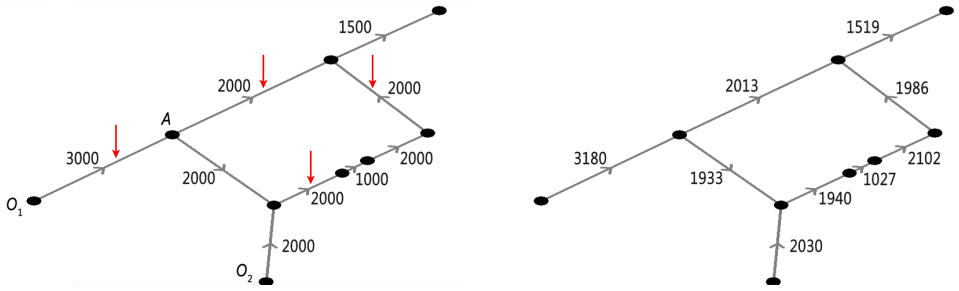


Figure 6.1: The ground truth network (a) and the network with 'process noise' (b). Numbers indicate the link capacities in veh/hr and arrows indicate measurement locations

measurements is hardly filtered, while right of it the measurements are hardly used at all. In the case of the Bayesian choice for $1/\beta[k]$ very little variation can be seen for different initial values $1/\beta[0]$. Moreover, in this case the error for the Bayesian parameters is nearly equal to the minimal possible error for constant parameters.

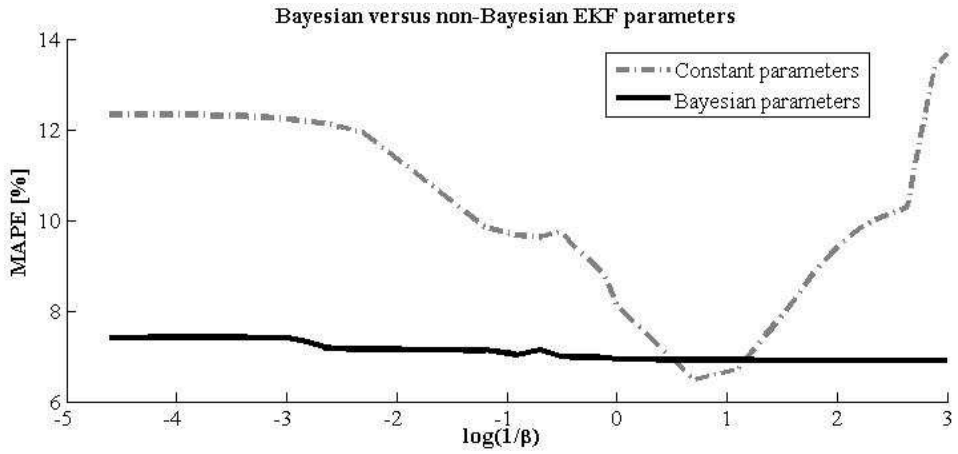


Figure 6.2: The errors for both constant $\beta[k]$ and continuously adapted Bayesian $\beta[k]$

6.4 Discussion and conclusions

This contribution has proposed a methodology for setting a value for the noise parameters (measurement and process covariances) in the Extended Kalman Filter using a two-stage Bayesian inference framework. First, the posterior distribution of the state is found, of which the maximum is found using a Maximum A Posteriori (MAP) approach. Second, the posterior distributions of the process covariance matrix $\boldsymbol{Q}[k]$ and of the measurement covariance $\boldsymbol{R}[k]$ are found from the posterior of the state. The maximum of the posterior for $\boldsymbol{R}[k]$ is found using MAP as well. As a heuristic $\boldsymbol{Q}[k]$ is held constant, leading to

optimal choices for $\boldsymbol{R}[k]$ for a given fixed value of $\boldsymbol{Q}[k]$. At the next time step $k+1$, the calculations are made with the new most probable estimate for the state and for $\boldsymbol{R}[k]$.

It is shown that the Bayesian choice for the measurement covariance leads to robustness with respect to a high or low initial choice of the process and measurement noise covariance-ratio. Using the two-stage Bayesian inference process, the modeler is given the tools for more robust Kalman filtering, also in cases where no ground truth is available. Especially in those cases, it is expected to be hard to choose appropriate values for the covariances as trial and error is not a feasible option then.

It is found that the Bayesian framework is sensitive to biased measurements, as the measurement noise covariance is in those cases overestimated. An overestimation of the measurement noise covariance leads to too small corrections of the state. Such bias will especially occur in congested conditions, where the noise distribution on the data often is not zero-mean Gaussian as negative flows or speeds do not occur. This issue will need to be further investigated.

Future work will need to resolve several other issues. First of all, adapting the process covariance as well may lead to better results. That will at least have the benefit that the results will become less sensitive to the initial value of $\boldsymbol{Q}[k]$. For this, approximations will be needed for the derivative of the log evidence to $\alpha[k]$, as an analytical solution for $\alpha[k]$ that sets this derivative to zero cannot be found. Which approximation is best suited will be the topic of future studies.

Furthermore, it was assumed that each element of the state or measurement vector is drawn from a single distribution with a single variance $1/\alpha[k]$ or $1/\beta[k]$. However, if for example measurements are obtained from different types of equipment, or from equipment that measures different quantities (for example occupancies from one detector and speeds from another), than this assumption is not valid. In that case, the Bayesian framework will need to be adapted to be able to incorporate different values on the diagonal of $\boldsymbol{R}[k]$. The foundations for this have already been laid in Bishop (1995).

In the next chapter, Chapter 7, the Extended Kalman Filter is used again in combination with the LWR model. In that chapter, one major problem that occurs when applying an EKF in real-time is solved: the EKF generally becomes too slow when applied to large networks with many measurements.

Chapter 7

The Localized Extended Kalman Filter for fast traffic state estimation

This chapter is an edited version of van Hinsbergen, C. P. I., Schreiter, T., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2010d). The localized extended kalman filter for scalable, real-time traffic state estimation. Submitted for publication in IEEE Transactions on Intelligent Transportation Systems.

Current or historic traffic states are essential input to Advanced Traveler Information, Dynamic Traffic Management and Model Predictive Control systems. As traffic states are usually not measured perfectly and everywhere, they need to be estimated from local and noisy sensor data. One of the most widely applied estimation method is the LWR model with an Extended Kalman Filter (EKF). A large disadvantage of the EKF is that it is too slow to perform in real-time on large networks. To overcome this problem the novel Localized EKF (L-EKF) is proposed in this chapter. The logic of the traffic network is used to correct only the state in the vicinity of a detector. The L-EKF does not use all information available to correct the state of the network; the resulting accuracy is however equal in case the radius of the local filters is taken sufficiently large. In two experiments it is shown that the L-EKF is much faster than the traditional Global EKF (G-EKF), that it scales much better with the network size and that it leads to estimates with the same accuracy as the G-EKF, even if the spacing between detectors is up to 5 kilometers. Opposed to the G-EKF, the L-EKF is hence a highly scalable solution to the state estimation problem.

7.1 Introduction

Advanced Traveler Information Systems (ATIS) and Dynamic Traffic Management (DTM) usually require some estimate of the current traffic state as an input. The estimated state can also be used in a Model Predictive Control (MPC) approach (Hegyi et al., 2005) to optimize traffic conditions. In general, ATIS/DTM/MPC applications need the traffic states in real-time.

Usually, the traffic state cannot be directly measured (everywhere), but needs to be estimated (interpolated) from incomplete, noisy and local traffic data. Commonly, volumes or average vehicle speeds are measured at certain locations in the traffic network, for example by double induction loop detectors or by floating car data. To estimate the total traffic state from these point measurements interpolation between the sensors is necessary.

In current state of practice often very simple methods are used to perform such a task, such as the Piece-wise Constant Speed-Based (PCSB) method and the Piece-wise Linear Speed Based (PLSB) method (van Lint and van der Zijpp, 2003). These simple methods assume that the behavior of traffic is always equal in all traffic conditions. In reality, the direction in which information travels through the network depends on traffic conditions: in free flow conditions information travels downstream, but in congested conditions information travels upstream. Therefore, these simple methods exhibit considerable bias (van Hinsbergen et al., 2008f). One reason for their continuous use in practice is that the alternatives are up to now too slow to perform in real-time.

One way to take the information direction into account is using a spatio-temporal interpolation method. The Adaptive Smoothing Method (Treiber and Helbing, 2002) is such a method that is able to interpolate traffic conditions correctly between detectors taking the information direction into account, but it cannot be used for prediction which makes it less appropriate for ATIS/DTM/MPC. A second approach that does allow for prediction is to use a traffic flow model, such as the LWR model (Lighthill and Whitham, 1955; Richards, 1956) or second order or higher order traffic flow models (Payne, 1971; Hoogendoorn and Bovy, 2001). The traffic flow models with increasing order are of increasing complexity, which comes at the cost of more parameters which makes calibration more difficult, and at the cost of larger computation times. The choice for a model thus should be based on the balance between model complexity and model abilities. In this chapter it is chosen to use the LWR model, but the presented theories are easily portable to higher order models.

What remains when a model is chosen is a method to combine local traffic data with the chosen model. One popular method that does so is the Extended Kalman Filter (EKF). This not only provides a way to use traffic data to correct the model state, but also allows for filtering of measurement noise. The latter is especially important when dealing with induction loop data because these detectors are infamous for their noisy performance. One disadvantage of the EKF is that it contains expensive matrix operations, which cause

the computation times to become very high in large scale applications. Therefore, until now it has been very hard to apply the LWR model with the EKF in real-time on large networks.

Another disadvantage of the EKF is that it is at least theoretically sensitive to the non-linearity of traffic. For the EKF a Taylor expansion is used, which is inaccurate around capacity: the derivative of the fundamental diagram that is used in the EKF shows a sudden sign change around this point, which potentially causes higher order errors (due to so-called flip-flop-behaviour). Alternatives exist, such as the Unscented Kalman Filter (UKF), which can overcome this problem by not using a Taylor expansion but by computing the covariance numerically. However, Hegyi et al. (2006) finds no considerable difference in accuracy in a freeway traffic state estimation example, while the computation times of UKF is reported to be considerably higher in one study (St-Pierre and Gingras, 2004). Because the goal of this chapter is to enable real-time filtering for online applications, the EKF is applied here. Over the last decades the EKF has been applied to traffic modeling with satisfying results (Sun et al., 2004; Wang and Papageorgiou, 2005; Tampère and Immers, 2007; Wang et al., 2007; van Hinsbergen et al., 2008f). The same ideas are believed to be portable to the UKF, just as they are portable to other models or other numerical solutions of the LWR model.

To create an EKF which is still fast enough for large scale real-time applications, in this chapter a new EKF implementation is proposed called the Localized EKF (L-EKF). In the methodology section it is shown that the L-EKF is able to rapidly combine traffic data with model information. In an experiment it is then shown that this method is not only much faster than the traditional Global EKF, but that the accuracy of the estimates is equal given an optimal radius of the L-EKF, and that the L-EKF scales much better in network size. Finally, a discussion and a conclusion are presented.

7.2 Methodology

In this section first a brief description of the first order model with the Godunov scheme is presented, along with a description of the (traditional) Extended Kalman Filter. Then, the newly proposed L-EKF is described.

7.2.1 The LWR model solved by the Godunov scheme

The basis of any macroscopic model is given by two relations: first, a partial-derivative-equation (PDE) called the conservation equation that states that no traffic can be created without external influences (Lighthill and Whitham, 1955; Richards, 1956):

$$\frac{\partial r}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{7.1}$$

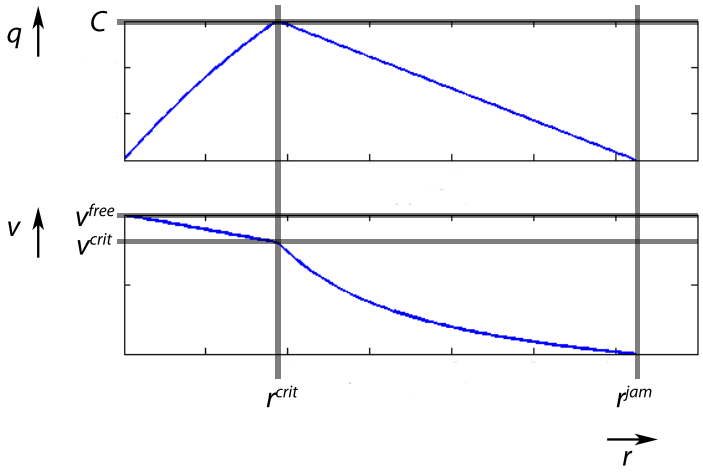


Figure 7.1: Example of the Smulders fundamental diagram

where r is the density, t the time, q the flow and x the road space, and second a relationship between density and space-mean-speed, which is given by

$$v = \frac{q}{r} \tag{7.2}$$

Given this system of two independent equations with three unknown variables a third relationship is needed. This third relation is the source of the differences in macroscopic traffic flow models. So-called second order models specify a second PDE that determines the dynamics of speed such as the Payne model (Payne, 1971), or improved versions thereof remedying problems related to isotropy and unrealistic speeds such as the models proposed by Rascle (2002) and Zhang (2002). In even higher order models a third PDE is added that governs the variation of speeds (Helbing, 1996).

The LWR model, as almost simultaneously proposed by Lighthill and Whitham (1955) and Richards (1956) is called a first-order model because it only contains one PDE: that of the conservation equation (7.1). The simplicity of the LWR model lies in the fact that it introduces a third relation in the form of an equilibrium relation $q(r)$ that specifies for each density an average flow. This relationship is usually known as the Fundamental Diagram. In this chapter the Smulders fundamental diagram is used (Smulders, 1990).

Figure 7.1 shows the shape of this fundamental diagram, which contains four parameters that are specific to a link j: the free flow speed v_j^{free}, the critical speed v_j^{crit}, the critical density r_j^{crit} and the jam density r_j^{jam}. These parameters also define the capacity of the link, $C_j = v_j^{crit} r_j^{crit}$, and the jam wave speed λ_j which equals the slope of the congested branch of the $q(r)$-plot. The flow q_j of a link j as a function of the density r is

given by:

$$q_j(r) = \begin{cases} r\left(v_j^{free} - r_{ij}[k]\dfrac{v_j^{free} - v_j^{crit}}{r_j^{crit}}\right) & \text{if } r \leq r_j^{crit} \\ C_j + \lambda_j(r - r_j^{crit}) & \text{otherwise} \end{cases} \tag{7.3}$$

Additional to modeling traffic on a link, a model needs to be chosen to propagate traffic over a node. In this chapter Daganzo's merge and diverge node model is used. Details of this node model are omitted here but can be found in (Daganzo, 1995a).

Given this fundamental diagram there are three independent equations with three unknown variables and the model can be solved. In order to apply the EKF a numerical solution is needed that allows formulating the LWR model in terms of a state-space equation. Several stable numerical solutions exist to the LWR model, such as the Godunov scheme (Lebacque, 1996) or methods based on the Lagrangian formulation (Leclerq et al., 2007). Because it is the most widely applied solution, the Godunov scheme is used as a numerical solution in this chapter. The methods developed in this chapter are easily applied to the alternatives.

The numerical solution is found using a finite volume method where each link j in the network is discretized into cells with homogeneous conditions of length Δl_j and time is discretized into intervals with length Δt during which the conditions are considered stationary. The length Δl_j of cells on a link j are chosen based on the Courant-Friedrichs-Lewy condition so that the numerical solution is stable (Courant et al., 1928):

$$\Delta l_j = v_j^{free} \Delta t \tag{7.4}$$

where v_j^{free} is the free flow speed of link j. Because the free speed on different links may vary (due to for example different speed limits), the cell length is allowed to vary between links, but all cells on one link are of equal length. Given this discretization, the conservation equation can be rewritten in state-space form:

$$r_{ij}[k+1] = r_{ij}[k] + \frac{\Delta t}{\Delta l_j}\left(F_{ij}^{in}[k] - F_{ij}^{out}[k]\right) \tag{7.5}$$

where $F_{ij}^{in}[k]$ [veh/h] denotes the flux into cell i of link j at time k and $F_{ij}^{out}[k]$ that out of the cell. For adjacent cells the flux-out of the upstream cell is equal to the flux-in of the downstream cell. To calculate these fluxes the well understood, simple and stable Godunov scheme is used (Lebacque, 1996). Note that there exist other solution methods of the first order model, for example those based on Lagrangian coordinates (Leclerq et al., 2007), and that here an explicit time stepping scheme is used, but that also be an implicit scheme could be applied (van Wageningen-Kessels et al., 2009). Application of the theories presented in this chapter to other numerical solutions is thought to be straightforward, but application to an implicit time stepping scheme is not.

The fluxes $F_{ij}[k]$ (in veh/hr) between cell borders are determined by comparing the available supply $S_{ij}[k]$ (the maximum flow that can still enter a certain cell i) and the prevailing demand $D_{i-1j}[k]$ (the maximum flow that wants the exit the upstream cell $i-1$) (Lebacque, 1996). These demand and supply functions read:

$$D_{ij}[k] = \begin{cases} r_{ij}[k]\left(v_j^{free} - r_{ij}[k]\dfrac{v_j^{free} - v_j^{crit}}{r_j^{crit}}\right) & \text{if } r_{ij}[k] \leq r_j^{crit} \\ C_j & \text{otherwise} \end{cases} \tag{7.6}$$

$$S_{ij}[k] = \begin{cases} C_j & \text{if } r_{ij}[k] \leq r_j^{crit} \\ C_j + \lambda_j(r_{ij}[k] - r_j^{crit}) & \text{otherwise} \end{cases} \tag{7.7}$$

7.2.2 Extended Kalman Filter

The Kalman Filter is a recursive filter that estimates the state of a linear model based on the last estimate of the state and a number of normally distributed observations (Kalman, 1960; Maybeck, 1979). When made applicable to non-linear models, an Extended Kalman Filter (EKF) can be used where a linearization of the non-linear model around its current state is used (Jazwinsky, 1970).

The traffic state in the network at time k is uniquely described by the vector $\boldsymbol{r}_k$ of all densities $r_{ij}[k]$ of all cell i on all links j. The EKF is based on a non-linear state space equation, which in this case expresses the density vector as a function of the density vector in the previous time step plus a zero-mean Gaussian noise vector $\boldsymbol{w}[k]$ which has a covariance matrix $\boldsymbol{P}[k]$:

$$\boldsymbol{r}[k] = f(\boldsymbol{r}[k-1]) + \boldsymbol{w}[k] \tag{7.8}$$

The function $f(\boldsymbol{r}[k-1])$ here represents the state space equation (7.5) for each cell. The EKF furthermore uses a measurement equation describing the measurement vector $\boldsymbol{z}[k]$ as a function of $\boldsymbol{r}[k]$ with zero-mean Gaussian measurement noise $\boldsymbol{v}[k]$ which has a covariance matrix $\boldsymbol{R}[k]$:

$$\boldsymbol{z}[k] = h(\boldsymbol{r}[k]) + \boldsymbol{v}[k] \tag{7.9}$$

The function $h(\boldsymbol{r}[k])$ expresses a function that maps the density to a variable in the same dimension as the measurements; $\boldsymbol{z}[k]$ denotes the vector of all measurements. In this chapter speeds are used as measurements; the fundamental diagram (7.3) together with (7.2) is used to map the density in a certain cell to a speed. Note that the EKF is derived from Gaussian assumptions on both the distributions of the data and the model. Generally, Gaussian distributions are not found in practice in traffic. However, the EKF can still be applied when distributions are non-Gaussian, in which case it becomes a meta-heuristic approach. The value of the EKF has been shown in the many cases where it has been applied successfully for traffic state estimation (Sun et al., 2004; Wang and Papageorgiou, 2005; Tampère and Immers, 2007; Wang et al., 2007; van Hinsbergen et al., 2008f).

The EKF algorithm consists of two steps: a prediction and a correction step. In the prediction step, the model under consideration is used to predict a new state vector along with an error variance-covariance matrix. The prediction step is defined by:

$$\boldsymbol{r}^{-}[k] = f(\boldsymbol{r}[k-1]) \tag{7.10}$$

$$\boldsymbol{P}^{-}[k] = \boldsymbol{J}[k]\boldsymbol{P}[k-1]\boldsymbol{J}[k]^{T} + \boldsymbol{Q}[k] \tag{7.11}$$

where $\boldsymbol{Q}[k]$ is the error covariance matrix of the model and the matrix $\boldsymbol{P}^{-}[k]$ equals an a priori estimate of the error variance-covariance matrix of the state vector that describes the noise vector $\boldsymbol{w}[k]$. Finally, the matrix $\boldsymbol{J}[k]$ is used for the linearization of the model; it equals the derivative of the model to the state:

$$\boldsymbol{J}[k] = \nabla_{\boldsymbol{r}[k]} f(\boldsymbol{r}[k])|_{\boldsymbol{r}^{+}[k-1]} \tag{7.12}$$

where $\boldsymbol{r}^{+}[k-1]$ is the a posteriori state vector of the previous time step which will be introduced later. Note that there only exist non-zero derivatives between adjacent cells, either on a link or when a node connects two cells.

In the second step, the correction step, measurements are used to make corrections to the state. For the EKF, the measurements also need to be linearized around the current state. For this, define $\boldsymbol{H}[k]$ to be the derivative of the measurement mapping function to the state:

$$\boldsymbol{H}[k] = \nabla_{\boldsymbol{r}[k]} h(\boldsymbol{r}[k])|_{\boldsymbol{r}^{-}[k]} \tag{7.13}$$

The second step of the EKF is now given by

$$\boldsymbol{K}[k] = \frac{\boldsymbol{P}^{-}[k]\boldsymbol{H}[k]^{T}}{\boldsymbol{H}[k]\boldsymbol{P}^{-}[k]\boldsymbol{H}[k]^{T} + \boldsymbol{R}[k]} \tag{7.14}$$

$$\boldsymbol{r}^{+}[k] = \boldsymbol{r}^{-}[k] + \boldsymbol{K}[k]\left(\boldsymbol{z}[k] - h(\boldsymbol{r}^{-}[k])\right) \tag{7.15}$$

$$\boldsymbol{P}[k] = \left(\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{H}[k]\right)\boldsymbol{P}^{-}[k] \tag{7.16}$$

where $\boldsymbol{I}$ is an identity matrix and $\boldsymbol{K}[k]$ is called the Kalman gain which indicates how much the state should be corrected based on the relative values of the uncertainties of the a priori state estimate (through $\boldsymbol{P}^{-}[k]$) and of the measurements (through $\boldsymbol{R}[k]$). The result of the EKF is an a posteriori state vector $\boldsymbol{r}^{+}[k]$, which is a balanced estimate of the traffic state given both the estimate of the model and the measurements, and an a posteriori estimate of the error covariance matrix $\boldsymbol{P}[k]$. A more detailed description of the EKF and its application to traffic are found in (Wang and Papageorgiou, 2005; Tampère and Immers, 2007).

The EKF contains two parameters: the values of the measurement covariance matrix $\boldsymbol{R}[k]$ as well as the process covariance matrix $\boldsymbol{Q}[k]$. These parameters may be state dependent, but the way to determine these is a discussion too long for this chapter. In Chapter

6 a method is proposed to adapt the EKF parameters dynamically. Here it is chosen to keep $\boldsymbol{Q}$ and $\boldsymbol{R}$ constant to keep the discussion focused. Also, as is common, both $\boldsymbol{Q}$ and $\boldsymbol{R}$ are taken to be diagonal matrices, assuming independence. Note that $\boldsymbol{P}[k]$ will not be a diagonal matrix but will have non-zero elements off the diagonal as well, indicating covariance between the errors in different cells.

7.2.3 Global Extended Kalman Filter

Usually, the EKF is applied at once to the entire network, so that the state vector $\boldsymbol{r}[k]$ represents all cells in the entire network and $\boldsymbol{P}[k]$ contains estimates of the covariance of the errors between all cells (Gosh and Knapp, 1978; Wang and Papageorgiou, 2005; Zuurbier et al., 2006). Each time when measurements become available somewhere in the network the densities in all cells are corrected at once. This process, which is termed Global EKF (G-EKF) here, uses the available data to its maximum potential, as all densities in all cells are corrected using the error covariance between all measured cells and all non-measured cells. However, this procedure has one major concern: the calculation times can become very high.

The EKF contains two expensive operations: the inverse operation in equation (7.14) that scales in the number of measurements and the matrix multiplications of (7.16) that scales in the number of cells in the network. Theoretically, both of these operations scale at best in the order of $O(M^{2.8074})$ with the Strassen algorithm (Strassen, 1969). For larger networks (containing more than say a few hundred measured cells) the complexity of these operations will make real-time calculations impossible on a normal computer, rendering the G-EKF infeasible for large-scale online applications.

7.2.4 Localized Extended Kalman Filter

In this section, a new EKF implementation is proposed that is much faster on larger networks because it simplifies the inverse operation. First, it is important to notice that the error covariance matrix $\boldsymbol{P}[k]$ generally contains many values that are close to zero.

Progression of covariance over the network

Over time a non-zero error covariance can exist between the errors of any two cell states. In this subsection it will be shown that the covariance under most conditions decrease as the distance between two cells increases.

Through (7.11) and (7.16) it can be seen that the covariance is influenced by the linearization of the fundamental diagram $\boldsymbol{H}[k]$, the linearization of the model $\boldsymbol{J}[k]$ and the Kalman Gain $\boldsymbol{K}[k]$. Because of the non-linearity of the system and the stochasticity of the model and the data, it is very hard to analytically prove under which conditions the covariance will decrease with increasing distance. However, through extensive experimentation

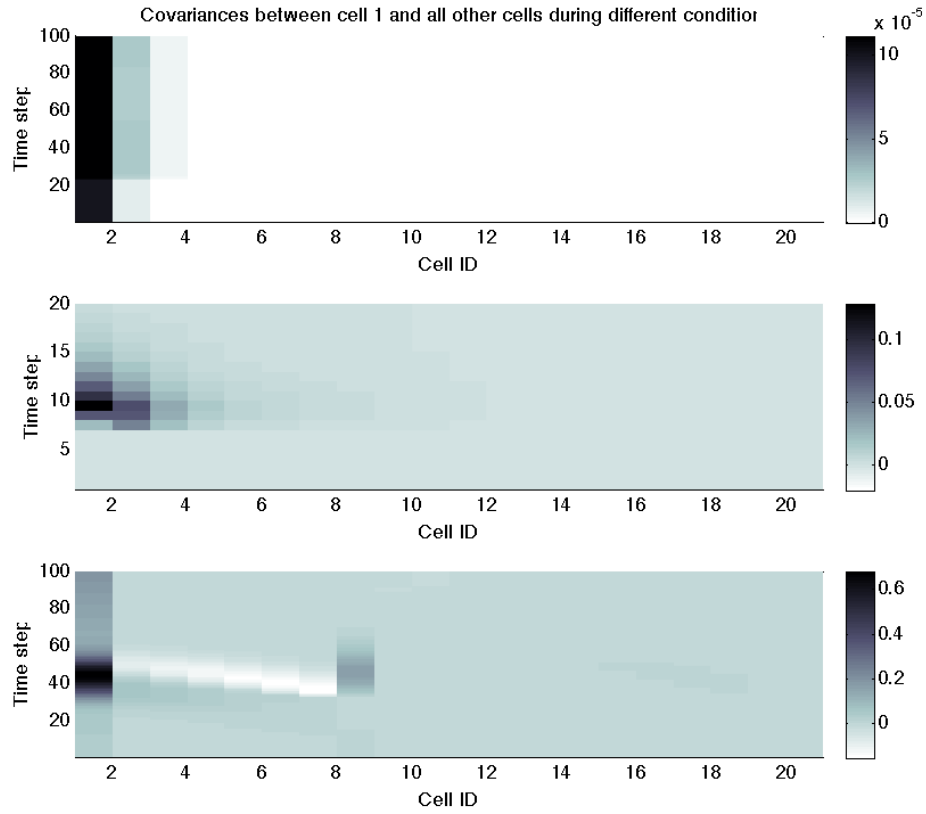


Figure 7.2: The error covariance values of the first cell of the 21-cell link and all other cells on the link in three different conditions: (a) free flow, (b) congestion and (c) a state change from free flow to congestion. A dark color indicates a high covariance. The gradient indicates that the covariance decreases the further the two cells are apart. Note that different scales apply to the different figures

on different networks with different sizes it has been observed that under most conditions the error covariance between two cells further away are smaller than between two cells close to each other. This is a very intuitive result: only a very small portion of traffic on a certain location will travel to another location for example 100 km away; therefore, it can be expected that the error covariance between these two locations is nearly zero.

Figure 7.2 shows the error covariance between the cells on a certain route. For this result the small network as will be presented later (see Figure 7.4) was simulated for 600 time steps. The demand and supply of the origins and destination were varied in order to cause state transitions to occur. In most cases the covariance between two cells are smaller the further they are apart; only in the third case the covariance between cell 1 and cell 9 is larger than before during a few time steps. However, further downstream the

error covariance is again very close to zero. Similar results were obtained on different networks with other structures and other congestion patterns.

The fact that the covariance values are usually smaller further away from a certain cell means that in the G-EKF the matrix $\boldsymbol{P}[k]$ will contain many values close to zero for cells far apart in the network. Corrections to states based on these very small covariance values will be negligible.

It is important to note that the non-zero values will not always be close to the diagonal of the matrix, because cells that are spatially close in a network cannot be guaranteed to be close to each other in the matrix. Also, experimentation has shown that $\boldsymbol{P}[k]$ is not always diagonally dominant. These two issues prevent more efficient algorithms to be applied for the inverse operation, and an alternative is required.

In this chapter it is therefore proposed to use the logic of the network topology in the corrections and to use a measurement of a detector to correct only the states of cells in the vicinity of that detector. The resulting scheme is named Localized Extended Kalman Filter (L-EKF) to indicate the local nature of the corrections.

The L-EKF algorithm

In the L-EKF, many local EKFs are called sequentially for each cell that contains measurements, instead of constructing one large EKF for the entire network. Local measurements are no longer used to correct the errors of cells far downstream or upstream, but are only used to correct the state of cells within a certain radius ζ of the measurement. Figure 7.3 shows the principle of the L-EKF with $\zeta = 2$. Note that ζ can be taken constant throughout the simulations or dynamic based on the prevailing traffic conditions. In order to remain focused, in this chapter it is chosen to keep ζ constant; future work needs to validate if a dynamic ζ can improve the results.

In the local EKF scheme, first a global estimate of the state vector $\boldsymbol{r}^{-}[k]$ and of the error covariance matrix $\boldsymbol{P}^{-}[k]$ is made using fully-sized $\boldsymbol{J}[k]$, $\boldsymbol{P}[k]$ and $\boldsymbol{Q}$ matrices. These global vectors and matrices are indicated by a superscript G and can be calculated quickly because the required matrix operations are relatively light:

$$\boldsymbol{r}^{G}[k] = f(\boldsymbol{r}^{G}[k-1]) \tag{7.17}$$

$$\boldsymbol{P}^{G}[k] = \boldsymbol{J}^{G}[k]\boldsymbol{P}^{G}[k-1](\boldsymbol{J}^{G}[k])^{T} + \boldsymbol{Q}^{G} \tag{7.18}$$

Then, a local EKF is constructed for the first measured cell. A new, local a priori density vector $\boldsymbol{r}^{L-}[k]$ is created by copying all elements within the filter radius ζ from $\boldsymbol{r}^{G}[k]$ and a local a priori error covariance matrix $\boldsymbol{P}^{L-}[k]$ is obtained by copying the relevant values from $\boldsymbol{P}^{G}[k]$. Finally, a new derivative matrix $H^{L}[k]$ is created substituting $\boldsymbol{r}^{L-}[k]$ in (7.13). Now, new estimates of the densities and of the (co)variances in the vicinity of

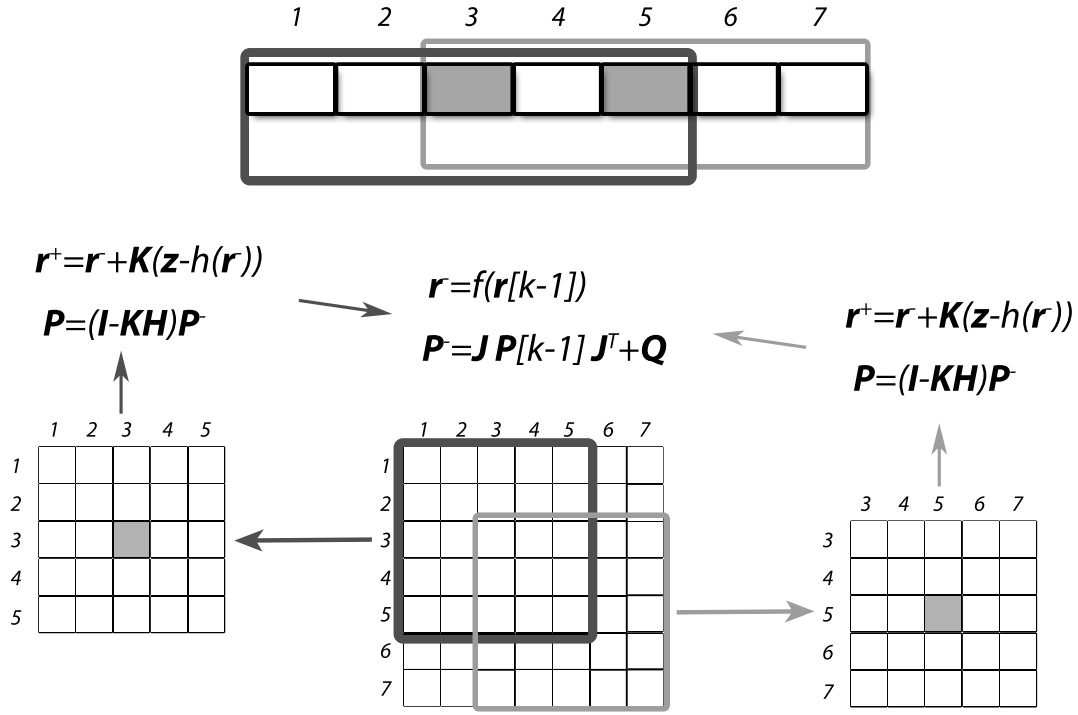


Figure 7.3: The principle of the Localized EKF on a 7-cell link with measurements in cell 3 and 5. On top a 7-cell link is shown. First the global a priori state vector $\boldsymbol{r}^G[k]$ and a priori error covariance matrix $\boldsymbol{P}^G[k]$ are computed using (7.17) *and* (7.18)*. The 7x7 matrix represents $\boldsymbol{P}^G[k]$. Then an L-EKF is constructed for cell 3 which extracts a $\boldsymbol{P}^{L-}[k]$ matrix (the dark gray 5x5 square). This L-EKF corrects the states of cells 1-5; the resulting estimates of $\boldsymbol{r}$ and $\boldsymbol{P}$ are copied back into the global matrices; then an EKF is constructed for cell 5 (light gray square) and the process is repeated for cells 3-7*

the measurement are determined using

$$\boldsymbol{K}^L[k] = \frac{\boldsymbol{P}^{L-}[k](\boldsymbol{H}^L[k])^T}{\boldsymbol{H}^L[k]\boldsymbol{P}^{L-}[k](\boldsymbol{H}^L[k])^T + \boldsymbol{R}^L} \tag{7.19}$$

$$\boldsymbol{r}^{L+}[k] = \boldsymbol{r}^{L-}[k] + \boldsymbol{K}^L[k]\left(\boldsymbol{z}^L[k] - h(\boldsymbol{r}^{L-}[k])\right) \tag{7.20}$$

$$\boldsymbol{P}^L[k] = \left(\boldsymbol{I}^L - \boldsymbol{K}^L[k]\boldsymbol{H}^L[k]\right)\boldsymbol{P}^{L-}[k] \tag{7.21}$$

The procedure now continues by substituting the state estimates $\boldsymbol{r}^{L+}[k]$ and error covariance estimates $\boldsymbol{P}^{L+}[k]$ back into the global vector $\boldsymbol{r}^G[k]$ and the global matrix $\boldsymbol{P}^G[k]$ at the correct coordinates. Then, the above process is repeated for the next measurement using the new values of the state and of the covariance wherever there is overlap (the center 3 cells of the link and the center 9 cells in the $\boldsymbol{P}$-matrix in Figure 7.3).

Note that the order in which the local filters are called is not of importance in case the model is linear. The Kalman Filter (so not the Extended Kalman Filter that is an approximation) is a Bayesian optimal estimator that finds the maximum of the posterior of the state of a cell i on a link j at time k given the data vector $\boldsymbol{z}[k]$:

$$p(r_{ij}[k]|\boldsymbol{z}[k]) = \frac{p(\boldsymbol{z}[k]|r_{ij}[k])p(r_{ij}[k])}{p(\boldsymbol{z}[k])} \tag{7.22}$$

Consider the case where two sequential corrections are made, one with the data point z_1 and one with the data point z_2, and where the posterior of the first correction is the prior of the second correction. Note that here the indices i, j and k will be omitted to simplify notations. In the case where z_1 is first used to correct, then the first correction step of the Kalman Filter can be written as:

$$p(r|z_1, z_2) = \frac{p(z_1|r, z_2)p(r|z_2)}{p(z_1|z_2)} \tag{7.23}$$

The second correction $p(r|z_2)$ can also be found using the KF, and using Bayes rule can be written as

$$p(r|z_2) = \frac{p(z_2|r)p(r)}{p(z_2)} \tag{7.24}$$

Substituting (7.24) into (7.23) and using the fact that $p(a|b)p(b) = p(a, b)$ the following result is obtained:

$$\begin{aligned} p(r|z_1, z_2) &= \frac{(p(z_1|z_2, r)p(z_2|r)p(r)}{p(z_1|z_2)p(z_2)} \\ &= \frac{p(z_1, z_2|r)p(r)}{p(z_1, z_2)} \end{aligned} \tag{7.25}$$

It can now be seen that the same result would be obtained if z_2 was first used, and then

z_1. Because the LWR model is non-linear, it can be expected that the order does influence the solution; however, no a priori knowledge is present on what order to follow. In case the state of the model is close to the actual state, i.e. if the model is well calibrated and previous corrections have lead to the state being approximately correct, then the linearization is more accurate; in that case, the order in which the corrections are applied will less influence the solution. In this chapter it is chosen to apply the filters in the order of which data arrives in the estimation processing computer.

The L-EKF process has two major advantages compared to the G-EKF. First of all, the measurement error covariance matrix R in (7.19) is of size 1×1. This means that the inverse operation becomes scalar and is thus very fast. Second of all, the matrix multiplications (7.21) are performed on much smaller matrices which again results in a gain in computation time. The L-EKF procedure scales linearly in the number of measurements; for each available measurement, equations (7.19)- (7.21) need to be carried out one more time, but each of these operations is very light. Opposed to the G-EKF, the L-EKF is therefore suitable for large-scale and real-time applications.

Opposed to the G-EKF, in the L-EKF the states of cells far away are not corrected. This leads to a potential loss of accuracy because not all covariance values are used for correction. However, as the error covariance between cells further apart is generally very small, the loss in accuracy is expected to be negligible in case the L-EKFs have a sufficiently large radius and in case of a sufficiently dense measurement network.

It is important to note that the radius of the L-EKF is taken symmetric. The number of cells upstream that are corrected is equal to the number of cells downstream. Because the error covariance matrix is symmetric, the covariance between cell A and B is equal to the covariance between B and A. A measurement in A can thus equally well be used to correct the state in cell B as a measurement in B can be used to correct A. Because a fixed radius is used throughout the simulation and a priori nothing can be said about the values of the covariance between the center cell and the upstream cells relative to those between the center cell and the downstream cells, the radius is taken symmetric.

To show the difference between the L-EKF and G-EKF both in accuracy and in computation time two separate experiments are conducted: one on a small scale with synthetic data, and one on a large scale with real-world data.

7.3 Experiment 1: synthetic data

To illustrate the accuracy of the L-EKF compared to the G-EKF, first an experiment on a small-scale network is conducted. The Localized and Global EKF have been programmed in the software package JDSMART which is a Java-based implementation of the LWR model solved by the Godunov scheme. For the Matrix operations, the fast UJMP Java-library has been used (UJMP, 2010). All computations are performed on a Windows XP machine with a 3.0 GHz dual core processor and with 2GB of memory.

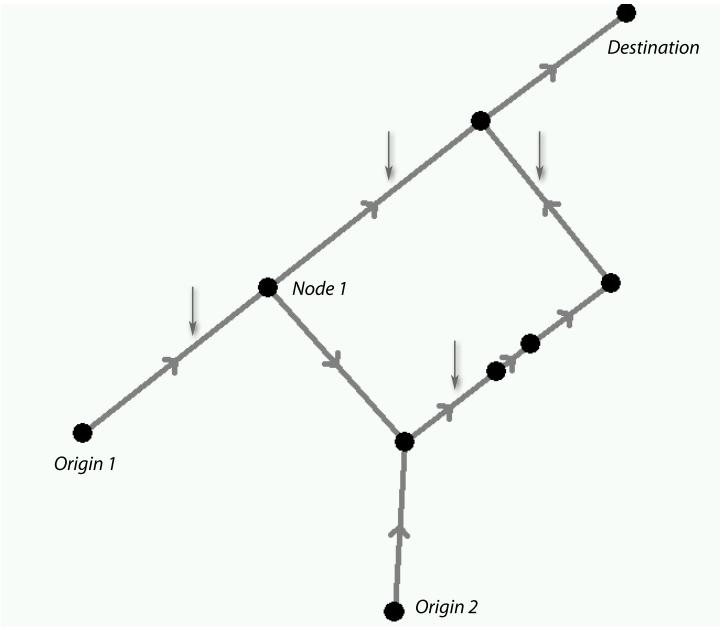


Figure 7.4: The experimental network on which the Localized EKF was verified. The vertical arrows indicate the four measurement locations

Figure 7.4 shows the network of this experiment. Arrows indicate the driving direction. The network is discretized into cells using (7.4) with a time step of 2 seconds, resulting in 59 cells. First, a ground-truth simulation is performed, with a certain demand pattern on Origin 1 and Origin 2 and a different set of fundamental diagram parameters for each link as shown in Table 7.1, which together caused a complex congestion pattern on the network. Each time step the densities of all cells were stored as the ground truth. The speeds in four cells throughout the network indicated by the vertical arrows in Figure 7.4 are stored each time step, which are distorted with zero-mean Gaussian noise with a standard deviation of 5 km/h.

Table 7.1: Parameters of all links in the synthetic data experiment

Link number	v^{free} [km/h]	v^{crit} [km/h]	C [veh/h]	r^{jam} [veh/km]
1	100	80	3000	125
2	100	80	2000	125
3	100	80	2000	125
4	100	80	2000	125
5	100	80	1000	125
6	100	80	1500	125
7	100	80	2000	125
8	100	80	2000	125

The network is then simulated again, with the same fundamental diagrams but with zero-mean Gaussian noise added to the demands at the two origins (standard deviation of 200 veh/hr) and the turn fractions of node 1 (standard deviation of 20%). This causes the resulting congestion pattern to be considerably different from the ground-truth experiment. Using the (noisy) speed measurements from the ground-truth simulation, the states can be corrected (the intentionally added noise removed) using either the L-EKF or the G-EKF. The process of adding noise to the speed measurements and to the demand and turn fractions is repeated 25 times in order to be able to generalize the results.

Figures 7.5 and 7.6 show an example of the ground-truth, distorted and corrected cell densities for one of the 25 simulations at the four selected locations. For the corrected densities the best performing G-EKF and L-EKF are plotted. It can be seen that the estimated densities are much closer to the ground truth densities when compared to the simulation without EKF, and that the L-EKF and G-EKF overlap for almost all time steps for all locations.

The parameters of the L-EKF and G-EKF (the matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ and the L-EKF radius ζ) were set as follows. The values on the diagonal of $\boldsymbol{R}$ are set to $25km^2/h^2$ for both the L-EKF and the G-EKF, because the measurement error has a standard deviation of 5 km/h. For each of the 25 simulations, the EKFs are tested with different values on the diagonal of $\boldsymbol{Q}$, and the best scoring values are chosen; for the L-EKF, the radius ζ is also varied (but taken equal for all filters in one simulation) between 0 and 59, the network size.

Figure 7.7(a) shows the average Root Mean Square Error (RMSE) between the corrected states and the ground-truth states for all time steps for all 25 simulations, along with the average computation times. As can be seen from the figure, both EKFs result in lower errors than when no correction is applied. It can also be seen that the L-EKF with a small radius (< 5) performs worse compared to the G-EKF, because not all data is used to its full potential; however, with sufficiently large radii (> 5) the same level of accuracy is obtained. This result confirms that corrections made by the G-EKF to cells far away are indeed negligible. Also, the results of the L-EKF with full radii (59 cells) confirm that the order in which the filters are used are in this case not important, as the sequentially called filters are as accurate as the G-EKF.

Figure 7.7(b) shows that even for this small network the L-EKF is faster than the G-EKF for $\zeta < 20$. Performing 4 individual corrections on a small radius is thus already faster than doing 1 large correction. For larger ζ the calculation times start to increase beyond the average computation times for the G-EKF, because of the overhead in copying the data back and forth and because of the other matrix operations (7.19)-(7.21).

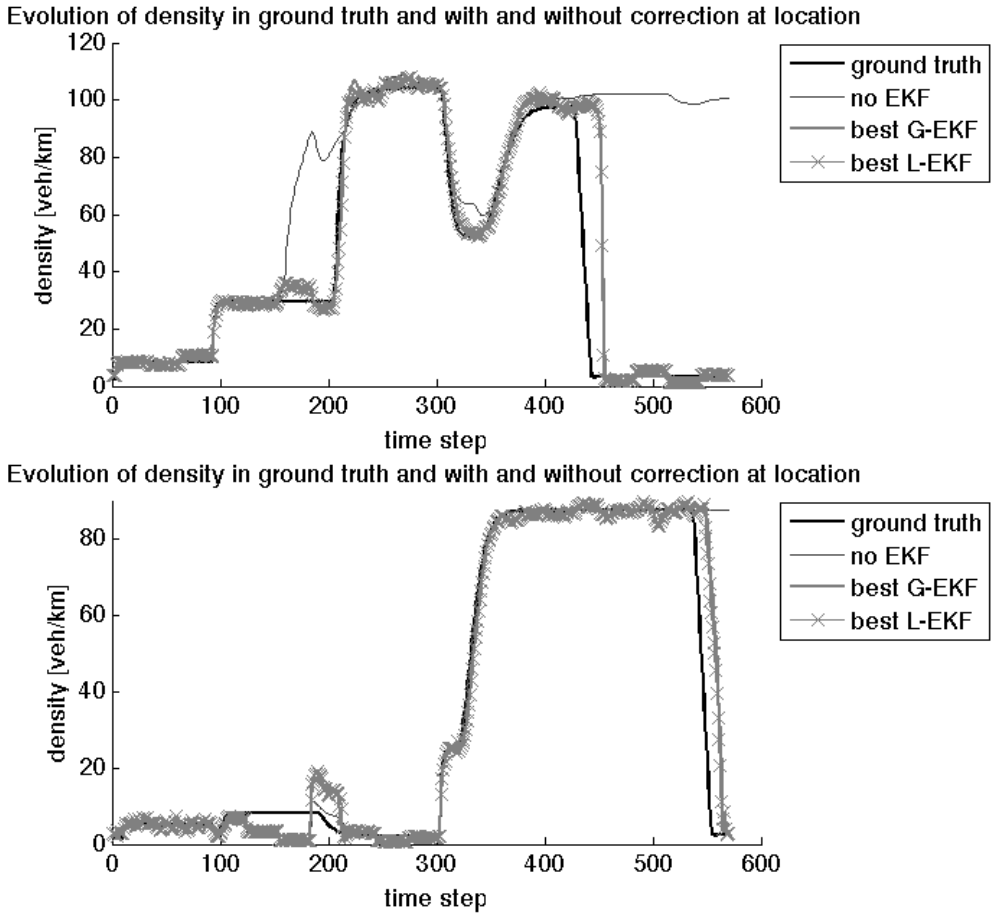


Figure 7.5: The resulting density patterns from one of the 25 simulations for the locations indicated by vertical arrows in Figure 7.4. The black solid line is the ground truth. The resulting 'wrong' pattern is shown in light gray, and the corrected densities using G-EKF/L-EKF in darker gray. For almost all time steps, the densities from G-EKF and L-EKF are almost equal

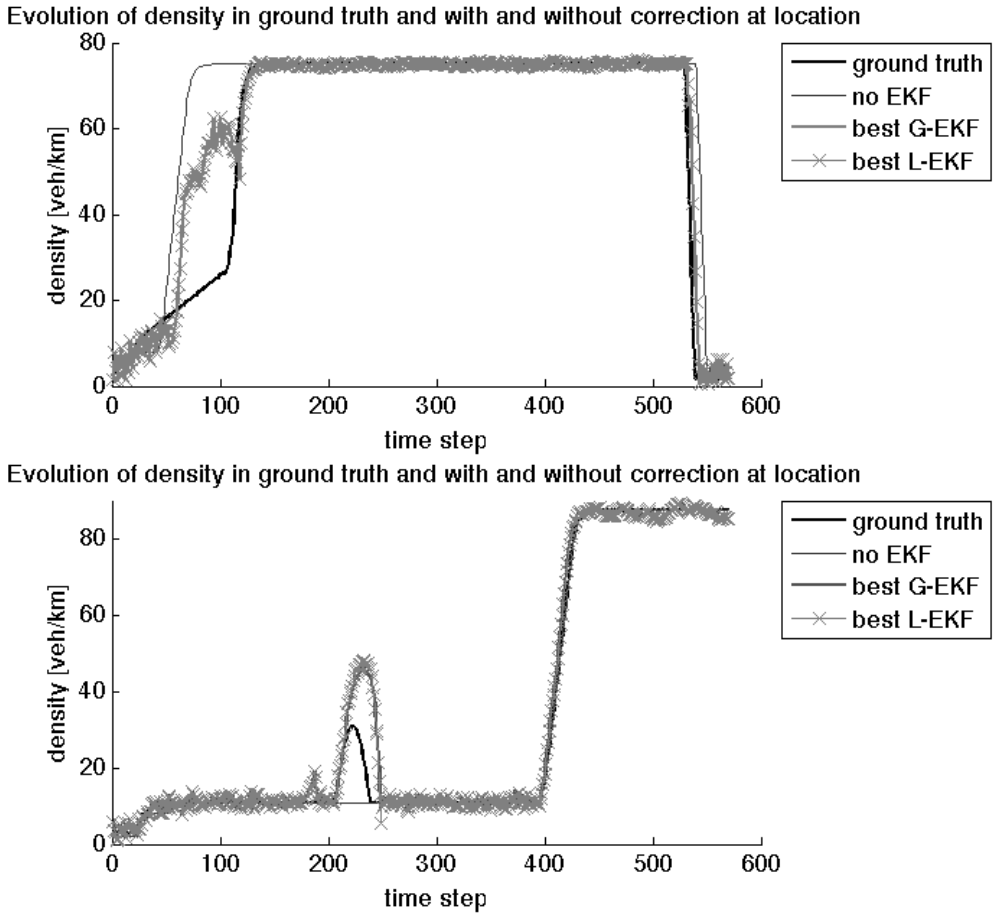
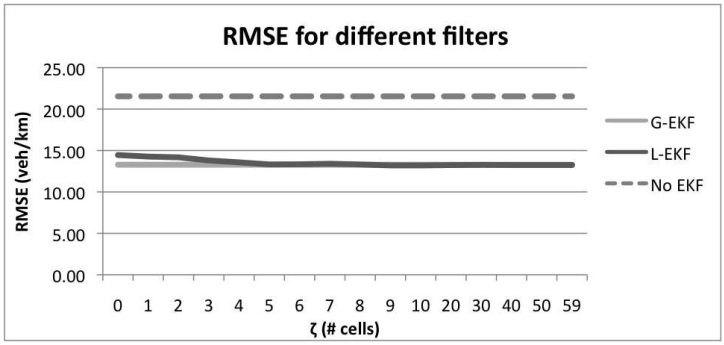
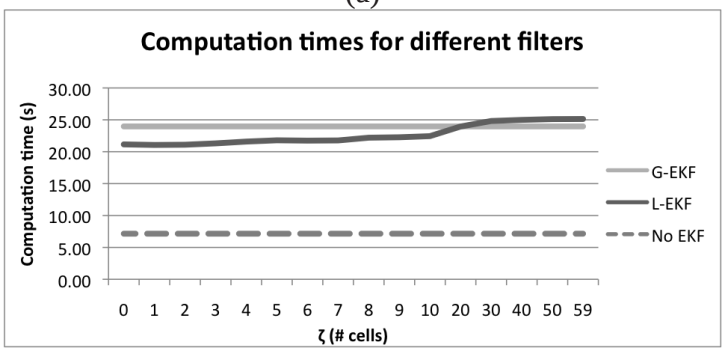


Figure 7.6: The resulting density patterns from one of the 25 simulations for the locations indicated by vertical arrows in Figure 7.4



(a)



(b)

Figure 7.7: Comparison of accuracy in terms of RMSE (a) and of computation times (b) for the different filters. The L-EKF (dark solid line), is compared for different horizons to the base simulation without EKF (dashed lines) and G-EKF (light solid line)

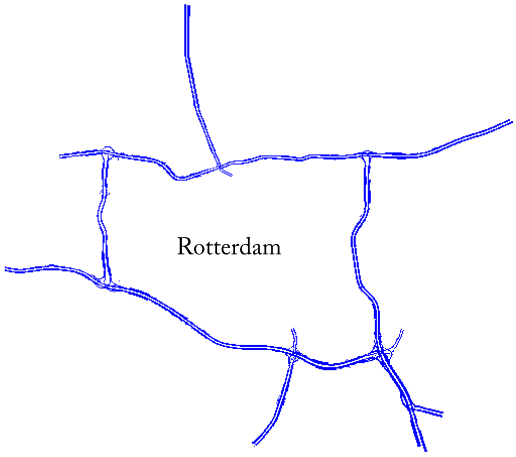


Figure 7.8: The freeway network around the city of Rotterdam has a total length of 272 km

7.4 Experiment 2: real data

To show the gain in computation time and the comparative accuracy of L-EKF versus G-EKF on a large real-world network the two EKFs are applied to the freeway network around Rotterdam, the Netherlands as shown in Figure 7.8.

This freeway network has a total length of 272 km. A time step of 5s is chosen, after which the links are discretized using (7.4) leading to a total size of the network of 1911 cells with an average cell length of approximately 142m. Throughout the network 531 double loop detectors are placed, which corresponds to an average spacing of about 500m, of which each minute speed data is available.

The fundamental diagrams are roughly calibrated using a heuristic approach that uses three years of historic data of all detectors in the network. The free speed and critical speed are found sorting the speeds of each detector in a cumulative curve as shown in Figure 7.9. The figure shows two points of sharp curvature. The curvature at the complete right of the graph is a point where only few vehicles drive faster and can thus be interpreted to be the free flow speed v^{free}. The second curve from the right indicates a point where suddenly few speed measurements are available due to traffic breakdown and can thus be interpreted to be v^{crit}. The estimation procedure of these two speeds is a heuristic based on two other empirical observations: (1) speeds are approximately evenly distributed between the critical and free flow speed so that the line between the two points is always a straight line and (2) no detector was found where more than 70% of the measurements are congested. The two speed parameters are found by taking the slope of the line at 70% of the total number of points and by finding the point right and left to it where the cumulative curve deviates more than a threshold from the slope. If multiple detectors

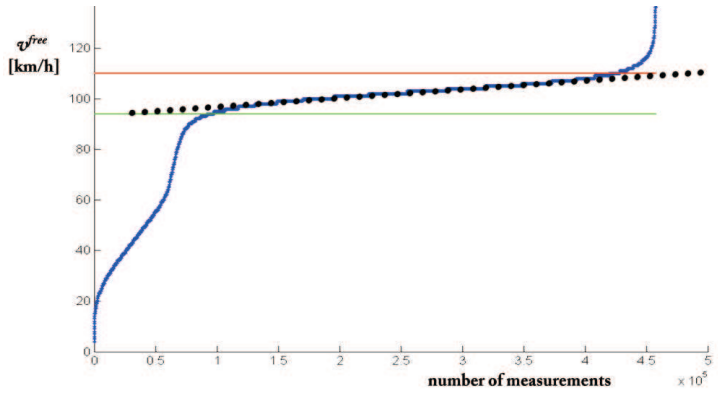


Figure 7.9: Sorted speeds of one detector. The horizontal lines indicate the free flow speed and critical speed. The dashed line indicates the linear nature of the cumulative curve in free flow conditions

exist for one link, the median of all estimates for v^{free} and v^{crit} are taken for the link; if no detectors exist of a link the parameters of the surrounding links are used. The remaining two parameters of the fundamental diagram are taken constant for all links based on experience: $r^{crit} = 25veh/km$ and $\lambda = -18km/h$.

For the experiment a regular Monday morning peak period is selected, 10 March 2008 from 6AM to 10AM. The 531 detectors are split in two: one part which is used for estimation, and another part which is used for validation. The validation detectors are used to compute the Root Mean Square Error (RMSE) between the estimated speeds and the modeled speeds. Four different scenarios are made with increasing scarcity of detectors used for estimation: 25%, 50%, 75% and 90% validation detectors, where it is always ensured that the remaining estimation detectors are evenly distributed over the network. With fewer detectors used for estimation, computation times are expected to decrease while the RMSE is expected to increase.

Each scenario the network is simulated, feeding data into the network each minute; the prediction step of the EKFs is performed each time step, while the correction step is performed only once when new data is loaded into the network (every 12 time steps). The 4-hour simulation is performed with the L-EKF with fixed radii varying between 1 and 30 and with the G-EKF. For both filter types, 8 different values of $\boldsymbol{Q}$ were tested on a large range from 0.01 (almost no correction) to 100,000 (a lot of correction) veh^2/km^2 with a fixed value of $\boldsymbol{R} = 25km^2/h^2$.

Table 7.2 shows the results for the different filters for the simulations without any EKF and the best performing L-EKF and G-EKF. For all four scenarios the number of validation detectors is given, as well as the average spacing between detectors. The last column shows the RMSE of the L-EKF. It can be seen that with a sufficiently large radius the RMSE for the L-EKF is always at least as low as the G-EKF. Both filters result in more accurate estimates of the speeds compared to no correction.

A very large difference between the computation times of the two types of filters is visible: the L-EKFs are between 12 and 51 times faster, depending on the radius and the number of detectors used for estimation. As expected, the computation time increases with more detectors used for estimation and fewer for validation. However, the computation times of the G-EKF steeply increase when more detectors are used for estimation, as shown in Figure 7.10.

Table 7.2: Best results of L-EKF versus G-EKF on a large scale network with 1911 cells for different numbers of detectors used for validation

Method	Optimal $\boldsymbol{Q}^2[veh^2/km^2]$	Computation time [s]	$\times$ Real-Time	RMSE [km/h]
	25% (133) validation detectors; spacing = 0.7 km			
No EKF	-	24	605.0	33.3
L-EKF ($\zeta = 1$)	100	730	19.7	17.6
L-EKF ($\zeta = 10$)	1000	876	16.4	17.3
L-EKF ($\zeta = 20$)	100	1067	13.5	16.9
L-EKF ($\zeta = 30$)	100	1566	9.2	16.8
G-EKF	100	37150	0.4	16.7
	50% (264) validation detectors; spacing = 1.0 km			
No EKF	-	20	720.0	33.3
L-EKF ($\zeta = 1$)	100	653	22.4	17.8
L-EKF ($\zeta = 10$)	100	723	19.9	17.6
L-EKF ($\zeta = 20$)	100	857	16.8	17.5
L-EKF ($\zeta = 30$)	10	1138	12.7	19.4
G-EKF	100	21813	0.7	17.8
	75% (398) validation detectors; spacing = 2.0 km			
No EKF	-	24	590.2	33.1
L-EKF ($\zeta = 1$)	100	711	20.3	23.3
L-EKF ($\zeta = 10$)	1000	1002	14.4	21.8
L-EKF ($\zeta = 20$)	100000	1021	14.1	21.5
L-EKF ($\zeta = 30$)	1000	910	15.8	21.7
G-EKF	1000	13779	1.0	22.1
	90% (478) validation detectors; spacing = 5.1 km			
No EKF	-	66	217.5	32.9
L-EKF ($\zeta = 1$)	100	681	21.1	28.1
L-EKF ($\zeta = 10$)	10	797	18.1	27.9
L-EKF ($\zeta = 20$)	10	792	18.2	27.8
L-EKF ($\zeta = 30$)	100	964	14.9	27.7
G-EKF	10	11670	1.2	27.9

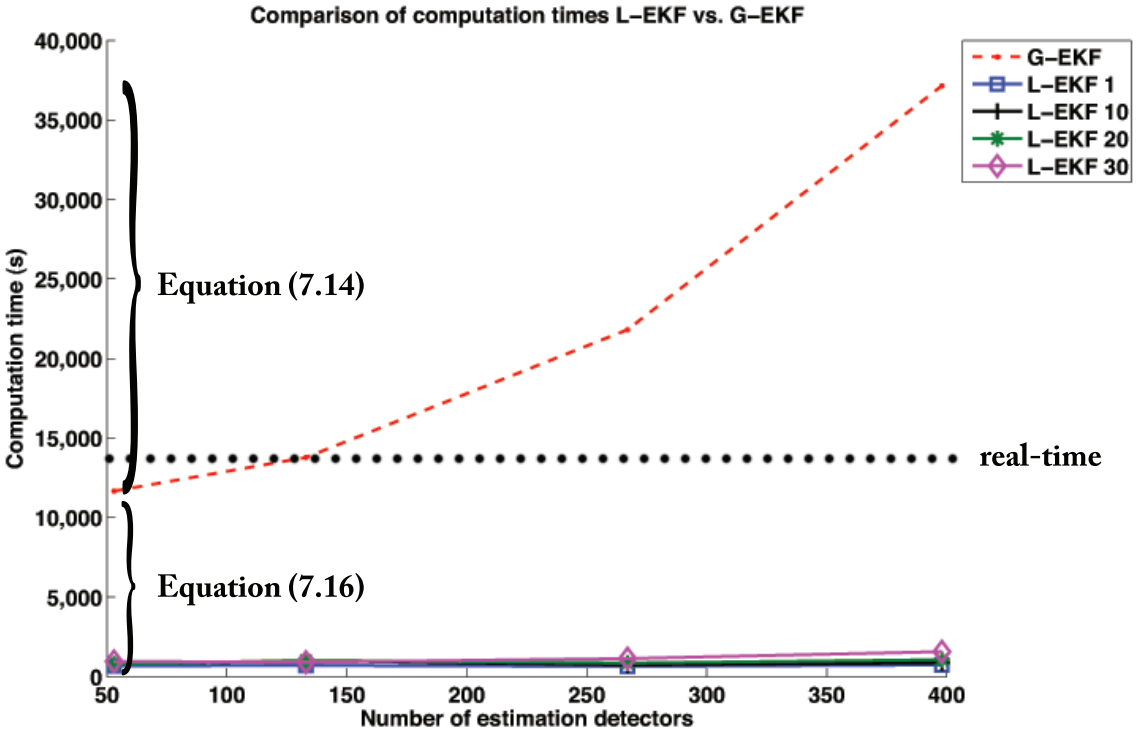


Figure 7.10: Computation times as a function of the number of detectors used for estimation. The L-EKFs show hardly an increase in computation time, while the G-EKF shows a rapidly increasing computation time

In Table 7.2 it can also be seen that the G-EKF is slower than real-time when more than 25% of the detectors are used for estimation. However, the L-EKF is still at least 9 times faster than real-time, even if 75% of the detectors are used for estimation. This means that even for a large network with hundreds of kilometers of road, such as the one used in this experiment, the L-EKF still is able to perform all necessary computations for the state estimation of the next minute within one minute.

7.5 Discussion and conclusion

In this chapter the Localized Extended Kalman Filter (L-EKF) has been proposed, opposed to the traditional Global EKF (G-EKF). The L-EKF is based on the observation that in the error covariance matrix $\boldsymbol{P}$ of a G-EKF many values are generally close to zero, leading to very small corrections which can be neglected. The L-EKF uses only local data in the physical vicinity of the measurement location, correcting only the states of the cells that have a considerable error covariance with the measured cell. The radius of the corrections is user-defined and influences the accuracy of the estimates on the one hand, and

the required computation time on the other hand. In two experiments, one with synthetic data and one with real data, it is found that for a sufficiently large radius, the accuracy of the L-EKF is equal to that of the G-EKF.

In the same experiments it has been shown that the order in which the local filters are used appears not to be important; in case the linearizations of the Extended Kalman Filter are accurate, it has been shown that the order in fact does not matter; otherwise a stochastic and unpredictable component exists in the calling order. Future study will need to validate if the calling order indeed never influences the accuracy to a considerable extend.

The L-EKF overcomes the major issue that has prevented the G-EKF being applied on a large scale: the calculation times of the G-EKF are very high on large-scale networks because the EKF procedure requires two expensive matrix operation, which scale to the power of 2.8 in the number of measured cells or the total number of cells in the network. In this chapter it has been shown that the complexity of the L-EKF scales linearly in the network size. Furthermore, in a real-world experiment on a large network it has been shown that the L-EKF was between 12 and 51 times faster, ensuring that it can still run within real-time, making it now possible to use the first order traffic flow model for real-time state estimation in large traffic networks, a task that was until now only possible on small-scale networks or corridors. This computation speed difference will be even larger when the size of the network or the number of measurements increases. Real-time application is therefore now possible, leaving time for additional computations for ATIS/DTM/MPC applications.

Based on the two experiments it can be stated that the L-EKF is always preferable over the G-EKF because it is faster and still delivers the same level of accuracy, even if the data is scarcely distributed. In the worst case experiment the average spacing between the detectors was 5.1 km. Even then the L-EKF delivered the same level of accuracy as the G-EKF. Of course, with fewer detectors both filters perform worse than when more detectors are used for estimation.

Increasing the radius of the L-EKF leads to higher computation times due to the overhead in copying values from the global matrices to local matrices and back, and due to the required matrix operations. However, when the network size is large, this overhead becomes negligible. Therefore, the radius of the local filters can safely be taken large to ensure a high accuracy, without increasing computation times considerably.

Future research needs to resolve several open questions. First of all, in this chapter many simulations were run with different fixed radii. Future research should investigate the possibility of predefining the optimal radius based on the expected influence area of a certain location, for example based on the shape and parameter values of the fundamental diagram. Also, it is possible that a dynamic radius of the L-EKF based on prevailing traffic conditions increases performance in accuracy and/or computation time. Finally, the authors believe that the same idea of localization could be applied to Unscented Kalman

Filter theory, other numerical solutions to the LWR models and other traffic flow models, making it possible to apply those theories to large scale networks as well with possibly higher accuracy than the LWR/EKF-combination used in this chapter.

Chapter 8

Conclusions and recommendations

In this thesis the Bayesian framework for data assimilation has been described. It has been applied to several problems in the traffic modeling domain, from an individual level (microscopic modeling of car-following behavior) to an aggregated level (network-wide state estimation). Furthermore, a fast new implementation of the Extended Kalman Filter has been proposed. In this chapter, first the conclusions are presented that are drawn based on the research that has been performed. Next, the implications for practitioners in the relevant fields of traffic modeling are treated. Finally, recommendations for future research are presented.

8.1 Conclusions

In this section, the main conclusions are presented that are drawn based on the studies that have been performed. First, recall the goal of this thesis that was defined in Chapter 1:

> *"to find a unified methodology for data assimilation for a wide range of models describing different road traffic phenomena, so that more accurate and consistent predictions can be made of the road traffic system".*

To reach this goal, first in the introduction a probabilistic perspective was chosen on the data assimilation problem. Then, a framework was found that uses Bayesian inference to develop equations for the validation & identification, calibration and estimation & prediction steps, based on the seminal work of Mackay (1992a, 1995) and Bishop (1995). Each of these steps are strongly interrelated, where generally the calibration task is performed first, after which the validation, identification, estimation and/or prediction can take place. Throughout the chapters of this thesis, this framework has been applied to a variety of traffic phenomena. In each chapter, the literature has been reviewed on the current state of practice in data assimilation, and the framework that was defined in Chapter 1 has been applied as an alternative.

The conclusions are organized in the following way. First, based on the literature reviews of all chapters together conclusions will be drawn on the current state of practice in data assimilation. Conclusions will be drawn on the Bayesian framework itself based on the experience of applying it to several problems. Then, conclusions are drawn in each field of traffic science individually to which the framework was applied. Finally, conclusions will be drawn on the Localized Extended Kalman Filter.

8.1.1 Current state of practice

In each chapter, the current state of practice has been studied by investigating the scientific literature for the specific problem at hand. As stated in the introduction, data assimilation consists of three steps: model validation/identification, model calibration and prediction/estimation. For the model identification step, the literature studies of each individual problem have revealed that:

- Usually many different models exist for the description/prediction of the same traffic phenomenon.
- Usually a model is chosen based on 'gut feeling' or experience, rather than based on numerical evidence that the chosen model is better than all alternatives.
- In the case when models are numerically compared, the comparison is based on one of the following indicators, each of which has issues:
  - *The prediction error of the last interval.* This approach has two problems: first, the prediction error of the last interval may not be available at the time a new prediction needs to be made, for example if the variable to be predicted is the travel time. Second, traffic is dynamic and stochastic so the prediction error is dynamic and stochastic too. Looking at the recent past for information on the current performance of models can thus be misguiding.
  - *The calibration error.* Calibrating models usually entails minimization of some performance measure. The values of these measures can be compared after each model is calibrated using the same method and the same data set. This approach tends to promote overly complex models because the absolute value of the calibration error is generally lower for models with more parameters, and it can even reach zero in case the number of parameters equals the number of data points. Comparing these values can therefore lead to the choice for models with low generalization ability[1] ('overfitted' models).

[1]The *generalization ability* reflects the notion that a model is able to predict the traffic phenomenon under consideration well, in all possible (likely) situations. A model that has a high generalization ability will therefore not only perform well on the data set that was used for calibration, but will also perform well in case new data is fed to the model.

  - *The validation error*. With this method, first a part of the data is used for calibration, and the performance of the models is then tested on the other part of the data. This requires the data set to be split in two leading to less data to be available for calibration and thus poorer predictions.
  - *The Likelihood-Ratio Test (LRT)*. This method overcomes all previously mentioned problems, but has one problem on its own: it can only be used on so-called *hierarchically nested models*, i.e. models where one model is a special case of the other. In general, the available models for description of traffic phenomena are not hierarchically nested.

  The Bayesian framework that is proposed in this thesis is a generalization of the Likelihood Ratio Test and is able to overcome all problems mentioned above. It balances the model fit with the model complexity, it allows all data to be used for calibration while still allowing for a numerical comparison of models and it is thus less sensitive to stochasticity and dynamics. It can be used to compare any set of models, also when they are not hierarchically nested.

- Usually only one model is studied or individual models are compared, but predictions of models are hardly ever combined in a 'committee'.

For the model calibration, the literature studies reveal that:

- Usually, single parameter values are found, while parameter *distributions* better represent the stochasticity of the traffic system.

- Prior information is hardly ever used in the calibration procedure. Because many parameters in traffic models usually have a physical meaning, it is a missed opportunity to improve the outcomes of the calibration procedure. Furthermore, the collected data does not always contain information on all parameters. In those cases, using prior information can prevent the parameters taking up unrealistic values based on random noise in the data.

Finally, for prediction/estimation, the literature reveals that:

- Usually, single values are predicted such as the travel time, while it may make sense to not only predict the most likely value, but also the prediction intervals. These prediction intervals may be directly communicated to the end user in Advanced Traffic Information Systems, but may also serve as an input to Dynamic Traffic Management systems. As the user's trust in the information is essential for effective management of traffic, this is an important missed opportunity.

8.1.2 Bayesian framework for data assimilation

For the framework in general, the following conclusions can be drawn:

- The Bayesian framework for data assimilation has proved to be a unified method and has been shown to be applicable to a wide range of models describing different road traffic phenomena.

- The application of the Bayesian framework for data assimilation generally leads to better performance (i.e. more accurate and with higher generalization ability) of the models that it is applied to.

- Assumptions need to be made explicit through the prior distributions on the parameters (model calibration) and the prior distributions of entire models (model identification). Given the assumptions and the data, Bayesian inference leads to an answer that is only as good as the assumptions and the data that were used as input.

One important feature of the Bayesian framework is that it leads to a numerical value for how good a model is expected to be (its generalization ability). This is called the *evidence* for a model, which can be used to compare a model to another model. The evidence balances how well a model fits on the data with the complexity of the model. A model with more parameters will always better fit to a data set, but will not necessarily make better predictions (have better generalization ability). A model with very few parameters may not be sophisticated enough to describe the problem at hand. The evidence measure balances between these two extremes. The evidence is calculated based on calibrated models. All available data can be used for calibration, because the evidence does not require the data set to be split up in two. In case data is scarce, this is a very beneficial property. The following conclusions can be drawn for this evidence measure:

- The evidence is preferable over other numerical comparison methods such as the Likelihood Ratio Test (LRT), because LRT requires models to be hierarchically nested while the evidence can be used to compare any set of models. In case the models are hierarchically nested, the outcomes of the two procedures are identical.

- In order to use the evidence for choosing between models, the correlation between evidence and the generalization ability needs to be strong. However, in studying this correlation in Chapter 5 it is found that this is not always the case because of the following possible problems:

  - The available data set that was used to represent the ‘ground truth’, i.e. to test the generalization ability, may be too small, i.e. the validation data set is not representative (enough) for the problem at hand.

  - The used models do not contain 'the perfect' model (the Bayesian inference framework makes a closed-world assumption, so that the probability that none of the alternatives is correct is zero, $P(\emptyset) = 0$). A weak correlation between evidence and generalization ability is an indication that the models require improvement.
  - Related to the previous point: the system under consideration contains 'system noise': not all explanatory factors will be present in the data. This can cause all models to fail, and a weak correlation between evidence and generalization ability.
  - The evidence is estimated using several assumptions: usually Gaussian distributions are assumed and some derivatives are approximated, such as the outer product approximation of the Hessian, in order to speed up calculations. The difference between generalization ability and evidence may be caused by a difference between the approximated evidence and the 'real' evidence.

- The evidence is useful for selecting high-potential models from a set of alternatives.
- The evidence is useful as a selection criterion and/or a weight in a model committee.
- Possible improvements to models, such as pruning (removing parameters, thus decreasing complexity) or using additional or alternative types of input data, can be evaluated using the evidence.

In several chapters of this thesis a *committee* was created: predictions of several models are combined. Concerning the committees, the following conclusions can be drawn:

- In all cases the use of a committee leads to improved prediction accuracy. Although the improvements are not spectacular, the additional effort to create a committee is very low in case the user already has multiple models at hand. Of course, running more models in parallel puts higher demands on computational power. The trade-off between more computation power and higher accuracy needs to be made for each application individually by the user.
- If all models have similar bias (for example, all models overestimate the quantity to be predicted), the committee generally leads to worse predictions than the best of the individual models. Increasing the heterogeneity of the available models very likely decreases the probability of all models having the same bias and thus leads to more accurate predictions.

In Chapter 4 and 5 error bars (prediction intervals) are constructed around the predictions. These error bars naturally follow from the Bayesian inference framework, because distributions are created on the data as well as on the parameters. A distribution thus

exists on the outputs. This distribution can be used to construct prediction intervals. This way, the end user receives information not only on the most likely traffic conditions, but also on the reliability of the information and of the traffic conditions. This will be useful in Advanced Traffic Information Systems, but also in case the predictions serve as input to Dynamic Traffic Management or Model Predictive Control systems.

8.1.3 Car-following behavior

The Bayesian framework for data assimilation has been applied to the problem of predicting car-following behavior in Chapter 2. Concerning this study, the following conclusions are drawn:

- One major issue in car-following behavior is driver heterogeneity: there are large inter-driver differences, so that one model may be best suited to one driver but another model to another driver. The Bayesian evidence has proved to be a useful tool for analyzing and *quantifying* these inter-driver differences. Using this tool, the driver heterogeneity can now be explicitly modeled.

- The Bayesian framework can also be used to construct the probability of models in an entire population (the distribution $P(H|D)$). This posterior distribution of the models can serve as a basis for a heterogeneous microscopic simulation.

8.1.4 Travel time prediction

In Chapters 3, 4 and 5 the Bayesian framework has been applied to travel time prediction. Concerning these studies, the following conclusions are drawn:

- There is a huge number of alternative models that have been used for travel time prediction. In most studies that have been published about these models, the authors compare in some way their model to a set of other models and conclude that the new model outperforms 'existing models'. However, in almost none of the studies the data assimilation is treated explicitly or consequently, so that the conclusion of better performance is at least dubious.

- Only a few studies exist where these models are combined in a committee. All of these use the error of the previous interval, but this error cannot be known in real-time because the travel time is only known after it has been realized. The Bayesian evidence is a solution to this problem that has proved in all three chapters to lead to improved prediction accuracy.

- In Appendix A the exact Hessian for recurrent neural networks is derived based on back-propagation theory. In the case study that was presented in Chapter 5 the

outer product approximation of the Hessian, which is much faster to compute than the exact Hessian, has proved to lead to well trained neural networks, but the exact Hessian may be preferable in case the evidence needs to be calculated accurately.

- In the case study of Chapter 5 it has been found that for a longer prediction horizon a recurrent layer in neural networks leads to better predictions, but that for a shorter horizon the added complexity of a recurrent layer does not help - in fact, for a 5-minute horizon the added complexity leads to overfitting and slightly worse predictions.

8.1.5 Extended Kalman Filter parameters

The same theories that were used for calibration of entire models have also been applied to the problem of defining the parameters of an Extended Kalman Filter that is combined with the LWR traffic model solved by the Godunov scheme. The derivation of equations for these parameters is very similar to those of the *hyperparameters* of training algorithms used with neural networks. Concerning this study, the following conclusions can be drawn:

- In the presented case study, the dynamic adaptation of the EKF parameters leads to almost equally accurate state estimates as when the optimal fixed EKF-parameters are used.

- However, the Bayesian adaptation of parameters leads to robustness of initial estimate of parameter values, while a wrongly chosen fixed EKF-parameter set may lead to a considerable loss in accuracy. This robustness is a very desirable property, as it is usually very hard to make an initial estimate of the variance of the model and of the data because no ground-truth is generally available.

- The Bayesian framework assumes Gaussian distributions on the data. The framework has been found to be sensitive to the distribution of the data not being Gaussian. In case of for example speeds of 0 km/h, the distribution cannot be Gaussian because negative speeds cannot exist. In that case, the covariance is overestimated leading to too small corrections of the state.

8.1.6 Localized Extended Kalman Filter

In Chapter 7 a new, fast and scalable implementation of the Extended Kalman Filter has been described: the Localized EKF (L-EKF). The L-EKF is an alternative method to compute the posterior distributions of the model state using the Kalman Filter equations. The L-EKF has been compared to the traditional ‘Global EKF’ (G-EKF) using the LWR

model solved by the Godunov scheme. Concerning this study, the following conclusions can be drawn:

- In the G-EKF many negligible corrections are made because the error covariance matrix contains many values close to zero. The L-EKF uses this fact, together with the network topology, to make only relevant corrections. A measurement is only used to correct the traffic state within the vicinity of the measurement location.
- The radius of the corrections is user-defined and influences the accuracy of the estimates on the one hand, and the required computation time on the other hand.
- In two experiments, one with synthetic data and one with real data, it is found that for a sufficiently large radius, the accuracy of the L-EKF is equal to that of the G-EKF.
- This result validates that the order in which the local filters are used is not important.
- The L-EKF overcomes the major issue that has prevented the G-EKF being applied on a large scale: the calculation times of the G-EKF are very high on large-scale networks because it requires expensive matrix operations which scale to the power of 2.8 in the number of cells or the number of measurements in the network. In the study it has been shown that the complexity of the L-EKF scales linearly with the network size.
- Because the L-EKF scales much better in the network size, it is now possible to use the Extended Kalman Filter on a very large scale. Real-time application is possible, leaving time for additional computations for Advanced Traveler Information Systems, Dynamic Traffic Management systems or Model Predictive Control systems.
- Based on the two experiments I have the opinion that the L-EKF is always preferable over the G-EKF because it is faster while maintaining the same level of accuracy, even if the data is scarcely distributed over the network.

8.2 Implications for practitioners

This research has aimed to provide tools for practitioners and researchers in the field of traffic information and traffic management that enable them to optimally use their models in combination with data. One of the most important notions of this thesis is that models and data go hand in hand and should always be treated together. The Bayesian inference framework is one way of approaching this, which has been shown to have various benefits as presented in the conclusions before.

For practitioners, the research has the following implications:

- The Bayesian framework is a single framework for model validation, model identification, model calibration and estimation/prediction.
- In Bayesian inference, assumptions need to be made explicit. Whether or not this is a good feature is part of a long debate between Bayesians and frequentists. In any case, the outcomes of the model identification, calibration and prediction steps are only as good as the assumptions that were made and the data that was used for each of the steps.
- This framework has been shown to be applicable to a variety of problems, such as car-following prediction, travel time prediction and continuous calibration of EKF parameters. The exact same ideas can be applied to any problem in traffic for which one or more models are available.
- Using the evidence, models can be compared based on a numerical measure. The comparison can be made while all data can still be used for calibration.

8.3 Recommendations and future research

In this final section of the thesis, several new applications and fields of study are identified which may direct future research. These new fields of study fall outside the scope of this thesis, but are deemed to be able to benefit from the Bayesian framework for data assimilation. As with the conclusions, these questions are first stated for the framework as a whole. Then, possible future research is defined for each application separately.

8.3.1 Bayesian framework for data assimilation

During this thesis, the Bayesian framework has been applied to a variety of problems: car-following behavior, travel time prediction and data assimilation for macroscopic traffic modeling.

In Chapters 4 and 5 the correlation between evidence and generalization ability was found not to be perfect. As noted before, a weak correlation between the evidence and the generalization ability is an indication that the models require improvement. Another way to look at this is that the probability of the empty set may not be zero: all models that are used may be wrong, i.e. the selected set of models does not contain the 'perfect' model. Recall from 1.4 that the Transferable Belief Model (TBM) explicitly takes this possibility into account. Now that the Bayesian framework has proven to be very useful in data assimilation in a wide variety of applications in traffic science, the TBM may be applied in a similar fashion. Because the Bayesian framework is a special case of the TBM, this thesis has laid the basis for such research.

Several ideas exist to apply the Bayesian framework to different problems, such as:

- In modeling of pedestrian behavior, multiple models exist to describe walking behavior. Calibration of these models is a challenge because ground-truth data is generally scarce compared to the number of parameters of the models. The Bayesian framework may lead to better answers to how well these models perform relative to each other, even in the case when data is scarce.

- Using one or more (data-driven) models for prediction of other traffic variables than travel time. The prediction of traffic flow, for example at an onramp, can be useful as an estimate of the demand at origins in a network-wide traffic state prediction. Also, predicted route choice, or aggregated route choice represented by split rates or turn fractions can serve as a parameter in the same network-wide traffic state prediction. Finally, there is often an interest to predict variables like level-of-service, crash rates and incident duration for a variety of applications.

- Using the framework for OD-matrix prediction. As with the other traffic phenomena, a multitude of models exist for the prediction of Origin-Destination matrices. The Bayesian evidence can be used to put a number to how well each model performs compared to the others. The OD-estimation problem is at the same time one of the most underdetermined problems in the field of traffic. The inclusion of prior knowledge may thus be a crucial factor in solving this problem. In the work of Bell (1991), prior information is for example already included in a generalized least squares approach, an approach that is easily extended to the Bayesian one. As an additional benefit, the Bayesian framework allows for estimation of the uncertainty of the predictions, which is a very desirable feature if the problem is so underdetermined.

8.3.2 Car-following behavior

In the application of the Bayesian framework to car-following behavior, the following recommendations are made for future research:

- Besides of inter-driver differences there are also intra-driver differences: one driver does not always behave according to the same model, but may change his behavior stochastically or depending on conditions. The use of a heterogeneous pool of models (a committee) for one single driver may increase the robustness towards this changing behavior and may increase the accuracy with which driver behavior can be predicted.

- Error bars have not yet been constructed on the predicted car-following behavior. In some applications, it may make sense to do so: for example when predicting the trajectory of a single driver in vehicle-to-vehicle or vehicle-to-roadside architectures.

- As was proposed in Chapter 2, the work that has been done has paved the road for a heterogeneous microscopic simulation. In this simulation, different models exist for the same task. Using a large data set, the Bayesian framework is used to find a distribution of the probability of a model best describing a driver's behavior. Each time a new car is entered into the network, first a model is drawn from this distribution. Then, for this model parameter values are drawn from the posterior distribution of the parameters. The vehicle can then be simulated through the network using his model and his parameter set. Experiments should then investigate if such a heterogeneous simulation better describes the traffic system than when a single model is used for all drivers.

8.3.3 Travel time prediction

In the application of the Bayesian framework to travel time prediction, the following open questions for future research have risen:

- In all applications the prior distributions of models have been taken equal ('flat') for all models. However, inclusion of prior knowledge may improve results: better assumptions to start with lead to better results from the Bayesian inference. Future study should investigate ways to find prior knowledge, for example by investigating literature comparatively.

- In Chapters 3 - 5 the Bayesian framework was applied to at maximum two *types* of models, although many neural networks with different structures have been trained for Chapter 4 and 5. In literature, literally hundreds of models have been found that have been used for prediction of traffic variables (van Hinsbergen et al., 2007). It is an interesting research project to test a multitude of models in one or more large scale experiments with the Bayesian framework. Such a study may be used as the basis for future practitioners to a priori select potentially well-functioning models, so that they do not have to test all possible models that have been developed over the last decades.

- It was found that for longer prediction horizons recurrent neural networks perform better than feed-forward neural networks, while for shorter horizons this is the other way around. Future study should validate if this result holds in general. If so, it is an interesting feature that can perhaps serve in a priori model selection, which can save work for practitioners.

8.3.4 Extended Kalman Filter parameters

In Chapter 6 the Bayesian theories have been applied to the continuous estimation of the parameters of the Extended Kalman Filter. The following open questions have risen:

- In the current approach, only the data covariance matrix is continuously changed, because the equations could not be solved for the model covariance. Future research can possibly further improve results by trying to find alternative ways to set the model covariance, for example by numerical estimates of the covariance or by approximation of some of the equations.
- The solution is found to be sensitive to bias in the data. One possible solution is not to use a Gaussian distribution but another, non-symmetric distribution. An analytical solution of the equations is in those cases probably harder, so assumptions or numerical approximations may be needed.
- The continuous adaptation of the EKF parameters has been tested on the 'Global' EKF. Future study should see if the same good results are obtained if they are applied to the Localized EKF of Chapter 7.

8.3.5 Localized Extended Kalman Filter

In Chapter 7 the Localized EKF has been proposed as an alternative implementation of the traditional Global EKF. The following future research topics are of interest:

- The order in which the L-EKFs are called are now based on the order in which data arrives in the processing computer. Future research should try to confirm the result that the order in which the filters are called is not important. If this result is found not to be general, ways should be proposed to optimally set the calling order of the sequential filters.
- In the study many simulations were run with different fixed radii. Future research should investigate the possibility of predefining the optimal radius based on the expected influence area of a certain location, for example based on the shape and parameter values of the fundamental diagram.
- Alternatively or additionally it is possible that a dynamic radius of the L-EKF based on prevailing traffic conditions increases performance in accuracy and/or computation time.
- The same ideas of localization can be applied to other filters, for example the Unscented Kalman Filter. Furthermore, it is interesting to validate the localization with other numerical solutions to the first order model or other models such as second or higher order models.

Appendix A

Exact gradient and Hessian for Recurrent Neural Networks

In this appendix the exact gradient and Hessian for recurrent neural networks are derived. In this appendix the definitions as given in Chapter 4 and 5 are used as a basis. In Figure 5.1 the layout of a State Space Neural Network (SSNN) can be seen, which is a special form of a general Recurrent Neural Network (RNN) with certain weights set to zero. The derivation for the exact gradient and Hessian hold for both the RNN as well as the SSNN.

A.1 Determination of the gradient

To determine the direction for each step in the conjugate gradient algorithm, the gradient of the error function to the weights is needed. The data error E_D and the regularizer errors $E_{W,v}$ will be considered separately, so:

$$\nabla E(\boldsymbol{\theta}) = \beta \nabla E_D + \nabla \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{A.1}$$

The derivative of the second term, the gradient of the weight errors (regularizers), is straightforward:

$$\nabla \sum_{v=1}^{V} \alpha_v E_{W,v} = \sum_{v=1}^{V} \alpha_v \boldsymbol{I}_v \boldsymbol{\theta}_v \tag{A.2}$$

where $\boldsymbol{I}_v$ is a matrix with all elements zero except for some diagonal elements $\boldsymbol{I}_{ii} = 1$ where i is the index in the weight vector $\boldsymbol{\theta}$ of a weight belonging to a group v.

The gradient of E_D is more complex. Because this term is a summation over all N input patterns, the gradient of the error over one pattern n (which is equivalent to the error at time step t as noted before) can first be considered, which is defined as $E_D^t =$

$\frac{1}{2}\sum_k (y_k^t - o_k^t)^2$, and later be summed over all patterns N to obtain the full gradient. Define the part of the error from one output k as $E_{D,k}^t = \frac{1}{2}(y_k^t - o_k^t)^2$. For an arbitrary weight θ_q in any layer of the network, it holds that it only influences E_D^t through the outputs y_k, so the chain rule for partial derivatives can be applied:

$$\begin{aligned}\frac{\partial E_D^t}{\partial \theta_q} &= \sum_k \frac{\partial E_{D,k}^t}{\partial y_k}\frac{\partial y_k}{\partial \theta_q} \\ &= \frac{1}{2}\sum_k \frac{\partial}{\partial y_k}\left(y_k^t - o_k^t\right)^2 \frac{\partial y_k}{\partial \theta_q} \\ &= \sum_k \left(y_k^t - o_k^t\right)\frac{\partial y_k}{\partial \theta_q} \\ &= \sum_k \delta_k^t \frac{\partial y_k}{\partial \theta_q} \end{aligned} \tag{A.3}$$

where $\delta_k^t = (y_k^t - o_k^t)$.

A.1.1 Determination of $\partial y/\partial w$

In this section, for each weight in the recurrent neural network the derivative (A.3) will be determined.

For a weight θ_{kj} in the output layer, it holds that:

$$\begin{aligned}\frac{\partial y_{k'}}{\partial \theta_{kj}} &= \frac{\partial}{\partial \theta_{kj}} f_2(a_{k'}^t) \\ &= f_2'(a_{k'}^t)\frac{\partial a_{k'}^t}{\partial \theta_{kj}} \\ &= f_2'(a_{k'}^t)\sum_{j'} \Delta_{jj'}\Delta_{kk'} z_{j'}^t \\ &= \Delta_{kk'} f_2'(a_{k'}^t) z_j^t \end{aligned} \tag{A.4}$$

with Δ_{kk} the Kronecker delta function.

For a weight θ_{ji} in the hidden layer:

$$\begin{aligned}
\frac{\partial y_k^t}{\partial \theta_{ji}} &= \frac{\partial}{\partial \theta_{ji}} f_2(a_k^t) \\
&= f_2'(a_k^t) \frac{\partial a_k^t}{\partial \theta_{ji}} \\
&= f_2'(a_k^t) \sum_{j'} \theta_{kj'} f_1'(a_{j'}^t) \frac{\partial a_{j'}^t}{\partial \theta_{ji}} \\
&= f_2'(a_k^t) \sum_{j'} \theta_{kj'} f_1'(a_{j'}^t) \left(\Delta_{jj'} x_i^t + \sum_l \theta_{j'l} f_1'(a_l^{t-1}) \frac{\partial a_l^{t-1}}{\partial \theta_{ji}} \right)
\end{aligned} \tag{A.5}$$

Define:

$$\omega_{j'ji}^t = \frac{\partial a_{j'}^t}{\partial \theta_{ji}} = \Delta_{j'j} x_i^t + \sum_l \theta_{j'l} f_1'(a_l^{t-1}) \omega_{lji}^{t-1} \tag{A.6}$$

and

$$h_{kji}^t = \frac{\partial a_k^t}{\partial \theta_{ji}} = \sum_{j'} \theta_{kj'} f_1'(a_{j'}^t) \omega_{j'ji}^t \tag{A.7}$$

Equation (A.5) then becomes:

$$\frac{\partial y_k^t}{\partial \theta_{ji}} = f_2'(a_k^t) h_{kji}^t \tag{A.8}$$

The starting condition for ω follows from the fact that if $t = 1$, the context layer contains constant values C, so that

$$\begin{aligned}
\omega_{j'ji}^1 &= \frac{\partial}{\partial \theta_{ji}} \left(\sum_{i'} \theta_{j'i'} x_{i'}^1 + \sum_l \theta_{j'l} C \right) \\
&= \Delta_{j'j} x_i^1
\end{aligned} \tag{A.9}$$

For a weight θ_{jl} in the context layer:

$$\begin{aligned}\frac{\partial y_k^t}{\partial \theta_{jl}} &= \frac{\partial}{\partial \theta_{jl}} f_2(a_k^t) \\ &= f_2'(a_k^t)\frac{\partial a_k^t}{\partial \theta_{jl}} \\ &= f_2'(a_k^t)\sum_{j'}\theta_{kj'}f_1'(a_{j'}^t)\frac{\partial a_{j'}^t}{\partial \theta_{jl}} \\ &= f_2'(a_k^t)\sum_{j'}\theta_{kj'}f_1'(a_{j'}^t)\left(\Delta_{j'j}z_l^{t-1} + \sum_{l'}\theta_{j'l'}f_1'(a_{l'}^{t-1})\frac{\partial a_{l'}^{t-1}}{\partial \theta_{jl}}\right)\end{aligned} \tag{A.10}$$

Define η by:

$$\eta_{j'jl}^t = \frac{\partial a_{j'}}{\partial \theta_{jl}} = \Delta_{j'j}z_l^{t-1} + \sum_{l'}\theta_{j'l'}f_1'(a_{l'}^{t-1})\eta_{l'jl}^{t-1} \tag{A.11}$$

and

$$g_{kjl}^t = \frac{\partial a_k^t}{\partial \theta_{jl}} = \sum_{j'}\theta_{kj'}f_1'(a_{j'}^t)\eta_{j'jl}^t \tag{A.12}$$

Equation (A.10) then becomes:

$$\frac{\partial y_k^t}{\partial \theta_{jl}} = f_2'(a_k^t)g_{kjl}^t \tag{A.13}$$

The starting condition for η is:

$$\eta_{j'jl}^1 = \frac{\partial}{\partial \theta_{jl}}\left(\sum_i \theta_{j'i}x_i^1 + \sum_{l'}\theta_{j'l'}C\right) = \Delta_{j'j}C \tag{A.14}$$

A.1.2 The gradients for each layer

Using (A.3), (A.4), (A.5) and (A.10) the gradient of the error function can now be constructed for each layer. For the output layer:

$$\begin{aligned}\frac{\partial E_D^t}{\partial \theta_{kj}} &= \sum_{k'}\delta_{k'}^t\frac{\partial y_{k'}^t}{\partial \theta_{kj}} \\ &= \sum_{k'}\Delta_{kk'}\delta_{k'}^t f_2'(a_{k'}^t)z_j^t \\ &= f_2'(a_k^t)\delta_k^t z_j^t\end{aligned} \tag{A.15}$$

For the hidden layer:

$$\begin{aligned}\frac{\partial E_D^t}{\partial \theta_{ji}} &= \sum_k \delta_k^t \frac{\partial y_k^t}{\partial \theta_{ji}} \\ &= \sum_k f_2'(a_k^t)\delta_k^t h_{kji}^t \end{aligned} \tag{A.16}$$

For the context layer:

$$\begin{aligned}\frac{\partial E_D^t}{\partial \theta_{jl}} &= \sum_k \delta_k^t \frac{\partial y_k^t}{\partial \theta_{jl}} \\ &= \sum_k f_2'(a_k^t)\delta_k^t g_{kjl}^t \end{aligned} \tag{A.17}$$

Note that in an actual application the values for ω and η have already been calculated in the previous time step and can be kept in memory for reference in the next time step.

The total gradient of the error function can now be obtained by concatenating all values into a vector of size W (the total number of weights in the network) and summing over all n. The gradient term of (A.2) is then added to obtain the entire gradient.

A.2 Determination of the Hessian

To determine the step size in the conjugate gradient algorithm and to calculate the Bayesian evidence the Hessian $\boldsymbol{A}$ is needed, which is considered separately for the two error parts E_D and E_W:

$$\boldsymbol{A} = \nabla^2 E(\boldsymbol{\theta}) = \beta \nabla^2 E_D + \nabla^2 \sum_{v=1}^{V} \alpha_v E_{W,v} \tag{A.18}$$

The second term again is straightforward:

$$\nabla^2 \sum_{v=1}^{V} \alpha_v E_{W,v} = \sum_{v=1}^{V} \alpha_v \boldsymbol{I}_v \tag{A.19}$$

The first term, the error part E_D, is first considered per pattern n (time step t), E_D^t, and later summed over all n to obtain the full value. Using the previously derived first derivatives of (A.15) - (A.17), the second derivatives can be found one by one.

A.2.1 Both output layer

$$\begin{aligned}\frac{\partial^2 E_D^t}{\partial\theta_{kj}\partial\theta_{k'j'}} &= \frac{\partial}{\partial\theta_{kj}}\left(\frac{\partial E_D^t}{\partial\theta_{k'j'}}\right)\\ &= \frac{\partial}{\partial\theta_{kj}}\left(f_2'(a_{k'}^t)\delta_{k'}^t z_{j'}^t\right)\\ &= z_{j'}^t\left(f_2'(a_{k'}^t)\frac{\partial\delta_{k'}^t}{\partial\theta_{kj}} + \delta_{k'}^t\frac{\partial f_2'(a_{k'}^t)}{\partial\theta_{kj}}\right)\\ &= \Delta_{kk'}z_{j'}^t z_j^t\left(\left(f_2'(a_{k'}^t)\right)^2 + f_2''(a_{k'}^t)\delta_{k'}^t\right)\end{aligned} \tag{A.20}$$

Note that if the output function is linear, $f_2'(a) = 1$ and $f_2''(a) = 0$, and the result reduces to $\Delta_{kk'}z_{j'}^t z_j^t$.

A.2.2 Output layer and hidden layer

$$\begin{aligned}\frac{\partial^2 E_D^t}{\partial\theta_{ji}\partial\theta_{kj'}} &= \frac{\partial}{\partial\theta_{ji}}\left(\frac{\partial E_D^t}{\partial\theta_{kj'}}\right)\\ &= \frac{\partial}{\partial\theta_{ji}}\left(f_2'(a_k^t)\delta_k^t z_{j'}^t\right)\\ &= f_2'(a_k^t)\left(z_{j'}^t\frac{\partial\delta_k^t}{\partial\theta_{ji}} + \delta_k^t\frac{\partial z_{j'}^t}{\partial\theta_{ji}}\right) + \delta_k^t z_{j'}^t f_2''(a_k^t)\frac{\partial a_k^t}{\partial\theta_{ji}}\\ &= f_2'(a_k^t)\left(z_{j'}^t f_2'(a_k^t)h_{kji}^t + \delta_k^t f_1'(a_{j'}^t)\omega_{j'ji}^t\right) + f_2''(a_k^t)\delta_k^t z_{j'}^t h_{kji}^t\end{aligned} \tag{A.21}$$

Note that in case of a linear output, the last term vanishes as in that case $f_2''(a) = 0$.

A.2.3 Output layer and context layer

$$\begin{aligned}\frac{\partial^2 E_D^t}{\partial\theta_{jl}\partial\theta_{kj'}} &= \frac{\partial}{\partial\theta_{jl}}\left(\frac{\partial E_D^t}{\partial\theta_{kj'}}\right)\\ &= \frac{\partial}{\partial\theta_{jl}}\left(f_2'(a_k^t)\delta_k^t z_{j'}^t\right)\\ &= f_2'(a_k^t)\left(z_{j'}^t f_2'(a_k^t)g_{kjl}^t + \delta_k^t f_1'(a_{j'}^t)\eta_{j'jl}^t\right) + f_2''(a_k^t)\delta_k^t z_{j'}^t g_{kjl}^t\end{aligned} \tag{A.22}$$

Note that in case of a linear output, the last term vanishes as in that case $f_2''(a) = 0$.

A.2.4 Both hidden layer

$$\begin{aligned}
\frac{\partial^2 E_D^t}{\partial\theta_{ji}\partial\theta_{j'i'}} &= \frac{\partial}{\partial\theta_{ji}}\left(\frac{\partial E_D^t}{\partial\theta_{j'i'}}\right) \\
&= \frac{\partial}{\partial\theta_{ji}}\left(\sum_k f_2'(a_k^t)\delta_k^t h_{kj'i'}^t\right) \\
&= \sum_k\left[f_2'(a_k^t)\left(h_{kj'i'}^t\frac{\partial\delta_k^t}{\partial\theta_{ji}} + \delta_k^t\frac{\partial h_{kj'i'}}{\partial\theta_{ji}}\right) + \delta_k^t h_{kj'i'}^t f_2''(a_k^t)\frac{\partial a_k^t}{\partial\theta_{ji}}\right] \\
&= \sum_k\left[f_2'(a_k^t)\left(f_2'(a_k^t)h_{kji}^t h_{kj'i'}^t + \delta_k^t\psi_{kj'i'ji}^t\right) + f_2''(a_k^t)\delta_k^t h_{kji}^t h_{kj'i'}^t\right] \qquad \text{(A.23)}
\end{aligned}$$

where the auxiliary variable ψ is defined as:

$$\begin{aligned}
\psi_{kj'i'ji}^t &= \frac{\partial h_{kj'i'}^t}{\partial\theta_{ji}} \\
&= \frac{\partial}{\partial\theta_{ji}}\sum_{j''}\theta_{kj''}f_1'(a_{j''}^t)\omega_{j''j'i'}^t \\
&= \sum_{j''}\theta_{kj''}\left(\omega_{j''j'i'}^t f_1''(a_{j''}^t)\frac{\partial a_{j''}^t}{\partial\theta_{ji}} + f_1'(a_{j''}^t)\frac{\partial\omega_{j''j'i'}^t}{\partial\theta_{ji}}\right) \\
&= \sum_{j''}\theta_{kj''}\left(f_1''(a_{j''}^t)\omega_{j''j'i'}^t\omega_{j''ji}^t + f_1'(a_{j''}^t)\chi_{j''j'i'ji}^t\right) \qquad \text{(A.24)}
\end{aligned}$$

with χ given by:

$$\begin{aligned}
\chi_{j''j'i'ji}^t &= \frac{\partial\omega_{j''j'i'}^t}{\partial\theta_{ji}} \\
&= \frac{\partial}{\partial\theta_{ji}}\left(\Delta_{j''j'}x_{i'}^t + \sum_l\theta_{j''l}f_1'(a_l^{t-1})\omega_{lj'i'}^{t-1}\right) \\
&= \sum_l\theta_{j''l}\left(\omega_{lj'i'}^{t-1}f_1''(a_l^{t-1})\frac{\partial a_l^{t-1}}{\partial\theta_{ji}} + f_1'(a_l^{t-1})\frac{\partial\omega_{l,j'i'}^{t-1}}{\partial\theta_{ji}}\right) \\
&= \sum_l\theta_{j''l}\left(f_1''(a_l^{t-1})\omega_{lj'i'}^{t-1}\omega_{lji}^{t-1} + f_1'(a_l^{t-1})\chi_{lj'i'ji}^{t-1}\right) \qquad \text{(A.25)}
\end{aligned}$$

with the starting condition for χ:

$$\chi_{j''j'i'ji}^1 = \frac{\partial}{\partial\theta_{ji}}\left(\Delta_{j''j'}x_{i'}^1\right) = 0 \qquad \text{(A.26)}$$

A.2.5 Hidden layer and context layer

$$\begin{aligned}
\frac{\partial^2 E_D^t}{\partial\theta_{ji}\partial\theta_{j'l}} &= \frac{\partial}{\partial\theta_{ji}}\left(\frac{\partial E_D^t}{\partial\theta_{j'l}}\right) \\
&= \frac{\partial}{\partial\theta_{ji}}\left(\sum_k f_2'(a_k^t)\delta_k^t g_{kj'l}^t\right) \\
&= \sum_k \left[f_2'(a_k^t)\left(g_{kj'l}^t\frac{\partial\delta_k^t}{\partial\theta_{ji}} + \delta_k^t\frac{\partial g_{kj'l}^t}{\partial\theta_{ji}}\right) + \delta_k^t g_{kj'l}^t f_2''(a_k^t)\frac{\partial a_k^t}{\partial\theta_{ji}}\right] \\
&= \sum_k \left[f_2'(a_k^t)\left(f_2'(a_k^t)g_{kj'l}^t h_{kji}^t + \delta_k^t\phi_{kj'lji}^t\right) + f_2''(a_k^t)\delta_k^t g_{kj'l}^t h_{kji}^t\right] \qquad \text{(A.27)}
\end{aligned}$$

with ϕ defined by

$$\begin{aligned}
\phi_{kj'lji}^t &= \frac{\partial g_{kj'l}^t}{\partial\theta_{ji}} \\
&= \frac{\partial}{\partial\theta_{ji}}\left(\sum_{j''}\theta_{kj''}f_1'(a_{j''}^t)\eta_{j''j'l}^t\right) \\
&= \sum_{j''}\theta_{kj''}\left(\eta_{j''j'l}^t f_1''(a_{j''}^t)\frac{\partial a_{j''}^t}{\partial\theta_{ji}} + f_1'(a_{j''}^t)\frac{\partial\eta_{j''j'l}^t}{\partial\theta_{ji}}\right) \\
&= \sum_{j''}\theta_{kj''}\left(f_1''(a_{j''}^t)\eta_{j''j'l}^t\omega_{j''ji}^t + f_1'(a_{j''}^t)\upsilon_{j''j'lji}^t\right) \qquad \text{(A.28)}
\end{aligned}$$

and υ by

$$\begin{aligned}
\upsilon_{j''j'lji}^t &= \frac{\partial\eta_{j''j'l}^t}{\partial\theta_{ji}} \\
&= \frac{\partial}{\partial\theta_{ji}}\left(\Delta_{j''j'}f_1(a_l^{t-1}) + \sum_{l'}\theta_{j''l'}f_1'(a_{l'}^{t-1})\eta_{l'j'l}^{t-1}\right) \\
&= \Delta_{j''j'}f_1'(a_l^{t-1})\frac{\partial a_l^{t-1}}{\partial\theta_{ji}} + \sum_{l'}\theta_{j''l'}\left(\eta_{l'j'l}^{t-1}f_1''(a_{l'}^{t-1})\frac{\partial a_{l'}^{t-1}}{\partial\theta_{ji}} + f_1'(a_{l'}^{t-1})\upsilon_{l'j'lji}^{t-1}\right) \\
&= \Delta_{j''j'}f_1'(a_l^{t-1})\omega_{lji}^{t-1} + \sum_{l'}\theta_{j''l'}\left(\eta_{l'j'l}^{t-1}f_1''(a_{l'}^{t-1})\omega_{l'ji}^{t-1} + f_1'(a_{l'}^{t-1})\upsilon_{l'j'lji}^{t-1}\right) \qquad \text{(A.29)}
\end{aligned}$$

The starting condition for υ equals

$$\upsilon_{j''j'lji}^1 = \frac{\partial}{\partial\theta_{ji}}\left(\Delta_{j''j'}C\right) = 0 \qquad \text{(A.30)}$$

A.2.6 Both context layer

$$\begin{aligned}\frac{\partial^2 E_D^t}{\partial\theta_{jl}\partial\theta_{j'l'}} &= \frac{\partial}{\partial\theta_{jl}}\left(\frac{\partial E_D^t}{\partial\theta_{j'l'}}\right)\\ &= \frac{\partial}{\partial\theta_{jl}}\left(\sum_k f_2'(a_k^t)\delta_k^t g_{kj'l'}^t\right)\\ &= \sum_k\left[f_2'(a_k^t)\left(g_{kj'l'}^t\frac{\partial\delta_k^t}{\partial\theta_{jl}}+\delta_k^t\frac{\partial g_{kj'l'}^t}{\partial\theta_{jl}}\right)+\delta_k^t g_{kj'l'}^t f_2''(a_k^t)\frac{\partial a_k^t}{\partial\theta_{jl}}\right]\\ &= \sum_k\left[f_2'(a_k^t)\left(f_2'(a_k^t)g_{kj'l'}^t g_{kjl}^t+\delta_k^t\tau_{kj'l'jl}^t\right)+f_2''(a_k^t)\delta_k^t g_{kj'l'}^t g_{kjl}^t\right]\end{aligned}\tag{A.31}$$

with τ

$$\begin{aligned}\tau_{kj'l'jl}^t &= \frac{\partial g_{kj'l'}^t}{\partial\theta_{jl}}\\ &= \frac{\partial}{\partial\theta_{jl}}\left(\sum_{j''}\theta_{kj''}f_1'(a_{j''}^t)\eta_{j''j'l'}^t\right)\\ &= \sum_{j''}\theta_{kj''}\left(\eta_{j''j'l'}^t f_1''(a_{j''}^t)\frac{\partial a_{j''}^t}{\partial\theta_{jl}}+f_1'(a_{j''}^t)\frac{\partial\eta_{j''j'l'}^t}{\partial\theta_{jl}}\right)\\ &= \sum_{j''}\theta_{kj''}\left(f_1''(a_{j''}^t)\eta_{j''j'l'}^t\eta_{j''jl}^t+f_1'(a_{j''}^t)\varsigma_{j''j'l'jl}^t\right)\end{aligned}\tag{A.32}$$

and ς

$$\begin{aligned}\varsigma_{j''j'l'jl}^t &= \frac{\partial\eta_{j''j'l'}^t}{\partial\theta_{jl}}\\ &= \frac{\partial}{\partial\theta_{jl}}\left(\Delta_{j''j'}f_1(a_{l'}^{t-1})+\sum_{l''}\theta_{j''l''}f_1'(a_{l''}^{t-1})\eta_{l''j'l'}^{t-1}\right)\\ &= \Delta_{j''j'}f_1'(a_{l'}^{t-1})\frac{\partial a_{l'}^{t-1}}{\partial\theta_{jl}}+\sum_{l''}\theta_{j''l''}\left(\eta_{l''j'l'}^{t-1}f_1''(a_{l''}^{t-1})\frac{\partial a_{l''}^{t-1}}{\partial\theta_{jl}}+f_1'(a_{l''}^{t-1})\frac{\partial\eta_{l''j'l'}^{t-1}}{\partial\theta_{jl}}\right)\\ &= \Delta_{j''j'}f_1'(a_{l'}^{t-1})\eta_{l'jl}^{t-1}+\sum_{l''}\theta_{j''l''}\left(f_1''(a_{l''}^{t-1})\eta_{l''j'l'}^{t-1}\eta_{l''jl}^{t-1}+f_1'(a_{l''}^{t-1})\varsigma_{l''j'l'jl}^{t-1}\right)\end{aligned}\tag{A.33}$$

with the starting condition for ς

$$\varsigma_{j''j'l'jl}^1 = \frac{\partial}{\partial\theta_{jl}}\left(\Delta_{j''j'}C\right)=0\tag{A.34}$$

The final Hessian is obtained by concatenating all the values into a matrix of size W by W, by summing over all t and by adding the part of equation (A.19).

A.3 Outer product approximation of the Hessian

Because the exact evaluation of the Hessian may become slow, an approximation of the Hessian is sometimes required. Consider the sum-of-squares error function, which is repeated here for convenience:

$$E_D = \frac{1}{2} \sum_{t=1}^{N} \sum_{k=1}^{c} \left(y_k^t - o_k^t\right)^2 \tag{A.35}$$

then the second derivative of E_D to two arbitrary weights θ_q and θ_r anywhere in the network can be written in the form

$$\begin{aligned} \frac{\partial^2 E_D}{\partial\theta_q \partial\theta_r} &= \frac{\partial^2}{\partial\theta_q \partial\theta_r} \left[\frac{1}{2} \sum_{t=1}^{N} \sum_{k=1}^{c} \left((y_k^t)^2 - 2 y_k^t o_k^t + (o_k^t)^2 \right) \right] \\ &= \frac{\partial}{\partial\theta_r} \left[\sum_{t=1}^{N} \sum_{k=1}^{c} \frac{\partial y_k^t}{\partial\theta_q} \left(y_k^t - o_k^t\right) \right] \\ &= \sum_{t=1}^{N} \sum_{k=1}^{c} \frac{\partial y_k^t}{\partial\theta_q} \frac{\partial y_k^t}{\partial\theta_r} + \sum_{t=1}^{N} \sum_{k=1}^{c} \left(y_k^t - o_k^t\right) \frac{\partial^2 y_k^t}{\partial\theta_q \partial\theta_r} \end{aligned} \tag{A.36}$$

As the quantity $(y_k^t - o_k^t)$ is a random variable with zero mean, uncorrelated with the value of the second derivative term, this whole term will tend to average to zero in the summation over t (Hassibi and Stork, 1993). This term can therefore be neglected:

$$\frac{\partial^2 E_D}{\partial\theta_q \partial\theta_r} \approx \sum_{t=1}^{N} \sum_{k=1}^{c} \frac{\partial y_k^t}{\partial\theta_q} \frac{\partial y_k^t}{\partial\theta_r} \tag{A.37}$$

This approximation is known as the outer-product approximation. As this term only involves first derivatives of the outputs to the weights, which were already derived in equations (A.4), (A.5) and (A.10), the evaluation is much easier and faster than the exact procedure.

Bibliography

Addison, P. S. and Low, D. J. (1998). A novel nonlinear car-following model. *Chaos*, 8:791–799.

Alecsandru, C. (2003). *A hybrid model-based and memory-based short-term traffic prediction system*. Ph.D. thesis, Louisiana State University and Agricultural and Mechanical College.

Atkeson, C. G., Moore, A. W., and Schaal, S. (1997). Locally weighted regression. *Artifical Intelligence Review*, 11:11–73.

Aw, A. and Rascle, M. (2000). Resurrectioonf secondordermodelsof trafficflow. *SIAM Journal on Applied Mathematics*, 60(3):916–938.

Aycin, M. F. and Benekohal, R. F. (1999). Comparison of car-following models for simulation. *Transportation Research Record: Journal of the Transportation Research Board*, 1678:116–127.

Baesens, B., Viaene, S., van den Poel, D., Vanthienen, J., and Dedene, G. (2002). Bayesian neural network learning for repeat purchase modelling in direct marketing. *European Journal of Operational Research*, 138:191–211.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., and Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51:1035–1042.

Barceló, J., Delgado, M., Funes, G., Garciá, D., and Torday, A. (2007). On-line microscopic traffic simulation to support real time traffic management strategies. In *6th ITS European Congress*. Aalborg, Denmark.

Bell, M. G. H. (1991). The estimation of origin-destination matrices by constrained generalised least squares. *Transportation Research Part B: Methodological*, 25(1):13–22.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. N., and Mishalani, R. (2001). Network state estimation and prediction for real-time traffic management. *Networks and spatial economics*, 1:293–318.

Bexelius, S. (1968). An extended model for car-following. *Transportation Research*, 2:13–21.

Bishop, C. M. (1992). Exact calculation of the hessian matrix for the multilayer perceptron. *Neural Computation*, 4:494–501.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, 24:377–380.

Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann.

Brackstone, M. and McDonald, M. (1999). Car-Following: A Historical Review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2:181 – 186.

Brockfeld, E., Kuhne, R. D., Skabardonis, A., and Wagner, P. (2003). Towards benchmarking of microscopic traffic flow models. *Transportation Research Record: Journal of the Transportation Research Board*, 1852:124–129.

Brockfeld, E., Kuhne, R. D., and Wagner, P. (2004). Calibration and validation of microscopic traffic flow models. *Transportation Research Record: Journal of the Transportation Research Board*, 1876:62–70.

Casella, G. and Berger, R. L. (1990). *Statistical Inference*. Duxbury Press, Belmont, California, USA.

Chakroborty, P. and Kikuchi, S. (1999). Evaluation of the general motors based car-following models and a proposed fuzzy inference model. *Transportation Research Part C: Emerging Technologies*, 7:209–235.

Chandler, R. E., Herman, R., and Montroll, E. W. (1958). Traffic dynamics: studies in car following. *Operations Research*, 6:165–184.

Chen, C., Kowon, J., Rice, J., Skabardonis, A., and Varaiya, P. (2003). Detecting errors and imputing missing data for single-loop surveillance systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1855:160–167.

Chen, Z. (2003). Bayesian filtering: from kalman filters to particle filters, and beyond. Technical report, Adaptive Syst. Lab., McMaster University, Hamilton, ON, Canada.

Chua, C. G. and Goh, A. T. C. (2003). A hybrid bayesian back-propagation neural network approach to multivariate modelling. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27:651–667.

Clark, S. (2003). Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering*, 129:161–168.

Courant, R., Friedrichs, K., and Lewy, H. (1928). Uber die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100:32–74.

Daganzo, C. F. (1995a). The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93.

Daganzo, C. F. (1995b). Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B: Methodological*, 29:277–286.

de Cervántes Saavedra, M. (1615). *El ingenioso hidalgo Don Quixote de la Mancha*.

Dharia, A. and Adeli, H. (2003). Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, 16:607–613.

Dia, H. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, 131:253–261.

Dougherty, M. S. and Cobbett, M. R. (1997). Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting*, 13:21–31.

Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15:105–111.

Golias, I. and Karlaftis, M. G. (2001). An international comparative study of self-reported driver behavior. *Transportation Research Part F: Traffic Psychology and Behaviour*, 4(4):243–256.

Gosh, D. and Knapp, C. H. (1978). Estimation of traffic variables using a linear model of traffic flow. *Transportation Research*, 12:395–402.

Greene, W. H. (2000). *Econometric Analysis*. Prentice-Hall, Upper Saddle River, NJ, USA.

Greenshields, B. D. (1934). A Study of Traffic Capacity. *Proceedings Highway Research Board*, 14:448–477.

Gull, S. F. (1989). *Developments in Maximum Entropy Data Analysis*. Kluwer, Dordrecht, The Netherlands.

Hamdar, S. H., Treiber, M., Mahmassani, H. S., and Kesting, A. (2008). Modeling driver behavior as a sequential risk taking task. *Transportation Research Record: Journal of the Transportation Research Board*, 2088:208–217.

Hassibi, B. and Stork, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems*. Morgan Kaufmann, San Mateo, CA, USA.

Haykin, S. (2001). *Kalman Filtering and Neural Networks*. Wiley, Hamilton, Ontario, Canada.

Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks*. Washington, DC, USA.

Hegyi, A., de Schutter, B., and Hellendoorn, H. (2005). Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13:185–209.

Hegyi, A., Girimonte, D., Babuska, R., and de Schutter, B. (2006). A comparison of filter configurations for freeway traffic state estimation. In *IEEE Intelligent Transportation Systems Conference*. Toronto, Canada.

Helbing, D. (1996). Gas-kinetic derivation of navier-stokes-like traffic equations. *Physical Review E*, 53:2266–2381.

Helly, W. (1959). Simulation of bottlenecks in single lane traffic flow. In *International symposium on the theory of traffic flow*. New York, NY, USA.

Ho, Y. C. and Lee, R. C. K. (1964). A bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9:333–339.

Hoogendoorn, S., Ossen, S. J. L., and van Lint, J. W. C. (2007a). Advanced calibration of car-following models. In *11th world conference on transport research*. Berkely, CA, USA.

Hoogendoorn, S. P. and Bovy, P. H. L. (2001). State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part 1: Journal of Systems and Control Engineering*, 215:283–303.

Hoogendoorn, S. P. and Ossen, S. J. L. (2005). Parameter estimation and analysis of car-following models. In *16th international symposium on transportation and traffic theory*. College Park, MD, USA.

Hoogendoorn, S. P., Ossen, S. J. L., and Schreuder, M. (2006). Empirics of multianticipative car-following behavior. *Transportation Research Record: Journal of the Transportation Research Board*, 1965:112–120.

Hoogendoorn, S. P., Ossen, S. J. L., and Schreuder, M. (2007b). Properties of a microscopic heterogeneous multi-anticipative traffic flow models. In Allsop, R. E., Bell, M. G. H., and Heydecker, B. G., editors, *Transportation and Traffic Theory 2007*. Elsevier, Amsterdam, the Netherlands.

Hoogendoorn, S. P., van Zuylen, H. J., Schreuder, M., Gorte, B. G. H., and Vosselman, G. (2003). Microscopic traffic data collection by remote sensing. *Transportation Research Record: Journal of the Transportation Research Board*, 1855:121–128.

Huisken, G. and van Maarseveen, M. (2000). Congestion prediction on motorways: a comparative analysis. In *Proceedings of the 7th World Congress on ITS*.

Innamaa, S. (2005). Short-term prediction of travel time using neural networks on an interurban highway. *Transportation*, 32:649–669.

Ishak, S., Kotha, P., and Alecsandru, C. (2003). Optimization of dynamic neural network performance for short-term traffic prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 1836:45–56.

Jazwinsky, A. H. (1970). *Stochastic Process and Filtering Theory*. Academic Press, New York, NY, USA.

Johansson, E. M., Dowla, F. U., and Goodman, D. M. (1991). Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, 2:291–301.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 (series D):35–45.

Kantowitz, B. H., Hanowski, R. J., and Kantowitz, S. C. (1997). Driver acceptance of unreliable traffic information in familiar and unfamiliar settings. *Human Factors*, 39:164–176.

Kesting, A. and Treiber, M. (2008). Calibrating car-following models by using trajectory data. *Transportation Research Record: Journal of the Transportation Research Board*, 2088:148–156.

Krogh, A. and Herts, J. A. (1995). A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, 4:950–957.

Kuchipudi, C. M. and Chien, S. I. J. (2003). Development of a hybrid model for dynamic travel-time prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 1855:22–31.

Lebacque, J. P. (1996). The godunov scheme and what it means for first order traffic flow models. In Lesort, J. B., editor, *Proceedings of the 13th International Symposium of Transportation and Traffic Theory*, pages 647–677. Lyon, France.

Leclerq, L., Laval, J., and Chevallier, E. (2007). The lagrangian coordinates and what it means for first order traffic flow models. In *Proceedings of the 17th International Symposium on Transportation and Traffic Theory*.

Lee, S. and Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, 1678:179–188.

Lee, S., Kim, D., Kim, J., and Cho, B. (1998). Comparison of models for predicting short-term travel speeds. In *Proceedings of the 5th World Congress on ITS*.

Lenz, H., Wagner, C. K., and Sollacher, R. (1999). Multi-anticipative car-following model. *The European Physical Journal B*, 7:331–335.

Lighthill, M. J. and Whitham, G. B. (1955). On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345.

Lisboa, P. J. G., Wong, H., Harris, P., and Swindell, R. (2003). Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. *Artificial Intelligence in Medicine*, 28:1–25.

Liu, H., van Zuylen, H. J., Chen, Y., and Zhang, K. (2005). Prediction of urban travel times with intersection delays. In *Proceedings of the 8th international IEEE conference on Intelligent Transportation Systems.*. Vienna, Austria.

Mackay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation*, 4:415–447.

Mackay, D. J. C. (1992b). A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472.

Mackay, D. J. C. (1994). Bayesian non-linear modelling for the prediction competition. *ASHRAE Transactions*, 100:1053–1062.

Mackay, D. J. C. (1995). Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505.

Mackay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithm*. Cambridge University Press, Cambridge, UK.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and spatial economics*, 1:267–292.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*. Academic Press, New York, NY, USA.

Minka, T. P. (2001). Automatic choice of dimensionality for pca. *Advances in Neural Information Processing Systems*, 13:598–604.

Møller, M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533.

Nagel, K., Wagner, P., and Woesler, R. (2003). Still flowing: Approaches to traffic flow and traffic jam modeling. *Operations Research*, 51(5):681–710.

Nihan, N. L. (1980). Use of the box and jenkins time series technique in traffic forecasting. *Transportation*, 9:125–143.

Nikovski, D., Nishiuma, N., Goto, Y., and Kumazawa, H. (2005). Univariate short-term prediction of road travel times. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 1074–1079. Vienna, Austria.

Okutani, I. and Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research Part B: Methodological*, 18:1–11.

Ossen, S. J. L. (2008). *Longitudinal Driving Behavior: Theory and Empirics*. Trail thesis series, Delft University of Technology.

Ossen, S. J. L. and Hoogendoorn, S. P. (2005). Car-following behavior analysis from microscopic trajectory data. *Transportation Research Record: Journal of the Transportation Research Board*, 1934:13–21.

Ossen, S. J. L., Hoogendoorn, S. P., and Gorte, B. G. H. (2006). Interdriver differences in car-following: a vehicle trajectory-based study. *Transportation Research Record: Journal of the Transportation Research Board*, 1965:121–129.

Özkan, T., Lajunen, T., Chliaoutakis, J. E., Parker, D., and Summala, H. (2006). Cross-cultural differences in driving behaviours: A comparison of six countries. *Transportation Research Part F: Traffic Psychology and Behaviour*, 9(3):227–242.

Panway, S. and Dia, H. (2005). Comparative evaluation of microscopic car-following behavior. *IEEE Transactions on Intelligens Transportation Systems*, 6:314–325.

Payne, H. J. (1971). Models of freeway traffic and control. *Mathematical Models of Public Systems, Simulation Council Proceedings*, 28:51–61.

Penny, W. D. Roberts, S. J. (1999). Bayesian neural networks for classification: how useful is the evidence framework? *Neural Networks*, 12(6):877–892.

Perrone, M. P. (1994). General averaging results for convex optimization. In *Connectionist Models Summer School*. Hillsdale, NJ, USA.

Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, S., Kiartzis, S., Panagiotou, H., and Maslaris, N. (2001). A bayesian multiple models combination method for time series prediction. *Journal of intelligent and robotic systems*, 31:69–89.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.

Pucher, J. (1988). Urban travel behavior as the outcome of public policy: The example of modal-split in western europe and north america. *Journal of the American Planning Association*, 54(4):509–520.

Punzo, V. and Simonelli, F. (2005). Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transportation Research Record: Journal of the Transportation Research Board*, 1934:53–63.

Punzo, V. and Tripodi, A. (2007). Steady-state solutions and multi-class calibration of gipps' microscopic traffic flow model. *Transportation Research Record: Journal of the Transportation Research Board*, 1999:104–114.

Rakha, H. and Crowther, B. (2003). Comparison and calibration of fresim and integration stready-state car-following behavior. *Transportation Research Part A: Policy and Practice*, 37:1–27.

Ran, B. (2000). Using traffic prediction models for providing predictive traveller information. *International Journal of Technology Management*, 20(3/4):326–339.

Ranjitkar, P., Nakatsuji, T., and Asano, M. (2004). Performance evaluation of microscopic traffic flow models with test track data. *Transportation Research Record: Journal of the Transportation Research Board*, 1876:90–100.

Ranjitkar, P., Nakatsuji, T., and Kawamura, A. (2005). Experimantal analysis of car-following dynamics and traffic stability. *Transportation Research Record: Journal of the Transportation Research Board*, 1934:22–32.

Rascle, M. (2002). An improved macroscopic model of traffic flow: derivation and links with the lighthill witham model. *Mathematical and Computer Modeling*, 35:581–590.

Rice, J. and van Zwet, E. (2004). A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207.

Richards, P. I. (1956). Shock waves on the highway. *Operations Research*, 4:4251.

Robinson, A. R. and Lermusiaux, P. F. J. (2001). *Data Assimilation in Models*. Academic Press Ltd., London.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning Internal Representations by Error Propagation*. MIT Press, Cambridge, MA, USA.

Schreiter, T., van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and Hoogendoorn, S. P. (2010). Data - model synchronization in extended kalman filters for accurate online traffic state estimation. In *Traffic Flow Theory and Characteristics Committee - Summer Meeting of the Transportation Research Board*. Annecy, France.

Sivia, D. S. (1996). *Data Analysis: a Bayesian Tutorial*. Oxford University Press, New York, NY, USA.

Smets, P. (1993). What is dempster-shafer's model? Technical report, IRIDIA, Iniversité Libre de Bruxelles.

Smith, B. L. and Demetsky, M. J. (1996). Multiple-interval freeway traffic flow forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, 1554:136–141.

Smith, B. L. and Demetsky, M. J. (1997). Traffic flow forecasting: comparison of modeling approaches. *Journal of Transportation Engineering*, 123(4):261–266.

Smulders, S. A. (1990). Control of freeway traffic flow by variable speed signs. *Transportation Research Part B: Methodological*, 24(2):111–132.

St-Pierre, M. and Gingras, D. (2004). Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. In *presented at the IEEE Intelligent Vehicles Symposition*. Parma, Italy.

Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356.

Sun, H., Liu, H. X., Xiao, H., He, R. R., and Ran, B. (2003). Use of local linear regression model for short-term traffic forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, 1836:143–150.

Sun, X., Munoz, L., and Horowitz, R. (2004). Mixture kalman filter based highway congestion mode and vehicle density estimator and its application. In *American Control Conference*. Boston, MA, USA.

Tampère, C. M. J. (2004). *Human-kinetic multiclass traffic flow theory and modelling*. Phd thesis, Delft University of Technology.

Tampère, C. M. J. and Immers, L. H. (2007). An extended kalman filter application for traffic state estimation using ctm with implicit mode switching and dynamic parameters. In *Proceedings of the 10th Intelligent Transportation Systems Conference*. Seattle, WA, USA.

Thodberg, H. H. (1993). Ace of bayes: Application of neural networks with pruning. Technical report, Roskilde, The Danish Meat Research Institute.

Treiber, M. and Helbing, D. (2002). Reconstructing the Spatio-Temporal Traffc Dynamics from Stationary Detector Data. *Cooper@tive Tr@nsport@tion Dyn@mics*, 1:3.1–3.24.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824.

UJMP (2010). http://www.ujmp.org.

van Hinsbergen, C. P. I., Hegyi, A., van Lint, J. W. C., and van Zuylen, H. J. (2009a). Application of bayesian trained neural networks to predict stochastic travel times in urban networks. In *16 World Congress on Intelligent Transport Systems*. Stockholm, Sweden.

van Hinsbergen, C. P. I., Hegyi, A., van Lint, J. W. C., and van Zuylen, H. J. (2010a). Bayesian neural networks for prediction of stochastic travel times in urban networks. Submitted for publication in IET Intelligent Transport Systems.

van Hinsbergen, C. P. I., Schreiter, T., van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2010b). Online estimation of kalman filter parameters for traffic state estimation. In *Proceedings of the Seventh Triennial Symposium on Transportation Analysis (TRISTAN VII)*. Tromso, Norway.

van Hinsbergen, C. P. I., Schreiter, T., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2010c). Fast traffic state estimation with the localized extended kalman filter. In *13th International IEEE Conference on Intelligent Transportation Systems*. Madeira Island, Portugal.

van Hinsbergen, C. P. I., Schreiter, T., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2010d). The localized extended kalman filter for scalable, real-time traffic state estimation. Submitted for publication in IEEE Transactions on Intelligent Transportation Systems.

van Hinsbergen, C. P. I., Tampère, C. M. J., van Lint, J. W. C., and van Zuylen, H. J. (2009b). Urban intersections in first order models with the godunov scheme. In *mobil.TUM - international scientific conference on mobility and transport*. Munich, Germany.

van Hinsbergen, C. P. I., Tampère, C. M. J., van Lint, J. W. C., and van Zuylen, H. J. (2010e). Urban intersections in the first order models. Submitted for publication in Transportation Research Part C: Emerging Technologies.

van Hinsbergen, C. P. I., van Lint, J. W. C., Hoogendoorn, S., and van Zuylen, H. J. (2010f). A unified framework for calibration and comparison of car-following models. Submitted for publication to Transportmetrica.

van Hinsbergen, C. P. I., van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2009c). Bayesian calibration of car-following models. In *12th IFAC Symposium on Control in Transportation Systems (CTS'09)*. Redondo Beach, California, USA.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008a). Bayesian combination of travel time prediction models. *Transportation Research Record: Journal of the Transportation Research Board*, 2064:73–80.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008b). Bayesian combination of travel time prediction models. In *87th meeting of the Transportation Research Board*.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008c). Bayesian trained neural networks to forecast travel times. In *Proceedings of 10th TRAIL Congress and Knowledge Market*. Rotterdam, the Netherlands.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008d). Neural network committee to predict travel times: comparison of bayesian evidence approach to the use of a validation set. In *11th international IEEE conference on intelligent transportation systems*. Beijing, China.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009d). Bayesian committee of neural networks to predict travel times with confidence intervals. *Transportation Research Part C: Emerging Technologies*, 17:498–509.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009e). Bayesian training and committees of state space neural networks for online travel time prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 2105:118–126.

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009f). Bayesian training and committees of state space neural networks for online travel time prediction. In *88th Annual Meeting of the Transportation Research Board*.

van Hinsbergen, C. P. I., van Lint, J. W. C., van Zuylen, H. J., and Sanders, F. M. (2007). Short Term Traffic Prediction Models. In *14th World Congress on Intelligent Transport Systems*. Beijing, China.

van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2008e). Macroscopic modelling of intersection delay with linearly decreasing turn capacities. In *Proceedings of the International Symposium on Dynamic Traffic Assignment*. Leuven, Belgium.

van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2008f). Using an lwr model with a cell based extended kalman filter to estimate travel times. In *Proceedings of the 3rd International Symposium of Transport Simulation*. Surfer's Paradise, QLD, Australia.

van Lint, J. W. C. (2004). *Reliable travel time prediction for freeways*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands.

van Lint, J. W. C. (2008). Online learning solutions for freeway travel time prediction. *IEEE Transactions on Intelligens Transportation Systems*, 9:38–47.

van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2002). Freeway travel time prediction with state-space neural networks - modeling state-space dynamics with recurrent neural networks. *Transportation Research Record: Journal of the Transportation Research Board*, 1811:30–39.

van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2005). Accurate travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13:347–369.

van Lint, J. W. C. and van der Zijpp, N. J. (2003). Improving a travel-time estimation algorithm by using dual loop detectors. *Transportation Research Record: Journal of the Transportation Research Board*, 1855:41–48.

van Wageningen-Kessels, F. L. M., van Lint, J. W. C., Hoogendoorn, S. P., and Vuik, C. (2009). Implicit and explicit numerical methods for macroscopic traffic flow models. In *88th Annual Meeting of the Transportation Research Board*. Washington, DC, USA.

Vlahogianni, E. I., Golias, J. C., and Karlaftis, M. G. (2004). Short-term traffic forecasting: overview of objectives and methods. *Transport Reviews*, 24(5):533–557.

Wang, Y. and Papageorgiou, M. (2005). Real-time freeway state estimation based on extended kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39:167–181.

Wang, Y., Papageorgiou, M., and Messmer, A. (2007). Real-time freeway traffic state estimation based on extended kalman filter: a case study. *Transportation Science*, 41:167–181.

Williams, P. M. (1991). A marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients. Technical Report Technical Report CSRP-229, University of Sussex.

Wu, J., Brackstone, M., and McDonald, M. (2003). The validation of a microscopic simulation model: a methodological case study. *Transportation Research Part C: Emerging Technologies*, 11:463–479.

Yun, S. Y., Namkoong, S., Rho, J. H., Shin, S. W., and Choi, J. U. (1998). A performance evaluation of neural network models in traffic volume forecasting. *Mathematical and Computer Modelling*, 27:293–310.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28.

Zhang, H. M. (2000). Recursive prediction of traffic conditions with neural network models. *Journal of Transportation Engineering*, 126:472–481.

Zhang, H. M. (2002). A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B: Methodological*, 36(3):275–290.

Zheng, W., Lee, D., and Shi, Q. (2006). Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of Transportation Engineering*, 132:114–121.

Zhong, M., Sharma, S., and Lingras, P. (2005). Refining genetically designed models for improved traffic prediction on rural roads. *Transportation Planning and Technology*, 28:213–236.

Zuurbier, F. S. (2010). *Intelligent Route Guidance*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands.

Zuurbier, F. S., van Lint, J. W. C., and Knoop, V. L. (2006). State estimation using an extended kalman filter and the first order traffic flow model dsmart. In *Eleventh IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands.

Summary

Bayesian Data Assimilation for Improved Modeling of Road Traffic

This thesis deals with the optimal use of existing models that predict certain phenomena of the road traffic system. Such models are extensively used in Advanced Traffic Information Systems (ATIS), Dynamic Traffic Management (DTM) or Model Predictive Control (MPC) approaches in order to improve the traffic system. As road traffic is the result of human behavior which is ever changing and which varies internationally, for each of these phenomena a multitude of models exist. The scientific literature generally is not conclusive about which of these models should be preferred. One common problem in road traffic science is therefore that for each application a choice has to be made from a set of available models. A second task that always needs to be performed is the calibration of the parameters of the models. A third and last task is the application of the chosen and calibrated model(s) to predict a part of the traffic system.

For each of these three steps, generally *data* (measurements of the traffic system) is required. In this thesis, all three uses of data are summarized into *data assimilation*, which is defined as "*the use of techniques aimed at the treatment of data in coherence with models in order to construct an as accurate and consistent picture of reality as possible. It comprises the use of data for model validation and identification (choosing between models), model calibration and estimation and prediction and specifically deals with the interactions between all these tasks*". In this thesis, a Bayesian framework is used in which these interactions can be treated consistently: solving one of these steps automatically leads to the solution of the other steps. Throughout the thesis, the calibration task is always performed first using standard optimization techniques such as regression or gradient-based algorithms. Once all available models are calibrated, a choice can be made between them. The selected model(s) can then be used to make an as accurate prediction as possible.

One very important feature of the Bayesian framework is that it takes the complexity of models into account in the model comparison step. More complex models generally show a lower calibration error than more simple models, but they do not necessarily make better predictions. This is known as the problem of overfitting. The Bayesian frame-

work deals with overfitting by penalizing models which contain more parameters and are thus more complex. The Bayesian assessment of models produces a measure called the *evidence*, which balances between a goodness of fit to the calibration data set and the complexity of the model. Besides this, the framework has more benefits. First, prior information can easily be included in each step of data assimilation. Second, error bars can be constructed on the predictions. This may be beneficial to the performance or public acceptance of ATIS, DTM or MPC systems. Third, a *committee* can be constructed, in which predictions of multiple models are combined. Committees generally produce more accurate predictions than individual models.

The Bayesian framework for data assimilation is applied to three different phenomena: (1) car-following modeling, (2) travel time prediction and (3) traffic state estimation using a first order traffic flow model (the LWR model) and an Extended Kalman Filter. Finally, a part of the research is devoted to speeding up the EKF such that it can be applied together with the LWR model in real time to large networks.

Car-following behavior

Recent research has revealed that there exists large heterogeneity in car-following behavior such that different car-following models best describe different drivers behavior. The choice of a car-following model thus has to be made for each individual driver. Current approaches to calibrate and compare different models for one driver do not take the complexity of the model into account or are only able to compare a specific set of models. Using the Bayesian framework for data assimilation the suitability of any set of models can be quantitatively assessed for each single driver. In this research the Bayesian framework for data assimilation is applied to two simple car-following models, the CHM model and the Helly model. The workings of the Bayesian framework are demonstrated in a real-world experiment using 229 trajectories of drivers who were in car-following mode. Aggregated over all drivers, the probabilities of each model relative to the probability of all used models can be computed. This can serve as input to a heterogeneous microscopic simulation of traffic. The outcomes of this experiment show that averaged over all drivers the CHM model has a probability of 31% and the Helly model of 69%.

Travel time prediction

In this research different types of models are applied to the problem of travel time prediction: linear regression models and neural networks. Three experiments are performed on an 8.5 km long stretch of the A12 motorway in the Netherlands. Travel time data was collected during a period of three months in early 2007. In every experiment the Bayesian framework is applied to calibrate a set of available models, to make choices between models and to make predictions of the travel times. In all experiments a *committee* is used.

In the first experiment two linear regression models are used. In this experiment the framework is applied dynamically: each time step, the available measured travel times and a set of historic loop detector data are used to recalibrate the models using standard regression tools. After this regression (calibration) is finished, the evidence measure assigns a preference for one of the two models over the other. Two strategies are tested: (1) the prediction of the model with the highest evidence is used and (2) the weighted average of the predictions of both models is used, where the evidence is used as a weight factor. The results show that both models perform similarly well, and that the committees show a slight improvement of accuracy. A clear difference between the two strategies was not found.

In the second experiment feed forward neural networks are used, with one hidden layer with different numbers of hidden nodes. The Bayesian framework is used to train (calibrate) 84 different neural networks, and the evidence measure is used to select high-potential networks. Using a separate validation data set, the evidence is tested as a predictor of the true prediction error. It is found that there is a correlation between the two, but that the evidence is not a perfect predictor of a well-performing neural network due to several reasons: (1) the size of the data sets may be too small so that the validation error does not equal the true error, (2) the models that are used may require improvement, such as weight pruning and (3) several assumptions were made in order to solve the necessary equations, such as the assumption that all distributions are Gaussian. In the same experiment a committee was tested using a simple average of the outcomes of a selection of models, ranked on the evidence. It was found that the average prediction error decreased from 8.1% of the best individual neural network to 7.8% for the committee. Finally, in the experiment the construction of error bars was tested, and it was shown that 97.4% of the true travel time fell within the 95% prediction intervals. The discrepancy between the two can be attributed to the relative simplicity of the used neural networks.

In the third and final experiment feed forward neural networks (FFNN) as well as state-space neural networks (SSNN, a specific type of a recurrent or Elman neural network) were applied. The SSNN generally contains more parameters than the FFNN, but potentially are more accurate because they can take time dependencies into account: a typical problem of the necessity of balancing complexity against the ability to fit to a data set. For the Bayesian framework to be applied, the Jacobian and Hessian of the SSNN were derived (see Appendix A). Then, the Bayesian framework could again be used to compute the evidence for each model. In the experiments 70 FFNN and 70 SSNN were trained. The evidence was then used to form a committee of neural networks to predict the travel time on the selected motorway. The results show that the FFNN perform better on a short prediction horizon (5 minutes ahead), while the SSNN perform better on a longer horizon (15 minutes). The results also show that the use of a committee improves the accuracy of the predictions. In this experiment the calibration error was found to be a better predictor of the true error than the evidence. Nevertheless, the experiments show

nearly no difference in performance of committees ranked on the evidence or ranked on the calibration error.

The first order model with an Extended Kalman Filter

In this research, two studies are performed on the application of a first order model (the LWR model) in combination with an Extended Kalman Filter (EKF) to create a network-wide estimate of the traffic state. The first study deals with the fact that the EKF itself contains parameters that require calibration. Using the Bayesian framework that has also been applied to calibrate car-following models and travel time prediction models, a method to calibrate the parameters of the EKF is derived. Using this result, the EKF parameters can be dynamically adapted during simulation. In an experiment on a small network it is then shown that the dynamic Bayesian choice for parameters leads to nearly the same accuracy compared to the optimal choice of fixed parameter values. This result is especially useful in large-scale applications, where it is impossible to test all possible fixed parameter values of the EKF.

Finally, the last study overcomes a large disadvantage of the EKF: it is too slow to perform in real-time on large networks. To overcome this problem the novel Localized EKF (L-EKF) is proposed. The logic of the traffic network is used to correct only the state in the vicinity of a detector. The L-EKF does not use all information available to correct the state of the network; the resulting accuracy is however equal in case the radius of the local filters is taken sufficiently large. In two experiments, one on synthetic data and one on real-world data, it is shown that the L-EKF is much faster than the traditional Global EKF (G-EKF), that it scales much better with the network size and that it leads to estimates with the same accuracy as the G-EKF, even if the spacing between detectors is up to 5 kilometers. Opposed to the G-EKF, the L-EKF is hence a highly scalable solution to the state estimation problem.

Samenvatting

Nederlandse vertaling van Bayesian Data Assimilation for Improved Modeling of Road Traffic

Dit proefschrift gaat over het optimaal inzetten van bestaande modellen die gebruikt worden om bepaalde verschijnselen van het wegverkeerssysteem te voorspellen. Dergelijke modellen worden uitvoerig gebruikt om het verkeerssysteem te verbeteren, bijvoorbeeld in geavanceerde verkeersinformatiesystemen, dynamisch verkeersmanagementsystemen of modelvoorspelde regelingssystemen. Voor de beschrijving van ieder onderdeel van het verkeerssysteem bestaan er in de regel verschillende modellen, omdat verkeer het resultaat is van menselijk gedrag dat altijd aan verandering onderhevig is en bovendien sterk varieert van land tot land. De wetenschappelijke literatuur is in het algemeen niet eenduidig over welk van deze modellen gebruikt zou moeten worden. Een algemeen probleem binnen de verkeerskunde is daarom dat voor iedere toepassing een keuze gemaakt moet worden uit een set van beschikbare modellen. Een tweede probleem is dat de parameters van deze modellen gekalibreerd moeten worden. Een derde en laatste taak is het toepassen van de gekozen en gekalibreerde modellen om een voorspelling te maken van een deel van het verkeerssysteem.

Voor elk van deze drie stappen is in het algemeen *data* (metingen van het verkeerssysteem) nodig. In deze dissertatie zijn alle drie de gebieden waarin data wordt gebruikt samengevat als *data assimilatie*, dat gedefinieerd is als "*het gebruik van technieken om data in samenspel met modellen in te zetten voor een zo nauwkeurig en consistent mogelijke reconstructie van de werkelijkheid. Het behelst het gebruik van data om modellen te valideren en te identificeren (het kiezen tussen modellen), modellen te kalibreren en om schattingen en voorspellingen te maken en het gaat expliciet om met interacties tussen deze stappen*". In dit proefschrift is een Bayesiaans raamwerk gebruikt waarin op een consistente wijze met de interacties tussen elk van de drie stappen wordt omgegaan: het oplossen van één van de drie problemen leidt automatisch tot het oplossen van de andere twee. In het hele proefschrift vindt telkens eerst de kalibratie plaats, gebruik makend van standaard optimalisatietechnieken zoals regressie en gradient-gebaseerde algoritmes. Nadat alle beschikbare modellen zijn gekalibreerd kan vervolgens daartussen een keuze

worden gemaakt. Het gekozen model kan of de gekozen modellen kunnen vervolgens worden gebruikt om een zo nauwkeurig mogelijke voorspelling te maken.

Een zeer belangrijk kenmerk van het Bayesiaanse raamwerk is dat het rekening houdt met de complexiteit van modellen in de vergelijkingsstap. Meer ingewikkelde modellen hebben in het algemeen een lagere fout na afloop van de kalibratie dan meer eenvoudige modellen, maar zij maken niet noodzakelijkerwijs betere voorspellingen. Dit is bekend als het probleem van 'overfitten'. Het Bayesiaanse raamwerk gaat om met overfitten door modellen te straffen die veel parameters bevatten en dus meer complex zijn. Bij het Bayesiaans vergelijken van modellen wordt de 'bewijsmaat' gebruikt, een maatstaf die de complexiteit van een model balanceert met de laagte van de fout tijdens de kalibratie. Daarnaast levert het gebruik van het raamwerk een aantal andere voordelen op. Ten eerste kan voorinformatie gemakkelijk in iedere stap van de data assimilatie worden verwerkt. Ten tweede kunnen betrouwbaarheidsintervallen worden berekend bij iedere voorspelling. Dit kan belangrijk zijn voor het presteren en de publieke acceptatie van verkeersinformatiesystemen, dynamisch verkeersmanagementsystemen of modelvoorspelde regelingssystemen. Ten derde kan met behulp van het raamwerk een *comité* worden geconstrueerd, waarin voorspellingen van meerdere modellen worden gecombineerd. Comités leveren in het algemeen meer nauwkeurige voorspellingen dan individuele modellen.

Het Bayesiaanse raamwerk voor data assimilatie is in dit proefschrift toegepast op drie verschillende onderdelen van het verkeerssysteem: (1) voertuigvolgmodellen, (2) reistijdvoorspelling en (3) toestandschatten met een eerste orde model en een Extended Kalman Filter (EKF). Tot slot is een deel van het onderzoek gewijd aan het versnellen van het EKF opdat het real time toegepast kan worden in combinatie met het eerste orde model op grote verkeersnetwerken.

Voertuigvolgmodellen

Recent onderzoek heeft aangetoond dat er grote heterogeniteit bestaat in voertuigvolggedrag, zodat verschillende modellen het beste het gedrag van verschillende bestuurders beschrijven. De keuze voor een voertuigvolgmodel moet daarom per individu gemaakt worden. Bestaande aanpakken om verschillende modellen te kalibreren en te vergelijken voor een bestuurder houden geen rekening met de complexiteit van de modellen, of zijn alleen in staat om met een specifieke set aan modellen om te gaan. Het Bayesiaanse raamwerk kan gebruikt worden om de geschiktheid van alle soorten modellen te kwantificeren voor iedere individuele bestuurder. In dit onderzoek is het raamwerk toegepast op twee eenvoudige voertuigvolgmodellen: het CHM-model en het Helly-model. De werking van het Bayesiaanse raamwerk is gedemonstreerd in een experiment met 229 werkelijk gemeten trajectorieën van bestuurders die hun voorganger volgden. Geaggregeerd over alle bestuurders kan voor ieder model de waarschijnlijkheid worden berekend relatief aan de waarschijnlijkheid van alle gebruikte modellen. Dit kan dienen als invoer van een

heterogene microsimulatie van verkeer. De resultaten van dit experiment laten zien dat gemiddeld over alle bestuurders de waarschijnlijkheid van het CHM model 31% is en van het Helly model 69%.

Reistijdvoorspelling

In dit onderzoek zijn verschillende soorten modellen ingezet om reistijden te voorspellen: lineaire regressiemodellen en neurale netwerken. Er zijn drie experimenten uitgevoerd op een 8,5 km lang stuk van de A12 tussen Zoetermeer en Voorburg. Op dat stuk snelweg zijn reistijden gemeten gedurende een periode van drie maanden in 2007. In ieder experiment is het Bayesiaanse raamwerk gebruikt om meerdere modellen te kalibreren, om keuzes te maken tussen de modellen en om voorspellingen van de reistijd te maken. In elk experiment is ook een comité ingezet.

In het eerste experiment zijn twee lineaire regressiemodellen gebruikt. In dit experiment is het raamwerk dynamisch toegepast: iedere tijdstap zijn alle voorgaande gemeten reistijden en een set van historische lusdetectordata gebruikt om beide modellen opnieuw te kalibreren door middel van standaard regressietechnieken. Nadat de regressie (kalibratie) was voltooid kon de bewijsmaat worden berekend om een voorkeur uit te drukken voor één van de twee modellen. Twee strategie en om een comité te vormen zijn vervolgens getest om tot een voorspelling te komen: (1) alleen het model met de hoogste bewijsmaat wordt gebruikt om te voorspellen en (2) de voorspellingen van beide modellen worden gewogen gemiddeld naar rato van de bewijsmaat. Het resultaat toont aan dat beide modellen ongeveer even nauwkeurig voorspellen, en dat het gebruik van een comité de resultaten iets verbetert ten opzichte van de individuele voorspellers. Een duidelijk verschil tussen de twee strategie en is niet gevonden.

In het tweede experiment zijn feed-forward neurale netwerken gebruikt, met één tussenlaag met verschillende aantallen neuronen. Het Bayesiaanse raamwerk is gebruikt om 84 verschillende neurale netwerken te trainen (kalibreren). De bewijsmaat is vervolgens gebruikt om een selectie te maken van kansrijke netwerken. Een aparte validatie-dataset is gebruikt om de bewijsmaat te testen als voorspeller van de werkelijke voorspellingsfout. Het resultaat toont dat er een correlatie bestaat tussen de twee, maar dat de bewijsmaat geen perfect selectiemiddel is van nauwkeurig voorspellende modellen om een aantal redenen: (1) de gebruikte dataset kan te klein zijn om representatief te zijn voor de werkelijke voorspellingsfout, (2) de gebruikte modellen dienen te worden verbeterd, bijvoorbeeld door het verwijderen (‘snoeien’) van parameters en (3) verschillende aannames zijn gemaakt om de benodigde vergelijkingen op te kunnen lossen, zoals de aanname dat alle verdelingen Gauss-verdelingen zijn. In hetzelfde experiment is een comité getest door simpelweg de voorspellingen van een selectie van modellen, gesorteerd op de bewijsmaat, te middelen. De resultaten tonen dat de gemiddelde voorspellingsfout van $8,1\%$ van het beste individuele model daalt naar $7,8\%$ door gebruikt te maken van een comité. Tot

slot zijn in het experiment de betrouwbaarheidsintervallen getest. De resultaten laten zien dat $97,4\%$ van de werkelijke reistijden binnen de 95%-betrouwbaarheidsintervallen liggen. De discrepantie tussen de twee kan worden toegeschreven aan de relatief eenvoudige structuur van de gebruikte neurale netwerken.

In het derde en laatste experiment zijn zowel feed-forward neurale netwerken (FFNN) als state-space neurale netwerken (SSNN, een speciaal soort recurrent of Elman neuraal netwerk) gebruikt. Het SSNN bevat in het algemeen meer parameters dan het FFNN, maar kan potentieel ook nauwkeurigere voorspellingen maken omdat het rekening kan houden met tijdsafhankelijkheden. Dit is daarom een typisch voorbeeld van de noodzaak om het vermogen om een kalibratiedataset te beschrijven te balanceren met de complexiteit van het model. Om het Bayesiaanse raamwerk toe te kunnen passen zijn eerst de Jacobiaan en de Hessiaan van het SSNN afgeleid (zie Appendix A). Daarna is het raamwerk gebruikt om voor ieder netwerk de bewijsmaat te berekenen. In het experiment zijn 70 FFNN's en 70 SSNN's getraind. De bewijsmaat is daarna gebruikt om een comité van neurale netwerken te construeren om de reistijd te voorspellen. De resultaten tonen dat de FFNN beter presteren bij een korte voorspellingshorizon (5 minuten vooruit), terwijl de SSNN beter presteren bij een langere horizon (15 minuten). Ook tonen de resultaten aan dat het gebruik van een comité de nauwkeurigheid van de voorspellingen verbetert. In het experiment is gevonden dat de kalibratiefout in dit geval een betere voorspeller is van de werkelijke fout dan de Bayesiaanse bewijsmaat. Desalniettemin tonen de experimenten nauwelijks verschil in nauwkeurigheid van de comités die gerangschikt zijn op de kalibratiefout in vergelijking met een rangschikking op de bewijsmaat.

Het eerste orde model met een Extended Kalman Filter

In dit onderzoek zijn twee studies verricht naar het toepassen van een eerste orde model (het LWR-model) in combinatie met een Extended Kalman Filter (EKF) om een netwerkbrede schatting te maken van de verkeerstoestand. De eerste studie richt zich op het feit dat het EKF zelf parameters bevat die moeten worden gekalibreerd. Het Bayesiaanse raamwerk dat eerder werd gebruikt voor voertuigvolgmodellen en reistijdvoorspelling is toegepast om een uitdrukking te vinden voor de parameters van het EKF. Gebruik makend van deze uitdrukking worden de parameters van het EKF tijdens de simulatie voortdurend aangepast. In een experiment op een klein netwerk is aangetoond dat de keuze voor de dynamische Bayesiaanse parameterwaarden leidt tot bijna dezelfde nauwkeurigheid in vergelijking met de optimale statische parameterwaarden. Dit resultaat is vooral bruikbaar bij grootschalige toepassingen, waarin het onmogelijk is alle mogelijke statische waarden van de parameters van het EKF te testen.

Tot slot richt de laatste studie zich op een groot nadeel van het EKF: het is te traag om real time toegepast te kunnen worden op grootschalige verkeersnetwerken. Om dit probleem te verhelpen is het nieuwe Lokale EKF (L-EKF) ontwikkeld. De logica van het

verkeersnetwerk wordt gebruikt om alleen de toestand in de nabijheid van een detector te corrigeren. Het L-EKF gebruikt niet alle beschikbare informatie om de toestand in het hele netwerk te corrigeren; de resulterende nauwkeurigheid is echter gelijk in het geval de radius van de lokale filters groot genoeg wordt genomen. In twee experimenten, een op een synthetisch netwerk en een op een grootschalig werkelijk netwerk, is aangetoond dat het L-EKF veel sneller is dan het traditionele Globale EKF (G-EKF), dat het veel gunstiger schaalt met de grootte van het netwerk en dat het leidt tot schattingen met dezelfde nauwkeurigheid als het G-EKF, zelfs als de gemiddelde afstand tussen de detectoren 5 kilometer is. In tegenstelling tot het G-EKF is het L-EKF daarom een zeer schaalbare oplossing voor het schatten van de verkeerstoestand.

Curriculum Vitae

Christopher Philip IJsbrand van Hinsbergen (Chris) was born in Enschede, the Netherlands, 10 February 1981. After graduating cum laude at grammar school he started his study Civil Engineering at Delft University of Technology in 1999. He obtained his Master of Science degree cum laude in 2006 on prediction of the individual chance of collisions for car owners using machine learning techniques. From 2007 to 2010 he was a PhD student at Delft University of Technology's Transport and Planning department of the faculty of Civil Engineering and Geosciences.

Chris is the first author of 7 papers submitted to peer reviewed journals, 3 of which have been accepted while the remaining are still under review. Furthermore, Chris has written 12 papers appearing in conference proceedings with full paper review.

Finally, the author has strong entrepreneurial ambitions. Together with his colleague Frank Zuurbier he has started the spin-off 'Fileradar' (Dutch for 'Queue Radar'). With this spin-off, they hope to be successful in bringing the state-of-the-art of traffic science to practice in predicted traffic information and traffic management products.

The following papers by the author have been accepted by or are still under review at journals:

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008a). Bayesian combination of travel time prediction models. *Transportation Research Record: Journal of the Transportation Research Board*, 2064:73–80

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009d). Bayesian committee of neural networks to predict travel times with confidence intervals. *Transportation Research Part C: Emerging Technologies*, 17:498–509

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009e). Bayesian training and committees of state space neural networks for online travel time prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 2105:118–126

van Hinsbergen, C. P. I., van Lint, J. W. C., Hoogendoorn, S., and van Zuylen, H. J.

(2010f). A unified framework for calibration and comparison of car-following models. Submitted for publication to Transportmetrica

van Hinsbergen, C. P. I., Schreiter, T., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2010d). The localized extended kalman filter for scalable, real-time traffic state estimation. Submitted for publication in IEEE Transactions on Intelligent Transportation Systems

van Hinsbergen, C. P. I., Hegyi, A., van Lint, J. W. C., and van Zuylen, H. J. (2010a). Bayesian neural networks for prediction of stochastic travel times in urban networks. Submitted for publication in IET Intelligent Transport Systems

van Hinsbergen, C. P. I., Tampère, C. M. J., van Lint, J. W. C., and van Zuylen, H. J. (2010e). Urban intersections in the first order models. Submitted for publication in Transportation Research Part C: Emerging Technologies

Furthermore, the author has presented the following papers at international conferences:

van Hinsbergen, C. P. I., van Lint, J. W. C., van Zuylen, H. J., and Sanders, F. M. (2007). Short Term Traffic Prediction Models. In *14th World Congress on Intelligent Transport Systems*. Beijing, China

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008d). Neural network committee to predict travel times: comparison of bayesian evidence approach to the use of a validation set. In *11th international IEEE conference on intelligent transportation systems*. Beijing, China

van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2008f). Using an lwr model with a cell based extended kalman filter to estimate travel times. In *Proceedings of the 3rd International Symposium of Transport Simulation*. Surfer's Paradise, QLD, Australia

van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2008e). Macroscopic modelling of intersection delay with linearly decreasing turn capacities. In *Proceedings of the International Symposium on Dynamic Traffic Assignment*. Leuven, Belgium

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008c). Bayesian trained neural networks to forecast travel times. In *Proceedings of 10th TRAIL Congress and Knowledge Market*. Rotterdam, the Netherlands

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2008b). Bayesian combination of travel time prediction models. In *87th meeting of the Transportation Research Board*

van Hinsbergen, C. P. I., Hegyi, A., van Lint, J. W. C., and van Zuylen, H. J. (2009a). Application of bayesian trained neural networks to predict stochastic travel times in urban networks. In *16 World Congress on Intelligent Transport Systems*. Stockholm, Sweden

van Hinsbergen, C. P. I., Tampère, C. M. J., van Lint, J. W. C., and van Zuylen, H. J. (2009b). Urban intersections in first order models with the godunov scheme. In *mobil.TUM - international scientific conference on mobility and transport*. Munich, Germany

van Hinsbergen, C. P. I., van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2009c). Bayesian calibration of car-following models. In *12th IFAC Symposium on Control in Transportation Systems (CTS'09)*. Redondo Beach, California, USA

van Hinsbergen, C. P. I., van Lint, J. W. C., and van Zuylen, H. J. (2009f). Bayesian training and committees of state space neural networks for online travel time prediction. In *88th Annual Meeting of the Transportation Research Board*

van Hinsbergen, C. P. I., Schreiter, T., van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2010b). Online estimation of kalman filter parameters for traffic state estimation. In *Proceedings of the Seventh Triennial Symposium on Transportation Analysis (TRISTAN VII)*. Tromso, Norway

van Hinsbergen, C. P. I., Schreiter, T., Zuurbier, F. S., van Lint, J. W. C., and van Zuylen, H. J. (2010c). Fast traffic state estimation with the localized extended kalman filter. In *13th International IEEE Conference on Intelligent Transportation Systems*. Madeira Island, Portugal

Finally, the author of this thesis is the co-author of the following paper:

Schreiter, T., van Hinsbergen, C. P. I., Zuurbier, F. S., van Lint, J. W. C., and Hoogendoorn, S. P. (2010). Data - model synchronization in extended kalman filters for accurate online traffic state estimation. In *Traffic Flow Theory and Characteristics Committee - Summer Meeting of the Transportation Research Board*. Annecy, France

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 100 titles see the TRAIL website: www.rsTRAIL.nl. The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Hinsbergen, C.P.IJ. van, Bayesian Data Assimilation for Improved Modeling of Road Traffic, T2010/9, November 2010, TRAIL Thesis Series, the Netherlands

Zuurbier, F.S., Intelligent Route Guidance, T2010/8, November 2010, TRAIL Thesis Series, the Netherlands

Larco Martinelli, J.A., Incorporating Worker-Specific Factors in Operations Management Models, T2010/7, November 2010, TRAIL Thesis Series, the Netherlands

Ham, J.C. van, Zeehavenontwikkeling in Nederland: naar een beter beleidsvormingsproces, T2010/6, August 2010, TRAIL Thesis Series, the Netherlands

Boer, E. de, School Concentration and School Travel, T2010/5, June 2010, TRAIL Thesis Series, the Netherlands

Berg, M. van den, Integrated Control of Mixed Traffic Networks using Model Predictive Control, T2010/4, April 2010, TRAIL Thesis Series, the Netherlands

Top, J. van den, Modelling Risk Control Measures in Railways, T2010/3, April 2010, TRAIL Thesis Series, the Netherlands

Craen, S. de, The X-factor: A longitudinal study of calibration in young novice drivers, T2010/2, March 2010, TRAIL Thesis Series, the Netherlands

Tarau, A.N., Model-based Control for Postal Automation and Baggage Handling, T2010/1, January 2010, TRAIL Thesis Series, the Netherlands

Knoop, V.L., Road Incidents and Network Dynamics: Effects on driving behaviour and traffic congestion, T2009/13, December 2009, TRAIL Thesis Series, the Netherlands

Baskar, L.D., Traffic Control and Management with Intelligent Vehicle Highway Systems, T2009/12, November 2009, TRAIL Thesis Series, the Netherlands

Konings, J.W., Intermodal Barge Transport: Network Design, Nodes and Competitiveness, T2009/11, November 2009, TRAIL Thesis Series, the Netherlands

Kusumaningtyas, I., Mind Your Step: Exploring aspects in the application of long accelerating moving walkways, T2009/10, October 2009, TRAIL Thesis Series, the Netherlands

Gong, Y., Stochastic Modelling and Analysis of Warehouse Operations, T2009/9, September 2009, TRAIL Thesis Series, the Netherlands

Eddia, S., Transport Policy Implementation and Outcomes: the Case of Yaounde in the 1990s, T2009/8, September 2009, TRAIL Thesis Series, the Netherlands

Platz, T.E., The Efficient Integration of Inland Shipping into Continental Intermodal Transport Chains: Measures and decisive factors, T2009/7, August 2009, TRAIL Thesis Series, the Netherlands

Tahmasseby, S., Reliability in Urban Public Transport Network Assessment and Design, T2009/6, June 2009, TRAIL Thesis Series, the Netherlands

Bogers, E.A.I., Traffic Information and Learning in Day-to-day Route Choice, T2009/5, June 2009, TRAIL Thesis Series, the Netherlands

Amelsfort, D.H. van, Behavioural Responses and Network Effects of Time-varying Road Pricing, T2009/4, May 2009, TRAIL Thesis Series, the Netherlands

Li, H., Reliability-based Dynamic Network Design with Stochastic Networks, T2009/3, May 2009, TRAIL Thesis Series, the Netherlands

Stankova, K., On Stackelberg and Inverse Stackelberg Games & their Applications in the Optimal Toll Design Problem, the Energy Markets Liberalization Problem, and in the Theory of Incentives, T2009/2, February 2009, TRAIL Thesis Series, the Netherlands