An application of Bayesian Hidden Markov Models to explore traffic flow conditions in an urban area

Lovisa Andersson
Department of Statistics
Supervisor: Patrik Andersson
This study employs Bayesian Hidden Markov Models as method to explore vehicle traffic flow conditions in an urban area in Stockholm, based on sensor data from separate road positions. Inter-arrival times are used as the observed sequences. These sequences of inter-arrival times are assumed to be generated from the distributions of four different (and hidden) traffic flow states; nightly free flow, free flow, mixture and congestion. The filtered and smoothed probability distributions of the hidden states and the most probable state sequences are obtained by using the forward, forward-backward and Viterbi algorithms. The No-U-Turn sampler is used to sample from the posterior distributions of all unknown parameters. The obtained results show in a satisfactory way that the Hidden Markov Models can detect different traffic flow conditions. Some of the models have problems with divergence, but the obtained results from those models still show satisfactory results. In fact, two of the models that converged seemed to overestimate the presence of congested traffic and all the models that not converged seem to do adequate estimations of the probability of being in a congested state. Since the interest of this study lies in estimating the current traffic flow condition, and not in doing parameter inference, the model choice of Bayesian Hidden Markov Models is satisfactory. Due to the unsupervised nature of the problematization of this study, it is difficult to evaluate the accuracy of the results. However, a model with simulated data and known states was also implemented, which resulted in a high classification accuracy. This indicates that the choice of Hidden Markov Models is a good model choice for estimating traffic flow conditions.

**Keywords:** Bayesian statistics, hidden states, Markov chain, traffic flow modeling, filtering, smoothing, most probable state sequence, MCMC, Hamiltonian Monte Carlo, No-U-Turn sampler
I would like to express my gratitude to IBM for giving me the opportunity to be a part of the project team while writing this master thesis. Special thanks to Holger Hellebro for all the support on the way. Furthermore I would like to thank my supervisor Patrik Andersson for the useful comments and guidance regarding the statistical modeling through the learning process of this master thesis.
# Contents

1 Introduction 1

2 Theory 3

2.1 Traffic flow modeling 3
2.2 Hidden Markov Model 4

2.2.1 Filtering and smoothing 6
2.2.2 Most probable state sequence 8

2.3 Bayesian parameter estimation 8

2.3.1 Hamiltonian Monte Carlo and No-U-Turn sampler 10
2.3.2 MCMC convergence and model diagnostics 13

3 Methodology 15

3.1 Data overview 15
3.2 Hidden Markov Model formulation 17

3.2.1 Emission probability distributions 18
3.2.2 Prior distributions for emission distribution parameters 20
3.2.3 Initial and transition distributions 21
3.2.4 Stan implementation 22

3.3 Validation of model 22

3.3.1 Simulation study 23

4 Results 26

4.1 HMM results for daily traffic flow conditions 26

4.1.1 Weekday flow 26
4.1.2 Weekend flow 30
4.1.3 Flow during event 31
4.1.4 MCMC convergence and model diagnostics 36

5 Discussion 38

5.1 Model improvements and future research 38

6 Conclusion 40
# List of Figures

1.1 Positions of the sensors in Slakthusområdet ........................................... 2  
2.1 Graphical representation of HMM dependencies ........................................... 5  
3.1 Vehicle traffic count of arrivals after every 15 minutes ................................ 16  
3.2 Gamma probability density function with different values of $\alpha$ and $\beta$ .... 19  
3.3 Performance of HMM based on simulated data and filtered/smoothed probabili-
    ties and most probable state path .......................................................... 25  
4.1 Weekday mean of filtered and smoothed probabilities, west entrance/exit ....... 28  
4.2 Weekday mean of filtered and smoothed probabilities, south entrance/exit ..... 29  
4.3 Weekday mean of filtered and smoothed probabilities, north entrance/exit ...... 30  
4.4 Most probable state sequence for one arbitrarily chosen day ......................... 30  
4.5 Flow during first weekend in the data set .................................................. 32  
4.6 Flow during second weekend in the data set .............................................. 33  
4.7 Flow during event day: soccer game at Tele2 Arena at 19:00 ....................... 34  
4.8 Flow during event day: soccer game at Tele2 Arena at 19:00 ....................... 35  
4.9 Trace plots for $\beta_1 - \beta_4$, sensor ID 194 – 195 ...................................... 37  
B.1 Smoothed and filtered probabilities first weekend, sensor IDs 191 and 194 .... 44  
B.2 Most probable state path first weekend, sensor IDs 191 and 194 ................... 44  
B.3 Smoothed and filtered probabilities second weekend, sensor IDs 191 and 194 .. 45  
B.4 Most probable state path second weekend, sensor IDs 191 and 194 ............... 45  
B.5 Smoothed and filtered probabilities for event day, sensor IDs 190 and 191 .... 46  
B.6 Most probable state path for event day, sensor IDs 190 and 191 .................. 46  
B.7 Smoothed and filtered probabilities for event day, sensor IDs 192 and 193 ..... 47  
B.8 Most probable state path for event day, sensor IDs 192 and 193 .................. 47  
B.9 Smoothed and filtered probabilities for event day, sensor IDs 194 and 195 .... 48  
B.10 Most probable state path for event day, sensor IDs 194 and 195 ................. 48  
B.11 Smoothed and filtered probabilities for event day, sensor IDs 196 and 199 .... 49  
B.12 Most probable state path for event day, sensor IDs 196 and 199 ................. 49  
B.13 Trace plots for $\beta_1 - \beta_4$, sensor ID 190 – 191 ...................................... 50  
B.14 Trace plots for $\beta_1 - \beta_4$, sensor ID 192 – 193 ...................................... 50  
B.15 Trace plots for $\beta_1 - \beta_4$, sensor ID 196 – 197 ...................................... 50  
B.16 Trace plots for $\beta_1 - \beta_4$, sensor ID 198 – 199 ...................................... 51
# List of Tables

3.1 Vehicle sensor information summary ........................................... 15
3.2 Confusion matrices and accuracy for simulation study ..................... 24
4.1 Proportion of classified states for each sensor according to the most prob-
able state sequence ...................................................................... 26
4.2 Summary of NUTS diagnostics ....................................................... 36
C.1 Simulation study: Accuracy statistics by state for filtered probabilities .. 52
C.2 Simulation study: Accuracy statistics by state for smoothed probabilities 52
C.3 Simulation study: Accuracy statistics by state for the most probable state (sequence) ................................................................. 52
C.4 Convergence diagnostics: HMM for simulation study and real data study 53
C.5 Posterior inference: HMM for simulation study and real data study .... 54
1 Introduction

This study utilizes a Bayesian statistical approach to create useful insights about traffic flow in the geographical area of Slakthusområdet in Stockholm, based on data from 10 vehicle traffic sensors. Given reliable estimations of traffic flow and dynamics, travelers are able to choose the best routes, according to the circumstances. City planners and decision makers can use such information to develop the traffic system to be more efficient and make better use of the available road network. In order to improve the availability of traffic data from Slakthusområdet, there is a need to apply a model that can estimate traffic flow conditions. The research question of this study is: How can vehicle traffic sensor data from an urban area be used in Bayesian statistical models to estimate traffic flow conditions, in different directions and different places?

The vehicle sensors are installed at three main road entrances and exits to and from the area, in both directions. At two spots, there are four sensors, since there are two lanes in each directions. The approximate geographical positions for the sensors in the area are shown in Figure 1.1.

Since the world is rapidly urbanizing, the growing cities need to apply smarter solutions for a sustainable development to be possible. GrowSmarter is a five year long EU project under Horizon 2020, with the aim to integrate and demonstrate different smart city solutions. Stockholm, Cologne and Barcelona are the three leading cities in the GrowSmarter project and the aim is to provide other world cities valuable insights and also give opportunities for them to replicate the smart solutions. The main focus areas for the smart solutions are energy, infrastructure and transport. One of the companies participating in the GrowSmarter project is IBM, where this thesis is written. In Slakthusområdet, data are collected for analysis and are then used when developing smart solutions for the infrastructure of the the city, for example related to the large traffic flows that occur during the events in the area. The purpose with the project is to develop accessibility and create a better environment (GrowSmarter 2019).
Section 2 covers the theoretical parts for the study. First, some previous work done on similar topics are introduced. Thereafter, the essential theory of the used model, Hidden Markov Model (HMM), is introduced. Three different algorithms for estimation of the hidden Markov chains are presented; forward, forward-backward and Viterbi algorithms. Thereafter follows a subsection on the theory of Bayesian parameter estimation and the simulation method Hamiltonian Monte Carlo. Section 3 describes how the theory is applied to the specific context of the present study. A data overview is first provided followed by a subsection where the formulation and implementation of the HMMs are described. Section 4 presents the results from the HMMs. Results from both weekdays, weekends and event days are presented. Section 5 analyses the results and critically assesses the performance of the models. Furthermore, some model improvements for further studies are explained and discussed. Section 6 provides the conclusions of the study.

Figure 1.1 Positions of the sensors in Slakthusområdet
2 Theory

In this section first an introduction about traffic modeling in general is presented. Next, the theory of Hidden Markov Models (HMMs), which is the primarily choice of model for this study, is introduced. Thereafter follows a subsection on the theory of Bayesian parameter estimation and the simulation method Hamiltonian Monte Carlo.

2.1 Traffic flow modeling

The modeling and estimation of traffic conditions mainly rely on various road sensors, and can approximately be divided among modeling on road networks, see for instance Carvalho and Loureiro (2010), or on separate roads or spots, for instance Xie et al. (2010).

One common way to model traffic flows is through Origin-Destination (O-D) matrices, see Li (2005), Carvalho and Loureiro (2010). The theory is based upon a transport network composed of nodes and links that connect the nodes (Li 2005). Carvalho and Loureiro (2010) use a Bayesian Multinomial-Poisson model for estimating traffic flow through O-D matrices in two different examples. In the present study, the location of the vehicle sensors are predetermined and cannot be moved. There are no vehicle sensors inside the area, also the number of sensors are predetermined and extra sensors cannot be added. This means that the vehicle traffic volumes on unobserved links inside the area are not obtained and that vehicle traffic data only from the three main road entrances and exits can be used. Therefore a model through O-D matrices would not be a good choice.

Xie et al. (2010) use Gaussian processes (GP) regression for modeling traffic flow and doing short-term predictions on separate road spots on highways. A GP regression would not fit for an urban area where the traffic flow is fluctuating and is not as constant as for highway traffic. That is due to that a GP regression would smooth out the flow and effects from, for instance, events. Therefore, GP regression is not used in this study.

Gramaglia, Fiore, and Calderón (2014) propose a Hidden Markov Model (HMM) that models inter-arrival times collected on separate road spots on highways. Via the parametrization of the HMM, the model they build can represent free flow traffic and congested states. A novelty approach in the study of Gramaglia, Fiore, and Calderón (2014) is that they consider per-lane inter-arrivals instead of aggregated inter-arrivals.
As in Gramaglia, Fiore, and Calderón (2014) the per-lane inter-arrivals are of interest in the present study, but for urban traffic instead of highway traffic. When estimating traffic flow conditions, it is not enough to only measure the time between the arrival of one car and the arrival time of the next car (e.g. inter-arrival time). Depending on the current traffic flow, the inter-arrival times can vary both in mean and variance. A large mean for inter-arrival times can indicate both that it is free flow, i.e. not many cars arrives and therefore the inter-arrivals gets longer. It can also indicate that it is congested traffic and that it therefore takes time for the cars to pass by a sensor. The variance is possibly larger for free flow traffic than for congested traffic, because the cars possibly drive by the sensors irregularly. To make us of the distributions of inter-arrival times for different traffic flow conditions, a similar HMM approach as Gramaglia, Fiore, and Calderón (2014) proposed is suitable for the present study, but adapted for urban traffic and to more states than only free flow and congestion.

2.2 Hidden Markov Model

In this section the theory of Hidden Markov Models (HMMs) is introduced. First, the general formulation and properties for HMMs are enclosed. Thereafter, the methods for estimation of the hidden Markov chains are presented, i.e. an introduction of filtering and smoothing. In the end of this theoretical introduction of HMMs, the Viterbi algorithm is introduced, which is an algorithm for calculating the most probable state sequence.

A Hidden Markov Model (HMM) is a model where a sequence of emissions (also known as outputs) is observed, but where this sequence is modeled through a latent (i.e. hidden) state sequence that is assumed to follow a Markov chain. The analysis of HMMs aims to recover the sequence of hidden states from the observed data. The mathematical formulation of Hidden Markov Models was first proposed by Baum and Petrie (1966) and the first wider practical use of HMM was by Rabiner (1989), where it was applied to speech recognition. Today, HMMs are found in various application areas and are commonly used for different types of time series modeling, where the observed sequence is driven by a latent Markov chain.

HMM gets its name from two defining properties. The first property is the assumption that an observation $y_n, n = 1, ..., N$, is generated by an underlying process whose state is hidden and unobservable. Let there be $K$ distinct states and let $q_n =$
1, 2, ..., K denote the state at time \( n \). The second property is that the state is a discrete time (first order) Markov process:

\[
p(q_n|q_{n-1}, q_{n-2}, \ldots, q_1) \equiv p(q_n|q_{n-1}).
\]

A stochastic process is a Markov process if given the value of \( q_{n-1} \), the current state \( q_n \), is independent of all the states earlier than \( n - 1 \). The HMM also satisfy the Markov property with respect to \( y_n \); that means: given \( q_n, y_n \) is independent of the states and observations at all other times (Ghahramani 2002). A graphical representation of HMM dependencies is shown in Figure 2.1.

Let \( y_{1:N} = \{y_n\}_{n=1}^N \) be the sequence of observed variables, indexed by time \( n = 1, \ldots, N \) and let \( q_{1:N} = \{q_n\}_{n=1}^N \) be the sequence of hidden states. The bivariate process \( \{y_n, q_n\}_{n=1}^N \) is called a HMM. The general form of a HMM is (Durbin and Koopman 2012):

\[
y_n \sim p(y_n|q_n), \quad q_n \sim p(q_n|q_{n-1}), \quad q_1 \sim \pi_i = p(q_1 = i), 1 \leq i \leq K.
\]

The transition matrix \( A = \{a_{ij}\}_{1 \leq i, j \leq K} \) is a \( K \times K \) matrix of state transition probabilities, where \( a_{ij} = p(q_n = j|q_{n-1} = i) \). The state transition coefficients have the properties \( \sum_{i=1}^K a_{ij} = 1 \) and \( a_{ij} \geq 0 \). \( \pi = \{\pi_i\}_{i=1}^K \) is a vector with the initial state probability distributions.

In the HMM literature, two functions are referred to as the likelihood. The first one is the so called complete-data likelihood or the joint distribution of the observations and states \( \{y_n, q_n\}_{n=1}^N \). That means the probability that \( y_{1:N} \) and \( q_{1:N} \) occurs simultaneously. The second one is the marginal likelihood or the joint distributions of only the observations (Leos-Barajas and Michelot 2018). The complete-data likelihood is given
by:

\[ \mathcal{L}_c = p(\mathbf{y}_{1:N}, \mathbf{q}_{1:N}) = p(\mathbf{q}_{1:N})p(\mathbf{y}_{1:N} | \mathbf{q}_{1:N}) \]

\[ = \pi_i \prod_{n=2}^N p(q_n | q_{n-1}) \prod_{n=1}^N p(y_n | q_n) \]  

\[ = \pi_i \prod_{n=2}^N a_{q_{n-1},q_n} \prod_{n=1}^N p(y_n | q_n). \]

The marginal likelihood requires summation over all possible state sequences and is given by (Scott 2002):

\[ \mathcal{L}_m = p(\mathbf{y}_{1:N}) = \sum_{q_1=1}^K \cdots \sum_{q_N=1}^K \pi_i \prod_{n=2}^N a_{q_{n-1},q_n} \prod_{n=1}^N p(y_n | q_n). \]

The state-dependent emission distribution, \( p(y_n | q_n) \), can either be discrete or continuous in HMMs, where each observation \( y_n \) is generated from state \( q_n \).

2.2.1 Filtering and smoothing

This subsection discusses on how to estimate the hidden Markov chain. There are several hidden quantities of interest that can be inferred using different algorithms, for instance the probability distributions of the hidden states given the data and model and the most probable state sequence. Here we summarize some of the most relevant methods for obtaining those quantities: forward, forward-backward and Viterbi algorithms.

The filtering process is based on a forward algorithm and infers the (marginal) posterior distribution of the hidden states at each time step, based on all previous information from the observations, \( p(q_n | \mathbf{y}_{1:n}) \). The filtering can be done online, i.e. recursively, as the data streams in (Murphy 2012).

The forward algorithm can be computed in two steps. The first step is to calculate the one-step-ahead predictive density, which is given by:

\[ p(q_n = j | \mathbf{y}_{1:n-1}) = \sum_i a_{ij} p(q_{n-1} = i | \mathbf{y}_{1:n-1}). \]

Assuming this has already been calculated, the forward probability \( \alpha_n(j) \) is updated, in
the second step, with observed data at time step $n$ using Bayes theorem:

$$
\alpha_n(j) \equiv p(q_n = j | y_{1:n}) = p(q_n = j | y_n, y_{1:n-1}) = Q_n^{-1} p(y_n | q_n = j, y_{1:n-1}) p(q_n = j | y_{1:n-1}) \propto p(y_n | q_n = j) p(q_n = j | y_{1:n-1}) = \sum_i a_{ij} \alpha_{n-1}(i) p(y_n | q_n = j),
$$

where $p(y_n | q_n = j)$ is called the local observation (evidence) at time $n$ and where the normalization constant $Q_n$ is given by:

$$
Q_n = p(y_n | y_{1:n-1}) = \sum_j p(q_n = j | y_{1:n-1}) p(y_n | q_n = j).
$$

The updating algorithm results in the filtered distribution at time step $n$. Since the complete-data likelihood in Equation 1-3 is in a more simple form than the marginal likelihood in Equation 4, it is often used for conducting inference for parameters and states jointly. However, when using Stan, which is introduced in Section 2.3.1, the evaluation of the marginal likelihood is needed (Leos-Barajas and Michelot 2018). The marginal likelihood in Equation 4 can be complex to calculate directly even for small $K$, when $N$ grows large (Scott 2002). The computations can be solved by using the forward procedure (instead of direct calculations) which reduces the complexity of the calculations (Rabiner 1989). So, in addition to compute the posterior of the hidden states, the forward algorithm can be used to compute the marginal likelihood:

$$
\log \mathcal{L}_m = \sum_{n=1}^{N} \log p(y_n | y_{1:n-1}) = \sum_{n=1}^{N} \log Q_n.
$$

The marginal log likelihood is presented, since the log likelihood is the one used in the Stan-implementation later on.

The smoothing process is based on a forward-backward algorithm and infers the (marginal) posterior distribution of the hidden states at each time step, $p(q_n | y_{1:N})$. The probability of being in state $j$ at time $n$, given all the observations, also called the smoothed posterior marginal, is defined by:

$$
\gamma_n(j) \equiv p(q_n = j | y_{1:N}) = \frac{\alpha_n(j) \beta_n(j)}{\sum_{j=1}^{K} \alpha_n(j) \beta_n(j)} \propto \alpha_n(j) \beta_n(j),
$$

where $\beta_n(j)$ is representing the backward procedure, i.e. the conditional likelihood of future observations given that the hidden state at time $n$ is $j$ and is defined by:

$$
\beta_n(j) \equiv p(y_{n+1:N} | q_n = j).
$$
The smoothing process is evaluated offline, given all available data. The uncertainty in the data is reduced by conditioning on both past and future. The forward-backward procedure is calculated first from left to right, described for the forward algorithm above, and then from right to left, combined at each node (Murphy 2012). If $\beta_n$ is already computed, $\beta_{n-1}$ can recursively be computed, from right to left, by:

$$
\begin{align*}
\beta_{n-1}(i) & \equiv p(y_{n:N}|q_{n-1} = i) \\
& = \sum_j p(q_n = j, y_n, y_{n+1:N}|q_{n-1} = i) \\
& = \sum_j p(y_{n+1:N}|q_n = j, g_{n-1}=\bar{i}, y_{n})p(q_n = j, y_{n}|q_{n-1} = i) \\
& = \sum_j p(y_{n+1:N}|q_n = j)p(y_n = j, g_{n-1}=\bar{i})p(q_n = j|q_{n-1} = i) \\
& = \sum_j \beta_n(j)p(y_n = j) a_{ij}.
\end{align*}
$$

The base case is defined by $\beta_N(i) \equiv 1$ and is the probability of a non-event.

2.2.2 Most probable state sequence

The Viterbi algorithm (Viterbi 1967) is an efficient maximum a posteriori probability (MAP) detector, when the state process is a finite-state Markov process. That is, the Viterbi algorithm provides an efficient way of finding the most probable state sequence. The result of the Viterbi algorithm is a sequence of states, $q_{1:N}^* = \{q_1^*, q_2^*, ..., q_N^*\}$, which represents the most likely state sequence given the observations and model parameters (Rabiner 1989). The most probable state sequence can be expressed as:

$$
q_{1:N}^* = \arg \max_{q_{1:N}} p(q_{1:N}|y_{1:N}). \quad (5)
$$

See Appendix A.1 for details on how to calculate the most probable state path with the Viterbi algorithm. Murphy (2012) points out that the (jointly) most probable sequence of states (Equation 5) is not necessarily the same as the sequence of (marginally) most probable states, which is instead defined by:

$$
\hat{q}_{1:N} = \left( \arg \max_{q_1} p(q_1|y_{1:N}), ..., \arg \max_{q_N} p(q_N|y_{1:N}) \right).
$$

2.3 Bayesian parameter estimation

The frequentist way of substituting in the maximum likelihood estimators (MLEs) of the parameters, $\theta$, in the filtering and smoothing recursion processes (which are described
in Section 2.2.1) can suffer from complications in taking properly into account the uncertainty about $\theta$. In the Bayesian way, a more consistent formulation of the problem is offered, where the unknown parameters $\theta$ are treated as random (Petris, Petrone, and Campagnoli 2009). The core of Bayesian inference is conditioning on data, in order to learn about parameter values. The Bayesian statistical approach has become more popular in applications in later years, much due to the availability of modern and efficient computational tools.

The prior distribution is the probability distribution that express the beliefs about some uncertain quantity before the data is taken into account. By having the ability to specify prior distributions, more information can be incorporated in the statistical inference. The posterior distribution is the conditional probability distribution of the unobserved quantities of interest, given the observed data. The posterior can be seen as a compromise between the prior and the data (Gelman, Carlin, et al. 2013).

The choice of and reliance on priors are often mentioned as the controversial aspects of Bayesian statistics (Murphy 2012). A prior can be non-informative and therefore plays a small role for the posterior. It can also be weakly informative, i.e. contains some information, enough to keep the posterior within reasonable bounds. Of course, a prior can also be informative including more precise information about the parameter of interest. The less information the data contains, the bigger role the prior plays (Gelman, Carlin, et al. 2013). The unknown parameters, $\theta$, in a HMM, that require specification by prior distributions are the transition probabilities, the parameters of the emission distributions, and the initial state distribution.

For the data sequence $y_{1:N}$, the posterior distribution over the parameters $\theta$ can, by using Bayes theorem, be computed by:

$$p(\theta|y_{1:N}) = \frac{p(y_{1:N}|\theta)p(\theta)}{\int_{\theta} p(y_{1:N}|\theta)p(\theta)d\theta} \propto p(y_{1:N}|\theta)p(\theta),$$

where $p(y_{1:N}|\theta)$ is the marginal likelihood $L_m$ and $p(\theta)$ is the prior distribution of the parameters. The product of these two is proportional to the posterior belief distribution. The term in the denominator is normalizing constant, that is, a constant that makes the posterior density integrate to one.

The following subsections are structured around tasks faced by the Bayesian modeler: obtaining the posterior distribution of model parameters and using diagnostics to assess MCMC convergence.
2.3.1 Hamiltonian Monte Carlo and No-U-Turn sampler

The posterior distribution of the parameters $p(\theta|y_{1:N})$ summarizes everything that is known about $\theta$ and is the core of Bayesian statistics (Murphy 2012). In Bayesian statistics, where one goal is to represent inference using posterior draws, *Markov Chain Monte Carlo* (MCMC) sampling methods plays an important part. In the traditional (frequentist) way, the inference of HMMs are often carried out by using the expectation-maximization (EM) algorithm to find the MLE or MAP estimates. When using a Bayesian approach, the inference is instead in general implemented through MCMC sampling (Rydén 2008).

Markov chain Monte Carlo (MCMC) algorithms generate posterior samples sequentially. The distribution of the sampled draw only depends on the previous draw, and therefore the draws forms a Markov chain. For each step in the simulation, the approximate posterior distribution is improved and the goal is to converge to the target posterior distribution, $p(\theta|y_{1:N})$. MCMC is used when it is not possible, or is computationally inefficient, to sample $\theta$ directly from $p(\theta|y_{1:N})$ (Gelman, Carlin, et al. 2013). MCMC sampling can be used to simulate HMM parameters from their posterior distribution given observed sequences. The MCMC sampler alternates between sampling the parameters conditional on the data and the hidden Markov chain, and updating the hidden chain conditional on the data and parameters (Rydén 2008).

Gibbs sampling and Metropolis-Hastings are well-known and easily implemented MCMC algorithms based on conditional sampling and are used in software such as BUGS. Gibbs sampling and Metropolis-Hastings converges very slowly to the target distribution if the models are complicated. The slow convergence is due to the random walk behaviour and the conditional sampling of the algorithms. This problem of slow convergence can be solved by using less complex models or more complicated algorithms, for instance *Hamiltonian Monte Carlo* (HMC). HMC is a family of MCMC algorithms that borrows ideas from physics that suppress the local random walk behaviour in Gibbs sampling and Metropolis-Hastings. The idea is to add auxiliary momentum variables $r$ for each model parameter $\theta$. That allows it to move much more rapidly through the target distribution and therefore become more efficient (Gelman, Carlin, et al. 2013).

HMC appeared the first time in Duane et al. (1987) applied to molecular dynamics and are today widely used by statisticians. Stan is a relatively new C++ built
software that implements HMC and is used for Bayesian modeling and inference (Stan Development Team 2018a). RStan, the R interface to Stan, is used in this study.

In HMC, an auxiliary momentum variable $r$ is introduced for each model parameter $\theta$. Both $r$ and $\theta$ are updated together in a new algorithm. $r$ gives the expected direction and distance of the jump in exploring the target distribution of $\theta$. Successive jumps tend to be in the same direction, which makes this algorithm faster through the exploration of the distribution than the random-walk behaviour algorithms (Neal 2011).

The momentum variables are in most applications of HMC, including RStan, drawn independently from the multivariate standard normal distribution $\mathcal{N}(0, \Sigma)$. The covariance matrix, $\Sigma$, may in RStan be set to a identity matrix (unit diagonal) or estimated to a diagonal matrix (Stan Development Team 2018b).

The joint density of $\theta$ and $r$, $p(\theta, r)$ defines the Hamiltonian dynamics system, $H(\theta, r)$. The Hamiltonian system is defined by:

$$H(\theta, r) = - \log p(\theta, r) = - \log p(r|\theta) - \log p(\theta) = K(r|\theta) + V(\theta),$$

where $K(r|\theta)$ is called the kinetic energy, which is the term corresponding to the density over the auxiliary momentum. $V(\theta)$ is called the potential energy, which is the term corresponding to the density of the target distribution. See Neal (2011) for more technical details.

The HMC algorithm has two steps for every iteration. The first one changes only the momentum and the second one can change both position and momentum (Neal 2011). In the first step, the momentum variables $r$ are drawn. In the second step an update is performed, using Hamiltonian equations, see Equations 6-7, to propose the next step for exploring the distribution.

The Hamilton’s equations, for time $t$ is:

$$\frac{d\theta}{dt} = + \frac{\partial H}{\partial r} = \frac{\partial K}{\partial r}, \quad (6)$$

$$\frac{dr}{dt} = - \frac{\partial H}{\partial \theta} = - \frac{\partial K}{\partial \theta} - \frac{\partial V}{\partial \theta}, \quad (7)$$

where $\partial V/\partial \theta$ is the gradient of the logarithm of the target distribution.

The Hamiltonian dynamics is simulated for $L$ steps, commonly using the leapfrog method (See Appendix A.2 and Neal (2011)), with the step size $\epsilon$. A new state, $(\theta^*, r^*)$,
is proposed, which is accepted with probability

\[
\alpha = \min \left[ 1, \exp \left( -H(\theta^*, r^*) + H(\theta, r) \right) \right]
\]

\[
= \min \left[ 1, \exp \left( -K(r^*|\theta^*) + K(r|\theta) - V(\theta^*) + V(\theta) \right) \right].
\]  

(8)  

(9)

If the the proposed step is rejected, the next state is the same as the current state.

In other words, when using HMC, it is possible to follow a direction assigned for each step, instead of searching the whole parameter space with random jumps. At every new point, a new direction to follow is assigned. When continuing this tracing process a coherent trajectory through the parameter space is obtained and the exploration of the distribution goes as quickly as possible.

Neal (2011) visualizes the (2 dimensional) target distribution exploration process by a hockey puck exploring a frictionless ice. The system is described by the position of the puck, \( \theta \), and the momentum of the puck (its mass times its velocity), \( r \). The puck will move when pushing it towards an arbitrary direction. Since the ice is frictionless, the puck will move forever, and on a totally flat ice it will move with constant velocity. If the ice has a positive slope \( \partial H/\partial \theta > 0 \), the puck can still climb due to the momentum and increasing potential energy \( V(\theta) \), but on the same time the kinetic energy \( K(r|\theta) \) will decrease. The puck will still climb upwards until \( K(r|\theta) \) (and \( p \)) is zero, and thereafter start to slide down.

HMC has a high efficiency compared to other random walk behaviour MCMC methods such as Gibbs sampling and Metropolis Hastings. But, the efficiency has a price. HMC requires the gradient of the log-posterior, see Equation 7, which is complex to compute and generates less iterations per second than Gibbs or Metropolis. Also, HMC requires the user to specify the number of steps, \( L \), and the step size, \( \epsilon \) for the sampling. The efficiency of HMC can decrease if the choice of either of these parameters is poor. RStan is able to automatically optimize \( \epsilon \) and automatically adapt \( L \) during sampling, by using the default No-U-Turn sampler (NUTS) algorithm, a special case of HMC (Stan Development Team 2018b). Therefore, the problematic choices of \( \epsilon \) and \( L \) are eliminated (Hoffman and Gelman 2011). Also, RStan estimates \( \Sigma \) based on warmup sample iterations (Hoffman and Gelman 2014).
2.3.2 MCMC convergence and model diagnostics

When the number of MCMC simulation draws approaches infinity, the convergence to the target distribution is usually said to be guaranteed. In contrast, the guarantees about the behavior after a finite number of simulation draws are seldom strong and it can be challenging to monitor the convergence of the iterative MCMC stochastic algorithm.

Divergence can occur when the parameter space is hard to explore. Divergent iterations indicates a bias in the posterior samples (Lieu et al. 2017). One attempt to improve concerns of poor convergence is to run more than one chain to see if the obtained distributions are similar (Vehtari et al. 2019). The default number of chains in Stan is four (Stan Development Team 2018b). Running multiple chains can be critical to MCMC convergence diagnostic, since, for instance, the chains can explore different parts of the target distribution or may fail to attain stationarity. These problems can arise when the target distribution is multimodal or when the chain is trapped in a region of high curvature and with a large step size, so problems of making an acceptable proposal for the next step arises. Therefore, it is important to use information both between and within chains. Also, it is not always realistic to make and interpret trace plots if the number of parameters is large and therefore a numerical summary of convergence is needed (Vehtari et al. 2019), for instance the measure of \( \hat{R} \) introduced by Gelman and Rubin (1992). If the chains converged, the ratio of between-to within-chain variance for the model parameters and other quantities of interest, \( \hat{R} \), should be small and close to 1. \( \hat{R} < 1.1 \) is recommended. split-\( \hat{R} \) is a version of the measure that compares the first half of the chain with the second half, to detect lack of convergence (see Gelman, Carlin, et al. (2013)).

Vehtari et al. (2019), where a majority of the authors are developers of Stan, introduce an improved measure for assessing convergence of MCMC. There it is shown that the classical convergence diagnostics \( \hat{R} \) has some serious flaws. For example, \( \hat{R} \) may fail to detect convergence when, for instance, the chains have different variances but the same mean. Another problem with \( \hat{R} \) is that it says little about the convergence in the tails.

Not only is a small value of \( \hat{R} \) preferred to ensure convergence. Also, the effective sample size (ESS) should be large. The ESS captures 'how many independent draws contain the same amount of information as the dependent sample obtained by the MCMC
algorithm. An ESS above 400 is preferred, i.e. when running four chains a ESS of 50 per split chain is preferred. Vehtari et al. (2019) points out that the ESS easily can be overestimated for multimodal distributions when using the split-$\hat{R}$-adjustment.

To improve the convergence diagnostics, Vehtari et al. (2019) recommends a rank-normalized split-$\hat{R}$ and a rank-normalized ESS for heavy tailed distributions. The authors recommend a folded-split-$\hat{R}$ when the locations are the same but the scales are different. See Vehtari et al. (2019) for more details and possible updates. Here however, only the traditional split-$\hat{R}$ and ESS will be used, since the rank-normalized and folded-split versions are not thoroughly evaluated in practice yet.

One other important diagnostic tool for NUTS is the number of steps to take in each iteration, i.e. tree depth. Even if NUTS select the tree depth itself, there is still a maximum number of steps that the sampler will try. If the tree depth is zero, it indicates that the first leapfrog step is immediately rejected, which in turn is an indication of extreme curvature and/or a poorly chosen step size. If the tree depth is equal to the maximum tree depth, it is an indication that NUTS takes too many steps. If the sampler takes many steps, and often hits the maximum tree depth, it can be a sign of poor adaption and can be due to a difficult posterior to sample from, or a very high acceptance rate. While divergence can lead to biased inference, a too small maximum tree depth affects the efficiency (Stan Development Team 2018b).
3 Methodology

In this section a data overview is first provided, including data visualizations of raw data and variable descriptions, followed by a subsection where the formulation and implementation of the HMMs are described.

3.1 Data overview

There are 10 vehicle traffic sensors installed in Slakthusområdet. In Table 3.1, information about the sensors positions are presented; if the sensor is located near the West, South or North entrance/exit port to/from Slakthusområdet, if the sensor register vehicles that are driving in to or out from the area and if the sensor focuses on right or left lane (or if it is only a single lane). In Table 3.1 also the number of observations, which are presented later in this section, are presented. In Figure 3.1 the count of passages after every 15 minutes are presented, for every sensor. The vehicle sensors identifies the cars by reading the number plates. The number plate information is used to connect with the Swedish Transport Agency (Transportstyrelsen) and different variables with information about the vehicle are obtained. The positions of all the sensors can be seen in Figure 1.1.

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Port</th>
<th>In/out</th>
<th>Lane</th>
<th>Number of total unfiltered arrivals</th>
<th>Number of used $\Delta \tau_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>West</td>
<td>In</td>
<td>Right</td>
<td>40 828</td>
<td>37 216</td>
</tr>
<tr>
<td>191</td>
<td>West</td>
<td>In</td>
<td>Left</td>
<td>29 950</td>
<td>25 440</td>
</tr>
<tr>
<td>192</td>
<td>West</td>
<td>Out</td>
<td>Left</td>
<td>39 758</td>
<td>35 514</td>
</tr>
<tr>
<td>193</td>
<td>West</td>
<td>Out</td>
<td>Right</td>
<td>21 429</td>
<td>18 148</td>
</tr>
<tr>
<td>194</td>
<td>South</td>
<td>In</td>
<td>Right</td>
<td>82 452</td>
<td>75 166</td>
</tr>
<tr>
<td>195</td>
<td>South</td>
<td>In</td>
<td>Left</td>
<td>32 739</td>
<td>26 938</td>
</tr>
<tr>
<td>196</td>
<td>South</td>
<td>Out</td>
<td>Left</td>
<td>79 524</td>
<td>70 371</td>
</tr>
<tr>
<td>197</td>
<td>South</td>
<td>Out</td>
<td>Right</td>
<td>32 743</td>
<td>26 247</td>
</tr>
<tr>
<td>198</td>
<td>North</td>
<td>In</td>
<td>Single lane</td>
<td>30 916</td>
<td>26 594</td>
</tr>
<tr>
<td>199</td>
<td>North</td>
<td>Out</td>
<td>Single lane</td>
<td>37 272</td>
<td>31 418</td>
</tr>
</tbody>
</table>
In the vehicle data set, there is a timestamp variable, which is defined as date and time (in seconds) when a vehicle passed a sensor. Inter-arrival time is a measure used in, for instance, queuing theory, and is therefore used for modeling traffic congestion. The measure of inter-arrival time is in the case of this study used as the observed data and is defined as the time between the arrival of one car and the arrival of the next car.

Time is divided into $N$ time intervals, by equally spaced time points $t_1, ..., t_N$, with $t_0 = 0$. In an time interval $[t_n, t_{n+1}]$, $M$ cars arrives at times $\tau_1, ..., \tau_M$, with $\tau_0 = 0$. 

Figure 3.1 Vehicle traffic count of arrivals after every 15 minutes
Hence, inter-arrival time is defined by \( \Delta \tau_i = \tau_i - \tau_{i-1} \). Inter-arrival time is calculated for each car after the first one, \( i = 2, \ldots, M \) and is measured in seconds in the present study.

The sample used in this study is from 2019-03-22 00:00:00 to 2019-04-04 23:59:59. The sample contains 10 weekdays and 4 weekend days. A transition from winter to summer time was done on 2019-03-31. That leads to a total number of 1340 time intervals, with length 15 minutes. More data are available, but for this study, no more than two weeks of data is used due to the time complexity of running the parameter sampling and recursive algorithms. In further work, using a larger sample would preferable.

After calculating inter-arrival times for all sensors it can be seen that there are some inter-arrival times that are 0. That can be due to, for instance, double registered passages. In the data set there is an ordinal variable of detection level that indicates if 1) something has passed the sensor, 2) also, a licence plate is successfully read, 3) also, the vehicle is found in Vägtrafikregistret or 4) also, a zip code exists for the vehicle. If the detection level is that something has passed the sensor, there is a risk that something has triggered the sensor to register a vehicle twice, at the same timestamp. It is logical to only save observations that satisfies \( \Delta \tau_i > 1 \), since two cars cannot pass the sensor at the same time. The final number of observations used, for each sensor, calculated after dividing in 1340 time intervals and after removing observations with \( \Delta \tau_i \leq 1 \), can be seen in Table 3.1. Also, the unfiltered counts of total arrivals are presented in Table 3.1.

One of the main events during the period of 2019-03-22 to 2019-04-04 was a national soccer game (Monday) 2019-04-01 19:00 at Tele2 Arena (Stockholm Live 2019).

### 3.2 Hidden Markov Model formulation

In this study, a Bayesian probabilistic approach utilizing Hidden Markov Model (HMM) is proposed to model the stochastic variation of traffic flow conditions.

Implementing forward, forward-backward and Viterbi algorithms for HMMs in Stan has been covered by Leos-Barajas and Michelot (2018). These Stan implementations for the recursive algorithms are used in the case of this study. The output from the forward algorithm, which is the filtered posterior distribution over the hidden states given observations until time \( n \), is interpreted as the posterior distribution over different traffic flow condition states given previous inter-arrival times. In the same manner, the output from the forward-backward algorithm, which is the smoothed posterior distribution over
the hidden states given all observations, is interpreted as the posterior distribution over traffic flow condition states given all the inter-arrival times.

The MCMC sampling method NUTS is then used to simulate HMM parameters from their posterior distributions given the observed sequences of data and for computing the conditional probabilities of hidden states.

Inter-arrival time sequences within 15 minutes time intervals are used as observation sequences for HMMs. The HMMs are initialized by defining prior distributions for the parameters of the the initial,- transition- and emission distributions and then the MCMC sampling is done.

The length of 15 minutes for the time interval is set due to the interest of knowing the traffic flow condition based on as recent time as possible, but still has a reasonable count of car passages to base the analysis on. To be noted is that the HMMs was also first modeled with 20 minutes intervals, with less adequate results and more divergent observations than the HMMs for 15 minutes intervals and are therefore not included in the results.

3.2.1 Emission probability distributions

Gramaglia, Serrano, et al. (2011) show that a Gaussian-exponential mixture model accurately characterizes inter-arrival times for vehicle traffic. They show that free flow conditions have exponential inter-arrivals and that congested conditions yield Gaussian inter-arrivals. Their work are based on highway vehicle traffic. In the case of this study the HMMs are applied for urban traffic inter-arrivals, which reasonably does not have the same density of the traffic flow as for the highways. The choice of a normal distribution is not here seen as a legitimate choice, since inter-arrival times cannot be negative.

In the case of this study, the inter-arrival times $\Delta \tau_i$ is modeled as gamma distributed, which only allows positive values. The gamma distribution is characterized using shape $\alpha$ and rate (inverse scale) $\beta$. Different values of $\alpha$ and $\beta$ changes the shape of the probability density function. A gamma distribution with shape parameter $\alpha = 1$ and scale parameter $\beta$ is an exponential distribution with expected value $\beta$, and a larger value for $\alpha$ makes the distribution look more normal shaped. This flexibility makes the gamma distribution a good choice for $\Delta \tau_i$. See examples of different values of $\alpha$ and $\beta$ in Figure 3.2. The hidden state for a time interval $[t_n, t_{n+1}]$ is $q_n$, according to the definitions in
Section 2.2. The corresponding probability density function for an observation $\Delta \tau_i$ is

$$f_{q_n}(\Delta \tau_i; \alpha_{q_n}, \beta_{q_n}) = \frac{1}{\Gamma(\alpha_{q_n})} \beta_{q_n}^{\alpha_{q_n}} (\Delta \tau_i)^{\alpha_{q_n}-1} e^{-\beta_{q_n} \Delta \tau_i}$$

for $\Delta \tau_i > 0$ and $\alpha_{q_n}, \beta_{q_n} > 0$, where $\Gamma(\cdot)$ is the gamma function. The mean of the gamma distribution is $E(\Delta \tau_i) = \alpha/\beta$ and the variance is $Var(\Delta \tau_i) = \alpha/\beta^2$.

![Figure 3.2 Gamma probability density function with different values of $\alpha$ and $\beta$](image)

The cumulative density function is

$$F_{q_n}(\Delta \tau_i; \alpha_{q_n}, \beta_{q_n}) = \frac{\gamma(\alpha_{q_n}, \beta_{q_n} \Delta \tau_i)}{\Gamma(\alpha_{q_n})},$$

where $\gamma(\cdot)$ is the lower incomplete gamma function.

The likelihood for the observations $\Delta \tau_i$ in $[t_n, t_{n+1}]$ becomes

$$
\mathcal{L}_m = f_{q_n}(\Delta \tau_1) f_{q_n}(\Delta \tau_2) \cdots f_{q_n}(\Delta \tau_{M-1})(1 - F_{q_n}(t_{n+1} - \tau_M)) = \\
= \left(1 - F_{q_n}(t_{n+1} - \tau_M)\right) \prod_{i=1}^{M} \frac{1}{\Gamma(\alpha_{q_n})} \beta_{q_n}^{\alpha_{q_n}} (\Delta \tau_i)^{\alpha_{q_n}-1} e^{-\beta_{q_n} \Delta \tau_i} = \\
= \left(1 - F_{q_n}(t_{n+1} - \tau_M)\right) \left(\frac{\beta_{q_n}^{\alpha_{q_n}}}{\Gamma(\alpha_{q_n})}\right)^N \left(\prod_{i=1}^{M} \Delta \tau_i\right)^{\alpha_{q_n}-1} e^{-\beta_{q_n} \sum_{i=1}^{N} \Delta \tau_i}.
$$
with the convention that $\prod_{i=1}^{\theta} \Delta \tau_i = 1$. That is, it can be zero (inter-)arrivals in a time interval, and then the product should be one so the likelihood becomes the probability of no arrivals. The term $1 - F_{q_n}(t_{n+1} - \tau_M)$ in $L_m$ is due to that no cars arrived between the last car’s arrival time $\tau_M$ and the upper interval limit $t_{n+1}$, which has probability $F_{q_n}(t_{n+1} - \tau_M)$. The likelihood gives the following triplet of sufficient statistics:

$$T(\Delta \tau) = \left( \prod_{i=1}^{M} \Delta \tau_i, \sum_{i=1}^{M} \Delta \tau_i, t_{n+1} - \tau_M \right).$$  \hspace{1cm} (10)

Likelihood based inference for $\Delta \tau_i$ can always be based on this triplet, independent of how many arrivals there are in an interval. This triplet of sufficient statistics is calculated for each time interval and is used as the observed data sequence $y_{1:N}$ input in the HMMs in Stan.

There are $K = 4$ hidden states in the case of this study; $q_t = 1$: nightly free flow, $q_t = 2$: free flow, $q_t = 3$: mixture and $q_t = 4$: congestion. Nightly free flow is included as a state to separate nights from days, since due to low traffic volume, the distribution of $\Delta \tau_i$ is completely different during night. In the days, the main interest is if it is either free flow or congestion. Due to a hypothesis that it is a small proportion of congestion during a day and a large proportion of free flow, also a mixed state is included. The mixed state would capture something closer to congestion than free flow, which also is interesting to know about even if it is not fully congested traffic.

### 3.2.2 Prior distributions for emission distribution parameters

To define the distributions of different traffic flow states, the parameter $\alpha$ is set as constant for all four state dependent distributions, which defines the shape of the gamma distributions. After looking at the data for different 15 minutes intervals, it is concluded that the choice of a normal shaped distribution, as in Gramaglia, Serrano, et al. (2011), for congested conditions is not fully reasonable in the present study, probably because of less traffic. Still, a larger value or $\alpha$ is set for the congested state (than for free flow), which makes it move more towards the normal shaped distribution. The $\alpha$’s for mixture state and congested state are set to larger values than 1 ($\alpha = 1$ would give an exponential distribution). For the two free flow states, the values for the $\alpha$’s are set to smaller values than 1; $\alpha_1 = 0.5, \alpha_2 = 0.8, \alpha_3 = 1.1, \alpha_4 = 1.4$.

The priors chosen for the $\beta$’s are $\beta_1 \sim Exponential(10), \beta_2 \sim Exponential(5), \beta_3 \sim Exponential(1)$ and $\beta_4 \sim Gamma(2,2)$. A prior belief about the rate parameters in the
emission distributions, the \( \beta \)'s, are that the mean of the \( \beta \)'s are smaller for free flow and larger for congestion. In combination with the constant values for the shape parameters, the \( \alpha \)'s, that would lead to larger variance for free flow in the emission distributions, and smaller variance for congestion, since the variance of the gamma distribution is \( Var(\Delta \tau_i) = \alpha / \beta^2 \). The priors are also chosen as to prevent divergences in the sampling. The priors are informative, with small expected values. The model was tried out with different priors, before deciding on the final quartet. First, non-informative priors was chosen, without satisfying results and without reasonable estimates of the hidden states. This can be due to that whilst allowing large values of \( \beta \), at the same time the variance of the posterior is allowed to be very small. By re-parametrizing the model to the above priors, the problems with divergence is reduced, but not yet fully solved.

### 3.2.3 Initial and transition distributions

The initial distributions reflects the probabilities of being in a certain traffic condition at the initial time step. It is assumed that \( \sum_{i=1}^{K} \pi_{qi} = 1 \) and that \( \pi_{qi} > 0 \ \forall K \). A uniform distribution on a simplex is set as prior distribution for \( \pi \).

The priors on the transition matrix \( A \) are, for each row, modeled as independent Dirichlet-distributed:

\[
A[1,] \sim Dirichlet(0.7, 0.1, 0.1, 0.1) \\
A[2,] \sim Dirichlet(0.1, 0.7, 0.1, 0.1) \\
A[3,] \sim Dirichlet(0.1, 0.1, 0.7, 0.1) \\
A[4,] \sim Dirichlet(0.1, 0.1, 0.1, 0.7).
\]

The Dirichlet priors used in this way assumes that the state is likely to change to any other state with equal probability, but has a higher probability of staying in the current state. The hidden states are sequentially sampled based on the transition probabilities.

The Dirichlet distribution is a distribution on the simplex, where the observations \( x_i \in (0, 1) \ \forall i \) and \( \sum_{i=1}^{K} x_i = 1 \). The number of parameters can be \( K \geq 2 \) are called concentration parameters; \( \alpha_1, ..., \alpha_K, \alpha_i > 0 \ \forall i \). The probability density function for the
Dirichlet distribution is defined by:

\[ f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \]

where the normalizing constant \( B(\alpha) \) is the multivariate beta function.

### 3.2.4 Stan implementation

The default number of iterations for each chain (including warm up) in Stan is 2000. The default number of warmup (or burn-in) iterations per chain is the number of iterations divided by 2. The warmup samples should not be used for inference. The default number of Markov chains in Stan is 4 (Stan Development Team 2018b), which is also used in this study.

The default target average proposal acceptance probability, is in Stan 0.8. See the acceptance probability in Equation 8-9. By increasing the acceptance rate it will force the sampling procedure to take smaller steps. This tend to make the sampling slower, since a smaller step size will require more steps, but it can reduce the chance of divergence (Stan Development Team 2018b). In this study an acceptance rate of 0.99 is used, which is also recommended by Stan Development Team (2018b) for reducing divergence.

The divergence is checked by monitoring the number of indicated divergent iterations that occurred whilst sampling. Convergence is checked using trace plots and calculation of the convergence criteria split-\( \hat{R} \) and ESS.

The maximum tree depth can be increased in Stan to ensure that the NUTS sampling can grow larger over the target posterior distribution. The default tree depth is 10 (Stan Development Team 2018b), and is in the case of this study increased to 12.

### 3.3 Validation of model

Since there is no way to know the hidden states due to the unsupervised nature of the problematization of this study, it is difficult to determine exactly how accurate the model is. A simulation study is implemented, to see if the model performs well on simulated data. The implementation and results of the simulation study are described in the following subsection. The results can be used as indication on if the model performs well.
3.3.1 Simulation study

A random vector with 4 different levels representing the different states; 1) nightly free flow, 2) free flow, 3) mixture and 4) congestion, respectively, at 1340 different time intervals are randomly generated with transitions based on the transition probabilities in

\[ A = \begin{bmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{bmatrix}. \]

The interval length is set to 15 minutes as in the real study. Then random inter-arrival times are generated in each time interval. If the current value of the state vector is 1, inter-arrival times from \( \text{Gamma}(0.5, 0.01) \) are generated. The other three distributions which are used are \( \text{Gamma}(0.8, 0.03), \text{Gamma}(1.1, 0.05) \) and \( \text{Gamma}(1.4, 0.08) \), which represents states 2-4. Then the sufficient statistics from Equation 10 are calculated for each time interval and are used as observation inputs to the Stan program. The prior distribution for emission parameters, initial distribution and transition distributions used for the real data are also implemented in the simulation study, and the same Stan implementation is done.

By comparing the randomly drawn "true" reference states, with the most probable state based on the filtered and smoothed probabilities for each time interval and the most probable state sequence, the overall accuracy, sensitivity and specificity for the model is evaluated. The most probable state according to the filtered and smoothed probabilities are calculated by taking the mean of the probabilities for each NUTS iteration and state. The largest mean for every time interval is seen as the most probable state for the interval.

The overall accuracy (ACC) is defined by:

\[ ACC = \frac{TP + TN}{TP + TN + FP + FN}, \]

where TP stands for true positive, TN for true negative, FP for false positive and FN for false negative.

In Table 3.2 confusion matrices for all three comparisons (filtered and smoothed probabilities and most probable state vs. predicted state), and also the overall accuracy, are presented. The accuracy is high for all three comparisons, which indicates that the model performs well for simulated data. Also, for each state and comparison, the
sensitivity, or true positive rate $TPR = (TP/(TP + FN))$ and the specificity, or true negative rate $TNR = (TN/(TN + FP))$ are calculated. Those two measures and a balanced accuracy $(TPR + TNR)/2$ are calculated and can be seen in Appendix C.1. The proportion of actual positive cases which are correctly identified is around 0.6 for all state 3, mixture state, for three comparisons. The proportion of actual negative cases which are correctly identified are above 0.92 for all comparisons and states. In Figure 3.3, the performance of the HMM is visualized for 60 time intervals, for filtered probabilities in Figure 3.3a, smoothed probabilities in Figure 3.3b and the most probable state sequence in Figure 3.3c. When comparing the lines representing the reference values, with the points, the HMM seems to perform well.

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Reference</th>
<th>(a) Filtered probabilities</th>
<th>(b) Smoothed probabilities</th>
<th>(c) Most probable state</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>352</td>
<td>363</td>
<td>363</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>126</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>18</td>
<td>624</td>
<td>624</td>
</tr>
</tbody>
</table>

$ACC = 0.87$  

$ACC = 0.88$  

$ACC = 0.88$

1.10% of the iterations ended with a divergence. None of the iterations saturated the maximum tree depth of 12. In Appendix C.2, $\hat{R}$ and ESS are presented. There are no problems with divergence according to $\hat{R}$ and ESS. See Appendix C.3 for mean posterior estimates and standard deviations.
Figure 3.3 Performance of HMM based on simulated data and filtered/smoothed probabilities and most probable state path
4 Results

The results of the different HMMs are presented in the following sections. There is one separate HMM for each of the 10 sensors. Not all details for every model are presented, but some highlights are shown. The results are divided into different subsections, in order to separate weekday traffic flow from weekend traffic flow and also for highlighting the flow during event days. The convergence and model diagnostics of the NUTS are then presented.

4.1 HMM results for daily traffic flow conditions

In Table 4.1 a brief overview of how the HMMs classify the traffic flow conditions over the 14 days in total, according to the most probable state sequence, calculated by the Viterbi algorithm. It is shown that the HMMs for sensor IDs 194 and 196 classifies more states as congested, and less states as nightly free flow, compared with the rest of the HMMs classifications.

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Nightly free flow</th>
<th>Free flow</th>
<th>Mixture</th>
<th>Congestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>0.461</td>
<td>0.132</td>
<td>0.265</td>
<td>0.143</td>
</tr>
<tr>
<td>191</td>
<td>0.540</td>
<td>0.022</td>
<td>0.352</td>
<td>0.087</td>
</tr>
<tr>
<td>192</td>
<td>0.457</td>
<td>0.308</td>
<td>0.148</td>
<td>0.087</td>
</tr>
<tr>
<td>193</td>
<td>0.621</td>
<td>0.111</td>
<td>0.213</td>
<td>0.055</td>
</tr>
<tr>
<td>194</td>
<td>0.205</td>
<td>0.169</td>
<td>0.280</td>
<td>0.346</td>
</tr>
<tr>
<td>195</td>
<td>0.465</td>
<td>0.236</td>
<td>0.223</td>
<td>0.075</td>
</tr>
<tr>
<td>196</td>
<td>0.209</td>
<td>0.129</td>
<td>0.275</td>
<td>0.388</td>
</tr>
<tr>
<td>197</td>
<td>0.467</td>
<td>0.343</td>
<td>0.096</td>
<td>0.094</td>
</tr>
<tr>
<td>198</td>
<td>0.481</td>
<td>0.283</td>
<td>0.154</td>
<td>0.082</td>
</tr>
<tr>
<td>199</td>
<td>0.445</td>
<td>0.097</td>
<td>0.275</td>
<td>0.182</td>
</tr>
</tbody>
</table>

4.1.1 Weekday flow

In this subsection the weekday flows are presented. Since there are 10 weekdays in the data set it is not possible to show all days for all 10 models. A mean of the filtered and
smoothed probabilities of being in each state, for each sensor, are calculated to enable visualizations of the entirety of the information in the results. Also the inter-arrival times shown in these figures are means. The inter-arrival time sequences are presented on a logarithmic scale in the figures (the top plot in the figures), to respond to skewness towards large values; i.e., during the night where the inter-arrival times in general are much larger than the bulk of the data. The mean plots for all 10 HMMs can be seen in Figure 4.1-4.3. In general, the results are satisfactory and it seems like the HMMs can detect the different hidden states in a adequate way, even if there are small differences in the inter-arrival times. By looking at the plots it can be concluded that the nightly free flow state has the highest probability during the nights, which is a satisfactory outcome. Free flow and mixtures states are spread out during the day. According to the filtered and smoothed probabilities, the congested state seems to arise during morning and afternoon which is expected and is usually seen as 'rush hours'. For the HMMs for sensor IDs 194 and 196, Figure 4.2a and Figure 4.2c, it seems like it is a higher probability of being in a congested state during a longer period of the day. It can be seen in Table 3.1 that these two sensors recorded the most (inter-)arrivals during the period.

By taking the mean over 10 days can smooth out the effects of small changes in daily inter-arrivals. Therefore, the filtered, smoothed and also the most probable state sequence are presented for one certain day and two sensors, see Figure 4.4, to see how one arbitrarily chosen day can look like. Figures from separate days can also be seen in the next two subsections, with results from weekends and from a day with a known event at Tele2 Arena.
(a) Model for sensor ID 190 (in, right lane)  
(b) Model for sensor ID 191 (in, left lane)  
(c) Model for sensor ID 192 (out, left lane)  
(d) Model for sensor ID 193 (out, right lane)

Figure 4.1  Weekday mean of filtered and smoothed probabilities, west entrance/exit
(a) Model for sensor ID 194 (in, right lane)  

(b) Model for sensor ID 195 (in, left lane)  

(c) Model for sensor ID 196 (out, left lane)  

(d) Model for sensor ID 197 (out, right lane)  

Figure 4.2  Weekday mean of filtered and smoothed probabilities, south entrance/exit
By looking at the filtered and smoothed probabilities of being in each state, and by looking at the most probable state path, it can be concluded that the weekend flow in general are less congested. By looking at Figure 4.5, for the first weekend in the data, it can be seen that the probability of being in the congested state is higher at two peaks during
the Saturday afternoon. Note that this result is for sensor 198. For the second weekend
in the data set, the results for sensor 190 are presented, in Figure 4.6. Comparing these
weekend results with the average weekday results in Figure 4.3a, it can be concluded
that it seems to be less congestion during the weekends. In Appendix B.1, figures for the
weekend flow for model 191 and 194 can be seen. All of the results shown are adequate
and it seems like the HMMs can capture small differences in inter-arrival times.

4.1.3 Flow during event

During the soccer game 2019-04-01 19:00 at Tele2 Arena. The daily vehicle traffic flow for
the HMM for sensor ID 198 is presented in Figure 4.7. Sensor ID 198 registers bypassing
cars that are going in to the area, at the north entrance. It can be seen, both by looking
at the filtered and smoothed probabilities and the most probable state path, that there
is a peak for the probability of being in a congested state around 18:00 to 19:00. By
looking at the filtered and smoothed probabilities of being in congestion, and at the most
probable state path for the HMM for sensor 197, in Figure 4.8, it can be seen that the
probability of being in a congested state is higher around 21:00 to 22:00, which should
be after the soccer game. Sensor 197 captures cars that are driving out from the area,
 Driving in the right lane, at the south port from the area. In Appendix B.2, the filtered
and smoothed probabilities and the most probable state path are presented for this event
day, for the remaining 8 sensor IDs. Similar results can be seen for these sensors IDs as
for sensor 197 and 198.
Figure 4.5  Flow during first weekend in the data set
(a) Model for sensor ID 198 (in, single lane)

Most probable sequence of hidden states

(b) Model for sensor ID 198 (in, single lane)

Figure 4.6   Flow during second weekend in the data set
Filtered and Smoothed probabilities of being in each state

(a) Filtered and smoothed probabilities for sensor ID 198 (in, single lane)

(b) Most probable state path for sensor ID 198 (in, single lane)

Figure 4.7 Flow during event day: soccer game at Tele2 Arena at 19:00
Figure 4.8  Flow during event day: soccer game at Tele2 Arena at 19:00
4.1.4 MCMC convergence and model diagnostics

In Table 4.2, some diagnostics from the NUTS are presented. As seen, there are divergent iterations for all 10 models, but none of the models had any iteration that saturated the maximum tree depth of 12. The convergence criteria $\hat{R} < 1.01$ and ESS $> 400$ are fulfilled for models for sensors 191, 193, 194 and 196. In Appendix C.2, $\hat{R}$ and ESS are presented for all 10 models and parameters. Also, the estimated posterior mean and standard deviation can be seen in Appendix C.3.

Table 4.2 Summary of NUTS diagnostics

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Iterations saturated maximum tree depth, %</th>
<th>Iterations with divergence %</th>
<th>$\hat{R}$ criterion fulfilled</th>
<th>ESS criterion fulfilled</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>0.00</td>
<td>6.20%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>0.00</td>
<td>3.90%</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>192</td>
<td>0.00</td>
<td>3.98%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>0.00</td>
<td>3.43%</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>194</td>
<td>0.00</td>
<td>7.48%</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>195</td>
<td>0.00</td>
<td>3.25%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>0.00</td>
<td>3.25%</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>197</td>
<td>0.00</td>
<td>3.56%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>0.00</td>
<td>7.40%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>0.00</td>
<td>2.83%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Convergence diagnostics are also done by analysis of trace plots. Trace plots for the $\beta$’s for two of the HMMs, based on sensor ID 194-195, can be seen in Figure 4.9. In the left plot, Figure 4.9a, it is clear that the chain converged. In the the right plot, Figure 4.9b, either sequence alone looks stable. However, it is clear that the chains have not converged to the same distribution, which is concluded by the juxtaposition of the sequences. It is a demonstration of the need to use a between-to within-chain information measure as $\hat{R}$ when assessing convergence. In Appendix B.3, trace plots for all models can be seen. By looking at the trace plots, the same conclusions about converged models are drawn as by looking at the $\hat{R}$ and ESS criteria, presented in Table 4.2.
Two of the models that converged, HMMs for sensors 194 and 196, are also the two models that are based on the most data in total. Also, these two models are the two models that classify most of the time periods as congested.
5 Discussion

The overall performance of the HMMs is satisfactory, both during weekdays, weekends and event days. In this section it is discussed why the models has problems with divergence and also model improvements for future research are suggested.

One reason of why the models has trouble converging can be due to that there are not enough observations within the time intervals, or within some of the time intervals. When having little information in the data, the priors can play an important part. In this study the priors for the HMMs are informative and therefore affects the posterior in a powerful manner. When non-informative priors first was used, the models was not able to identify the different hidden states. Informative priors was therefore used to detect the hidden states.

Two of the models that converged (HMMs for sensors 194 and 196) seems to overestimate the presence of congested traffic. It can also be seen that these two sensors had the most bypassing cars during the period. However, this indicates that the HMMs based on data from sensors with more bypassing cars should be re-parametrized.

5.1 Model improvements and future research

A next step for the HMMs proposed in this study would be to do different adjustments to improve the convergence. For instance, the inter-arrival times may be modeled as e.g. Weibull distributed, instead of gamma distributed, to see if that increases the convergences. In further studies it would also be interesting to run the HMMs for more data than 14 days, which was not possible in this study due to the time complexity of the implemented algorithms in the HMMs.

In addition to this, one step in the process would be to investigate whether it is a robust method for other data sets with more dense traffic. In the present study, the method have only been tested on separate road spots where the traffic is not always very dense.

In further studies it could also be a possibility of trying the models with wider time intervals. Beyond the used 15 minutes intervals, the HMMs in this study was also tried out with 20 minutes time intervals, without satisfactory results. However, the returns versus losses for using wider time intervals must be considered. Wider time intervals
would give more information in the data for every time interval, but the information about
the current hidden state, based on a wider time interval, may not be as interesting as a
more narrow one.
6 Conclusion

The main research question in this study is to investigate how vehicle sensor data from an urban area can be used in Bayesian statistical models to estimate traffic flow conditions, in different directions and different places. This study concludes that it is possible to use Bayesian HMMs for estimating those traffic flow conditions.

The accuracy of the proposed method is difficult to determine, due to the unsupervised manner of the problematization of this study. A simulation study is implemented, which results in a high accuracy, which in turn indicates that the model choice is accurate. However, the obtained results of the HMMs based on real data are satisfactory for the different models and scenarios and it seems that the Hidden Markov Models can detect different traffic flow conditions, even for small changes in inter-arrival times. Some of the models had problems with divergence, but the obtained results from those models still showed adequate results. In fact, two of the models that converged (HMMs for sensors 194 and 196) seemed to overestimate the presence of congested traffic.

If the interest lies in only estimating the current traffic flow conditions based on earlier bypassing cars, the model choice of Bayesian HMMs is reasonable. If the interest lies in parameter inference as well, Bayesian HMMs might not be a first choice, since problems with convergence to the target posterior distributions arise.
References


Leos-Barajas, V. and T. Michelot (2018). “An Introduction to Animal Movement Modeling with Hidden Markov Models using Stan for Bayesian Inference”. In:


Xie, Y., K. Zhao, Y. Sun, and D. Chen (2010). “Gaussian Processes for Short-Term Traffic Volume Forecasting”. In: *Transportation Research Record* 2165.1, pp. 69–78.
A Appendices: Theoretical explanations

A.1 Viterbi algorithm

To calculate the most probable state path $q^*_{1:T}$, first define

$$\delta_n(j) := \max_{q_1,\ldots,q_{n-1}} p(y_{1:n-1}, q_n = j | y_{1:n}),$$

which is the probability of ending up in state $j$ at time $n$, given that the most probable path is taken. The most probable state sequence to state $j$ at time $n$ also consists of the most probable sequence to some other state $i$ at time $n-1$, followed by a transition from $i$ to $j$. Hence, $\delta_n(j)$ can be expressed by:

$$\delta_n(j) = \max_i \delta_{n-1}(i) a_{ij}^{(n)} p(y_n | q_n = j).$$

The most likely previous state, on the most probable sequence to $q_n = j$ is given by:

$$a_n(j) = \arg\max_i \delta_{n-1}(i) a_{ij}^{(n)} p(y_n | q_n = j).$$

The initialization is made by:

$$\delta_1(j) = \pi_q p(y_n | q_n = j).$$

The most probable final state is then calculated by:

$$q^*_N = \arg\max_i \delta_N(i).$$

The most probable sequence of states is then computed by using traceback:

$$q^*_n = a_{n+1}(q^*_{n+1})$$

(Murphy 2012).

A.2 Leapfrog method

The leapfrog method for HMC, mentioned in Section 2.3.1, works in the following way for the momentum variables $r$ and the parameter of interest $\theta$ (Neal 2011):

$$r(n + \epsilon/2) = r(n) - (\epsilon/2) \frac{\partial V}{\partial \theta}(\theta(n)),$$

$$\theta(n + \epsilon) = \theta(n) + \epsilon \frac{r(n + \epsilon/2)}{m},$$

$$r(n + \epsilon) = r(n + \epsilon/2) - (\epsilon/2) \frac{\partial V}{\partial \theta}(\theta(n + \epsilon)),$$

where $n$ is time, $\epsilon$ is the step size and $m$ is the mass of $\theta$. 

43
B Appendices: Figures

B.1 Weekend flow figures

Figure B.1 Smoothed and filtered probabilities first weekend, sensor IDs 191 and 194

Figure B.2 Most probable state path first weekend, sensor IDs 191 and 194
Figure B.3  Smoothed and filtered probabilities second weekend, sensor IDs 191 and 194

Figure B.4  Most probable state path second weekend, sensor IDs 191 and 194
B.2 Event day flow figures

(a) Model for sensor ID 190 (in, right lane)  
(b) Model for sensor ID 191 (in, left lane)

**Figure B.5** Smoothed and filtered probabilities for event day, sensor IDs 190 and 191

(a) Model for sensor ID 190 (in, right lane)  
(b) Model for sensor ID 191 (in, left lane)

**Figure B.6** Most probable state path for event day, sensor IDs 190 and 191
(a) Model for sensor ID 192 (out, left lane)   (b) Model for sensor ID 193 (out, right lane)

Figure B.7   Smoothed and filtered probabilities for event day, sensor IDs 192 and 193

(a) Model for sensor ID 192 (out, left lane)   (b) Model for sensor ID 193 (out, right lane)

Figure B.8   Most probable state path for event day, sensor IDs 192 and 193
(a) Model for sensor ID 194 (in, right lane)  
(b) Model for sensor ID 195 (in, left lane)

**Figure B.9** Smoothed and filtered probabilities for event day, sensor IDs 194 and 195

(a) Model for sensor ID 194 (in, right lane)  
(b) Model for sensor ID 195 (in, left lane)

**Figure B.10** Most probable state path for event day, sensor IDs 194 and 195
(a) Model for sensor ID 196 (out, left lane)  
(b) Model for sensor ID 199 (out, single lane)

**Figure B.11** Smoothed and filtered probabilities for event day, sensor IDs 196 and 199

(a) Model for sensor ID 196 (out, left lane)  
(b) Model for sensor ID 199 (out, single lane)

**Figure B.12** Most probable state path for event day, sensor IDs 196 and 199
B.3 Trace plots

Figure B.13  Trace plots for $\beta_1 - \beta_4$, sensor ID 190 – 191

(a) Model for sensor ID 190  
(b) Model for sensor ID 191

Figure B.14  Trace plots for $\beta_1 - \beta_4$, sensor ID 192 – 193

(a) Model for sensor ID 192  
(b) Model for sensor ID 193

Figure B.15  Trace plots for $\beta_1 - \beta_4$, sensor ID 196 – 197

(a) Model for sensor ID 196  
(b) Model for sensor ID 197
Figure B.16  Trace plots for $\beta_1 - \beta_4$, sensor ID 198 – 199
## C Appendices: Tables

### C.1 Accuracy measures by state, for simulation study

**Table C.1** Simulation study: Accuracy statistics by state for filtered probabilities

<table>
<thead>
<tr>
<th></th>
<th>$q_n = 1$</th>
<th>$q_n = 2$</th>
<th>$q_n = 3$</th>
<th>$q_n = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.95</td>
<td>0.82</td>
<td>0.57</td>
<td>0.88</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.99</td>
<td>0.96</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>0.97</td>
<td>0.89</td>
<td>0.74</td>
<td>0.92</td>
</tr>
</tbody>
</table>

**Table C.2** Simulation study: Accuracy statistics by state for smoothed probabilities

<table>
<thead>
<tr>
<th></th>
<th>$q_n = 1$</th>
<th>$q_n = 2$</th>
<th>$q_n = 3$</th>
<th>$q_n = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.98</td>
<td>0.87</td>
<td>0.58</td>
<td>0.88</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.99</td>
<td>0.97</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>0.99</td>
<td>0.92</td>
<td>0.75</td>
<td>0.92</td>
</tr>
</tbody>
</table>

**Table C.3** Simulation study: Accuracy statistics by state for the most probable state (sequence)

<table>
<thead>
<tr>
<th></th>
<th>$q_n = 1$</th>
<th>$q_n = 2$</th>
<th>$q_n = 3$</th>
<th>$q_n = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.98</td>
<td>0.86</td>
<td>0.58</td>
<td>0.87</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.99</td>
<td>0.97</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>0.99</td>
<td>0.91</td>
<td>0.75</td>
<td>0.92</td>
</tr>
</tbody>
</table>
### C.2 Convergence diagnostics: HMM for simulation study and real data study

<table>
<thead>
<tr>
<th>Parameter</th>
<th>ID 190</th>
<th>ID 191</th>
<th>ID 192</th>
<th>ID 193</th>
<th>ID 194</th>
<th>ID 195</th>
<th>ID 196</th>
<th>ID 197</th>
<th>ID 198</th>
<th>ID 199</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{R} )</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>ESS</td>
<td>1398</td>
<td>952</td>
<td>925</td>
<td>662</td>
<td>33.560</td>
<td>2.173</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

**Table C.4** Convergence diagnostics: HMM for simulation study and real data study

*Simulated represents the HMM for the simulated data, see Section 3.3.

*ID190 - ID 199* represents the HMMs based on data from respectively sensor ID

Simulated represents the HMM for the simulated data, see Section 3.3.
Table C.5  Posterior inference: HMM for simulation study and real data study

<table>
<thead>
<tr>
<th>Simulated ID</th>
<th>ID 190</th>
<th>ID 191</th>
<th>ID 192</th>
<th>ID 193</th>
<th>ID 194</th>
<th>ID 195</th>
<th>ID 196</th>
<th>ID 197</th>
<th>ID 198</th>
<th>ID 199</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.010</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.002</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.030</td>
<td>0.001</td>
<td>0.040</td>
<td>0.029</td>
<td>0.059</td>
<td>0.003</td>
<td>0.035</td>
<td>0.007</td>
<td>0.047</td>
<td>0.001</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.057</td>
<td>0.001</td>
<td>0.058</td>
<td>0.022</td>
<td>0.048</td>
<td>0.001</td>
<td>0.035</td>
<td>0.016</td>
<td>0.035</td>
<td>0.001</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.086</td>
<td>0.000</td>
<td>0.090</td>
<td>0.034</td>
<td>0.033</td>
<td>0.001</td>
<td>0.128</td>
<td>0.005</td>
<td>0.021</td>
<td>0.001</td>
</tr>
<tr>
<td>$\pi[1]$</td>
<td>0.357</td>
<td>0.211</td>
<td>0.400</td>
<td>0.196</td>
<td>0.408</td>
<td>0.203</td>
<td>0.402</td>
<td>0.197</td>
<td>0.396</td>
<td>0.201</td>
</tr>
<tr>
<td>$\pi[2]$</td>
<td>0.240</td>
<td>0.176</td>
<td>0.200</td>
<td>0.159</td>
<td>0.198</td>
<td>0.162</td>
<td>0.196</td>
<td>0.164</td>
<td>0.199</td>
<td>0.164</td>
</tr>
<tr>
<td>$\pi[3]$</td>
<td>0.199</td>
<td>0.155</td>
<td>0.199</td>
<td>0.161</td>
<td>0.199</td>
<td>0.154</td>
<td>0.199</td>
<td>0.154</td>
<td>0.197</td>
<td>0.154</td>
</tr>
<tr>
<td>$\pi[4]$</td>
<td>0.205</td>
<td>0.159</td>
<td>0.202</td>
<td>0.162</td>
<td>0.195</td>
<td>0.161</td>
<td>0.202</td>
<td>0.163</td>
<td>0.199</td>
<td>0.163</td>
</tr>
<tr>
<td>$A[1,1]$</td>
<td>0.723</td>
<td>0.024</td>
<td>0.962</td>
<td>0.009</td>
<td>0.954</td>
<td>0.008</td>
<td>0.935</td>
<td>0.012</td>
<td>0.962</td>
<td>0.007</td>
</tr>
<tr>
<td>$A[2,1]$</td>
<td>0.117</td>
<td>0.033</td>
<td>0.065</td>
<td>0.065</td>
<td>0.003</td>
<td>0.012</td>
<td>0.033</td>
<td>0.058</td>
<td>0.020</td>
<td>0.012</td>
</tr>
<tr>
<td>$A[3,1]$</td>
<td>0.042</td>
<td>0.034</td>
<td>0.043</td>
<td>0.076</td>
<td>0.002</td>
<td>0.003</td>
<td>0.217</td>
<td>0.130</td>
<td>0.014</td>
<td>0.008</td>
</tr>
<tr>
<td>$A[4,1]$</td>
<td>0.121</td>
<td>0.016</td>
<td>0.046</td>
<td>0.037</td>
<td>0.021</td>
<td>0.046</td>
<td>0.037</td>
<td>0.046</td>
<td>0.037</td>
<td>0.046</td>
</tr>
<tr>
<td>$A[1,2]$</td>
<td>0.213</td>
<td>0.028</td>
<td>0.021</td>
<td>0.021</td>
<td>0.000</td>
<td>0.000</td>
<td>0.016</td>
<td>0.025</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>$A[2,2]$</td>
<td>0.086</td>
<td>0.036</td>
<td>0.686</td>
<td>0.158</td>
<td>0.540</td>
<td>0.110</td>
<td>0.833</td>
<td>0.063</td>
<td>0.910</td>
<td>0.026</td>
</tr>
<tr>
<td>$A[3,2]$</td>
<td>0.144</td>
<td>0.044</td>
<td>0.051</td>
<td>0.054</td>
<td>0.032</td>
<td>0.011</td>
<td>0.133</td>
<td>0.035</td>
<td>0.036</td>
<td>0.012</td>
</tr>
<tr>
<td>$A[4,2]$</td>
<td>0.086</td>
<td>0.017</td>
<td>0.081</td>
<td>0.083</td>
<td>0.001</td>
<td>0.003</td>
<td>0.228</td>
<td>0.136</td>
<td>0.011</td>
<td>0.013</td>
</tr>
<tr>
<td>$A[1,3]$</td>
<td>0.051</td>
<td>0.027</td>
<td>0.010</td>
<td>0.017</td>
<td>0.006</td>
<td>0.008</td>
<td>0.083</td>
<td>0.013</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>$A[2,3]$</td>
<td>0.211</td>
<td>0.064</td>
<td>0.176</td>
<td>0.225</td>
<td>0.431</td>
<td>0.110</td>
<td>0.110</td>
<td>0.070</td>
<td>0.110</td>
<td>0.070</td>
</tr>
<tr>
<td>$A[3,3]$</td>
<td>0.140</td>
<td>0.075</td>
<td>0.843</td>
<td>0.096</td>
<td>0.912</td>
<td>0.016</td>
<td>0.627</td>
<td>0.124</td>
<td>0.896</td>
<td>0.020</td>
</tr>
<tr>
<td>$A[4,3]$</td>
<td>0.209</td>
<td>0.034</td>
<td>0.154</td>
<td>0.189</td>
<td>0.399</td>
<td>0.014</td>
<td>0.644</td>
<td>0.041</td>
<td>0.627</td>
<td>0.041</td>
</tr>
<tr>
<td>$A[1,4]$</td>
<td>0.074</td>
<td>0.044</td>
<td>0.073</td>
<td>0.073</td>
<td>0.014</td>
<td>0.040</td>
<td>0.086</td>
<td>0.030</td>
<td>0.086</td>
<td>0.030</td>
</tr>
<tr>
<td>$A[2,4]$</td>
<td>0.074</td>
<td>0.045</td>
<td>0.072</td>
<td>0.070</td>
<td>0.001</td>
<td>0.001</td>
<td>0.013</td>
<td>0.006</td>
<td>0.013</td>
<td>0.006</td>
</tr>
<tr>
<td>$A[3,4]$</td>
<td>0.074</td>
<td>0.047</td>
<td>0.072</td>
<td>0.050</td>
<td>0.004</td>
<td>0.004</td>
<td>0.013</td>
<td>0.004</td>
<td>0.013</td>
<td>0.004</td>
</tr>
<tr>
<td>$A[4,4]$</td>
<td>0.074</td>
<td>0.047</td>
<td>0.072</td>
<td>0.050</td>
<td>0.004</td>
<td>0.004</td>
<td>0.013</td>
<td>0.004</td>
<td>0.013</td>
<td>0.004</td>
</tr>
</tbody>
</table>

Simulated represents the HMM for the simulated data, see Section 3.3. ID 190 - ID 199 represents the HMMs based on data from respectively sensor ID