CHRISTOPH SCHNEIDER

CLASSIFICATION OF VIGILANCE USING HIDDEN MARKOV MODELS ON EOG DATA

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMARBEIT

CLASSIFICATION OF VIGILANCE USING HIDDEN MARKOV MODELS ON EOG DATA

carried out at the

Institute of Analysis and Scientific Computing
Faculty of Mathematics and Geoinformation
Vienna University of Technology

under the supervision of

Dipl.-Ing. Dr.techn. Johannes Kropf
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker

by

CHRISTOPH SCHNEIDER
Weyringergasse 25, 1040 Wien

May 2012

Christoph Schneider

Christoph Schneider: *Classification of Vigilance using Hidden Markov Models on EOG data,* Diploma Thesis in Technical Mathematics, © May 2012

SUPERVISORS:
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker
Dipl.-Ing. Dr.techn. Johannes Kropf

LOCATION:
Institute of Analysis and Scientific Computing
Vienna University of Technology

TIME:
June 2010 - May 2012

The life of a mathematician is dominated by an insatiable curiosity, a desire bordering on passion to solve the problems he is studying.

— Jean Dieudonne

In most sciences one generation tears down what another has built and what one has established another undoes. In mathematics alone each generations adds a new story to the old structure.

— Hermann Hankel

Mathematics is not a careful march down a well-cleared highway, but a journey into a strange wilderness, where the explorers often get lost. Rigour should be a signal to the historian that the maps have been made, and the real explorers have gone elsewhere.

— William S. Anglin

ABSTRACT

A lack of vigilance is nowadays one of the most frequent reasons for severe accidents, and this is even enhanced by our modern way of life. Hazardous situations can often be prevented when warning from a high level of fatigue. Therefore a Gaussian Hidden Markov Model (GHMM) was developed and implemented which takes electrooculography recordings (EOG) and additional car-based features of the SENSATION study in account to estimate the actual level of vigilance. It is shown that this is possible in the offline experiment for three coarse states - awake, neutral, sleepy. The results encourage to pursue that subject (with extended feature set) for the purpose of developing an online-monitoring system.

ZUSAMMENFASSUNG

Verminderte Aufmerksamkeit ist heutzutage einer der häufigsten Gründe für schwere Unfälle, und wird durch unseren modernen Lebenswandel begünstigt. Gefährliche Situationen können aber oftmals mit rechtzeitigen Warnungen entschärft werden. Deshalb wurde ein Gauss'sches Hidden Markov Model (GHMM) auf Basis von elektrookulographischen (EOG) und fahrzeugtechnischen Features entwickelt, welche im Zuge des SENSATION-Projekts aufgezeichnet wurden. Es wird gezeigt, dass es im Offline-Experiment möglich ist, eine grobe Aufteilung der Daten in sinnvolle Vigilanzniveaus (wach, neutral, müde) vorzunehmen. Die vorliegenden Resultate ermutigen weitere Forschungsarbeit auf diesem Gebiet, mit dem Ziel (mit erweiterter Feature-Menge) Online-Monitoring-Anwendungen zu konstruieren.

This is ten percent luck, twenty percent skill,
Fifteen percent concentrated power of will,
Five percent pleasure, fifty percent pain,
And a hundred percent reason to remember the name!

— Fort Minor

ACKNOWLEDGMENTS

Throughout my formation, currently cumulating in the creation of this work, I owe a lot to many different people, who I fear I will not be able to name all.

My greatest thank goes to my parents who supported me all my life, financially as well as emotionally, to defy the rough times we all encounter from time to time. I also owe a lot to my friends all over the world who spurred me on when my motivation was – as so often – at rock bottom.

I also want to thank all the people who taught me, from my math teachers in school up to my professors at the TU Wien. It has not always been easy getting along with you, but without you I would not be where I are now. At that place I also want to express my gratitude for all the scientists who laid the ground for my work as well as for the very active R-community, who provided me with suiting packages and answers to all of my questions.

At last one big ”thank you“ goes to my supervisor Johannes at the AIT, who opened his office to me in order to master the trickiest problems together.

I have the hope that this diploma thesis may prove to be helpful for some readers, giving me the opportunity to contribute to the scientific world whose accomplishments nourished myself so excellently.

CONTENTS

ACRONYMS

ACF Autocorrelation function

AIC Akaike information criterion

AICC Corrected Akaike information criterion

AIT Austrian Institute of Technology

BEM Bioelectromagnetism

BIC Bayesian information criterion

DC Direct current

DNA Deoxyribonucleic acid

EM Expectation-Maximization

EMG Electromyography

EOG Electrooculography

ESS Epworth Sleepiness Scale

FFT Fast Fourier Transformation

GHMM Gaussian Hidden Markov Model

GMM Gaussian Mixture Model

HMI Human machine interface

HMM Hidden Markov Model

IST Information society technologies

KDS Karolinska Drowsiness Score

KSS Karolinska Sleepiness Scale

LLH Log Likelihood

MC Markov chain

MP Markov property

NAN Not a Number

PDF Probability density function

Q-Q Quantile-Quantile (plot)

REM Rapid eye movement

RNA Ribonucleic acid

SEM Slow eye movement

SHMM Student's-t Hidden Markov Model

SVD Singular Value Decomposition

VTI Statens Väg- och Transportforskningsinstitut

Part I

INTRODUCTION

1 MOTIVATION

1.1 HYPOVIGILANCE AND ITS DANGERS

"Addressing Human Fatigue" is currently the number one problem on the most wanted list of the American National Transportation Safety Board. The Oxford Dictionaries define the word *fatigue* as

> "extreme tiredness resulting from mental or physical exertion or illness."

Another word, which is often used to describe the opposite is *vigilance*, which is given by the Oxford Dictionaries as

> "the action or state of keeping careful watch for possible danger or difficulties".

So the term *hypovigilance* refers to a state of diminished attention and reactivity bordering on fatigue, and it names a factor which is responsible for thousands of dead and a lot more injured people, not speaking about the economic loss it causes. It is not straightforward to put a number on the economic costs created by people's fatigue since one can think of many cases in which economic loss is thinkable e.g. goods destroyed, rehabilitation costs, loss of profits and revenues, health care for disabled people, etc. It would be necessary to determine a set of variables for a meaningful and comparable statistic but unfortunately such efforts have not been fruitful ever since.

So it has been up to several independent studies and estimations to give figures about the costs of hypovigilance. One of the most detailed sources to be found on that topic is the book "The twenty-four-hour society: understanding human limits in a world that never stops" by Moore [1]. In chapter 5 "The Costs of Human Breakdown" the author sums up economical losses to the society in four disjunct parts (extrapolated from U.S.-based statistics), see table 1.1. A detailed version with extensive explanations can be read in [1].

The figures in the book originate (in the most recent case) from the year 1993. So for good comparison it is necessary to adjust it to the cumulative inflation since then, which is for the U.S. economy 58.95%. That leaves us today with a worldwide annually economic loss of 600+ billion dollars. This is a sum which is not negligible any more and therefore draws additional attention to that topic, apart from all the human and environmental tragedies arising from errors due to hypovigilance. Prominent examples which shook the world emotionally as well as economically are the oil spill of *Exxon Valdez* the

	World Total Cost in billion $
Accident costs	80
Productivity costs	267
Health care costs	30
Societal costs	cannot be determined
Total costs	377+

Table 1.1: Annual cost chart for fatigue-based effects on society and economy

catastrophe of *Chernobyl* and the severe incident at *Three Mile Island* [1].

1.2 A PROBLEM OF OUR TIMES

Apart from such big catastrophes, the domain which draws the most attention to that problem is traffic – be it on the road, on rails at the sea or in the air. Technical improvement together with better constructed roads have managed to make transport as save as it has never been before [2, 3, 4, 5]. But as weired as it sounds, these technical solutions which help us in achieving those reduction in accidents and fatalities also lead to another problem. The isolation of the vehicle pilot from its environment, e.g. through noise and shock absorption, and the take-over of routine work by the machine, e.g. cruise control, autopilot, cause the attention of the driver or pilot to decrease.

One of the most telling incidents which has happened in the nearer past was the Go! flight 1002 on February 13, 2008, operated by Mesa airlines. Due to fatigue and the monotonous task during autopilot flight path, both pilots dozed off leading to the fact that the airplane passed by its original destination. When the crew awoke they managed to return and land normally at their destination with no casualties [6].

Insofar it is possible to refer to hypovigilance and fatigue related accidents as a special problem of our times, even though those of course existed in every period of human history – but maybe not to that extent.

Many professions nowadays do not include physical activity – which is known to raise the vigilance and maintains it at a high level. On the contrary, there exist many jobs with an usually monotonous occupation but where it is nonetheless indispensable to be alert all the time, e.g. pilots, truck drivers, assembly-line workers, safety inspectors, lifeguards, etc. And in all that fields of work fatigue is a risk factor not to underestimate!

Because this risk has become more and more prominent over the last decades, there exist in the meantime a large pool of studies which examine the effects of hypovigilance in different fields of activity.

Horne and Reyner found out that up to 20% of all road accidents are caused by hypovigilance [7] and the National Transportation Safety Board discovered, that 30% of all truck crashes fatal to the driver are caused by fatigue [8]. The clear connection between pilot schedules and accidents is shown in [9] and a cautious estimate of 8% of fatigue related aviation accidents is stated in [10]. But the true figure is believed to be much higher.

Gaba et al. investigated the effect of extensive work schedules of clinicians on patient safety and also briefly discussed the financial aspects of a reduction of the working hours [11]. Sleepiness in nurse shifts is responsible for an augmentation of potential errors during the shift of 3.4% [12].

Similar studies can be found for many other professions too. In the end all those studies show the urge for the development of functional tools for vigilance surveillance, which warn drowsy people before they make a possible hazardous error.

1.3 STRUCTURE OF THE MASTER THESIS

PART 1 In chapter 2 a short explanation is given, where the project arose from, which goals we like to achieve with it and under which framework the results are utilized. In chapter 3 we discuss the physiological backgrounds of sleep and sleepiness, which processes are run in human bodies when getting fatigued and we will discuss some feasible countermeasures in order to prevent dangerous situations due to hypovigilance. Chapter 4 will provide a detailed introduction into Electrooculography (EOG), drawing a bow from anatomical and physiological properties of the eye, over bioelectromagnetism to the actual technical realization of the EOG device with the most prominent problems and solutions. In addition we focus on the role of EOG-devices in vigilance surveillance applications. Chapter 5 gives an overview of several existing methods of vigilance detection, which are based on a variety of different (bio)signals. This should provide information to roughly compare our EOG-based model with other approaches, with the advantages and disadvantages becoming tangible.

PART 2 Then the data processing, all mathematical tools and the implementation used to achieve the results are going to be explained in detail. Chapter 6 deals with the mathematical concept of Hidden Markov Models (HMMs), giving an introduction into discrete HMMs before extending the approach to continuous HMMs and other types of basis-distributions than Gaussian. There we will also discuss possibilities for model verification. In chapter 7 we will explain where the

data we use comes from, how it has been provided and how we processed it in order to obtain our input features for the model. Chapter 8 provides information about technical details in the implementation, challenges which have been met and solutions we have come up with.

PART 3 Here, in chapter 9 we eventually jump right into the setting of the parameters and the final results. We are showing detailed plots of many different characteristics of the output, like state sequences, information criteria, pseudo-residuals and comparative distribution plots. We will also take the time to discuss our findings in chapter 10. In the concluding chapter 11, we will sum up our achievements and drawbacks in the process of the model building and serve a general outlook into future extensions and possible ameliorations concerning the topic of EOG-based vigilance detection. Ultimately we will dare to take an outlook into the future of EOG-based medical devices.

APPENDIX In the appendix the commented source code of the implementation in R can be found, alongside with the detailed bibliography.

2 AIM OF THE PROJECT

This diploma thesis is done in the framework of the project "Classification of vigilance states based on the EOG and EEG" – a cooperation between the department of Biomedical Systems of the *Austrian Institute of Technology* (AIT), *The Siesta Group*, the *Medical University of Vienna* and the *Institut für Schlaf-Wach-Forschung* (ISWF). The whole venture is funded by the Austrian Research Promotion Agency (FFG).

The aim of this project is to develop a reliable model for a classification of vigilance states, based on two biosignals, the EOG and the EEG. To achieve that, the goal is to first develop two independent models – one based on EOG (*Electrooculography*), the other one on EEG (*Electroencephalography*) – and to compare the results respectively explain possible differences.

My work focuses on building a model on EOG-data solely, using a three channel EOG recorded in the course of the SENSATION project (see chapter 7). Therefore we decided on a (continuous) Hidden Markov Model, with the vigilance states being the hidden information and the EOG-signal the linked observations.

WHY HIDDEN MARKOV MODELS? Hidden Markov models have already gained widespread use in different fields of pattern recognition, which seems to be a good basis four our research topic. Larue et al. showed in [13] that first order Hidden Markov Models are sufficient to obtain a fairly good result on vigilance surveillance. However, they related their findings to the *Sensation Seeking Scale* and discovered that people with higher risk and sensation disposition show a larger drop when becoming drowsy as overall calmer subjects.

At this place one has to mention that the range from alertness to sleepiness cannot be easily separated in a number of distinctive patterns like sleep itself, but remains mostly a continuous spectrum. So one has to bear in mind that every classification in a number of "wakefulness states" is purely arbitrary, but which does not affect the classification itself, i.e. the question: "How drowsy is someone? " can be answered nevertheless. Since we will evaluate our model with the labeled data provided by the measurements of the Karolinska Drowsiness Scale (see section 7.2.1), we will either use also 10 different states, or a subset of them.

The advantages of using Hidden Markov Models are:

- Uses an iteratively adapting optimization algorithm
- No need to work with predefined states

- Easy interpretability of the yielded model and parameters

The disadvantages we could encounter in the future are:

- The states might not be clearly separable
- There is a probability to get stuck in a local optimum
- Large amounts of training data needed

Another research group in the project is studying the connection between brain wave patterns (EEG) and drowsiness whereas a third group is taking charge of designing a feasible, robust and non-disturbing sensor device for EOG measurement in everyday situations, which could be integrated in automatic monitoring devices. Therefore it has to be applicable by the users themselves, affecting them as little as possible in their everyday lives, be it work, travel or at sport.

In the long term, which is not part of my diploma thesis any more, the developed model should take its place in such *online*, i.e. real time operating, applications. The interest in simple to handle vigilance surveillance, classification and warning systems is big especially in the field of workplace safety, road safety and medicine. Thinkable areas of application are e.g.

- Alarm devices for drowsiness detection at operator's stands, for drivers and flight control personnel
- Test devices for quick vigilance checks, e.g. at traffic controls, shift starts or in medical practices
- Wearable monitoring devices which record over a long time period (24h+) in the natural environment of the user for
  - detecting, classifying and treating diseases like chronic fatigue syndrome, narcolepsy, attention deficit disorder or sleep deprivation
  - controlling effects and side effects of medications
  - measuring the impact of different workplace ergonomics on the productivity

We will discuss our results, the advantages and disadvantages, as well as their relevance for possible future applications in detail in part iii. In the following we will give an overview of the EOG and its basics and characteristics before closing the introductory part with other methods of vigilance classification.

3 PHYSIOLOGY OF HYPOVIGILANCE

When talking about Hypovigilance, which is defined as a reduced state of vigilance, there are many terms which are used synonymously to a large extend:

SLEEPINESS The state of being sleepy

FATIGUE Extreme tiredness resulting from mental or physical exertion

TIREDNESS The state of wishing for sleep or rest

DROWSINESS A feeling of being sleepy and lethargic

All the definitions above are taken from Oxford Dictionaries and, as can be seen easily, are largely self-referencing. So throughout this master thesis all those terms will be used in the same sense, which is to describe a physical state which in which a person is driven to sleep and therefore no longer capable of reacting adequately to certain stimuli. Even when distinguishing between different word definitions, the effects on the human body are the same [14].

3.1 WHY HUMANS GET SLEEPY

That humans must sleep in order to survive goes without saying, and the importance of sufficient restorative sleep for the well-being has been shown in many studies [15].

The urge to sleep is mostly dependent on the individual *circadian cycle*, which does depend primarily on the light-dark circle on earth. Generally this rhythm makes us sleepy twice every 24h-period – once in the nighttime and once approximately 12 hours later, in the afternoon [14]. This is due to an increase of the melatonin level in the blood, which leads to a lack of concentration, sleepiness and eventually the onset of sleep itself [16].

3.2 REASONS FOR WORKPLACE FATIGUE

When talking about workplace fatigue we mean sleepiness during working hours, but unfortunately the problem extends to commuting, which is mostly done by car. But why are so many people tired when working respectively on their way to and from work?

One of the most affected professionals are such who do shift work. This is because of varying timetables or night time shifts contradicting with the natural circadian cycle, forcing the body to be fully alert at a time where it wishes to sleep instead [17].

Another weighty factor is sleep deprivation. We can distinguish between *total sleep deprivation*, which is a complete abstinence of sleep e.g. in a 24h shift and *partial sleep deprivation*, which is the reduction of the total sleep time, e.g too short rests between several shifts. This can be due to excessive working hours, which still exist in e.g. medical professions with more than 70 hours a week, which can go up in individual cases to 100 hours [18]. Another possibility to be sleep deprived despite balanced working hours is suffering from a sleep disorder as e.g. insomnia, narcolepsy or the restless legs syndrome. People with medical conditions impairing the vigilance in daytime, are strongly advised to consult a physician and to undergo treatment, because some sleep disorders bear also severe health risks.

A third factor is the monotony of the work. Challenging tasks are usually met with high vigilance (unless severely sleep deprived) whereas dull and repetitive tasks promote inattention and sleepiness [19]. This should be considered when it comes to the design of workplace environments.

3.3 EFFECTIVE COUNTERMEASURES

One of the most obvious countermeasures to fatigue is of course sleep [17]. A measure which is more and more applied is the construction of well-balanced timetables with shifts of maximal 12 hours length [20]. This helps to reduce the total as well as partial sleep deprivation. In addition there exist also a lot of studies which prove the usefulness of recreational breaks during the shift, which should preferably be used for short naps [21, 22, 23].

Another countermeasure which is applied nearly automatically when possible is physical activity. Nearly everyone has already experienced the beneficial effect of a short walk to get focused again. Like summed up by van den Hurk in [17], the effect of exercise or any kind of movement achieve in rendering the test subjects alert again, but only for a very short amount of time. A more durable impact could be achieved when executing a durably physical work task, e.g. lumberjacks, brick layers, etc.. Also very interesting is the connection between regular physical activity and fatigue at work, although it is not very strong [24].

The intake of stimulating substances with food and drinks is also a popular method to battle sleepiness at work or while driving. The effectiveness of stimulants, especially of the group of *xanthines* like *caffeine* and *theobromine* are well explored [25, 26].

4

ELECTROOCULOGRAPHY

The *electrooculography* (EOG) is a biomedical signal source which is based on the principle of *bioelectromagnetism* (BEM). Therefore we want to start out with a short overview of history and how such signals are created and measured. Then we will go into detail explaining the setup and physiological phenomena detected in the EOG before speaking about its applications in scientific research nowadays – especially in vigilance detection and classification.

4.1 HISTORY

At the turn of the 19th century Galvani and Volta experimented with the effect of electrical current on living organisms [27, 28], which lead to great leaps in the understanding of electromagnetic stimulation. One big step forward was the year 1865, when Maxwell published his work on the now famous *Maxwell Equations*, describing the inextricably connection between electricity and magnetism in a set of differential equations [29]. That lead to the understanding how to obtain and interpret electromagnetic signals coming from electrical currents in inner-body organs. But due to the extremely low amplitudes of body-generated biosignals it took again some time until those bioelectromagnetic fields could be recorded sensibly.

One of the pioneers in recording electric body signals was future Nobel Prize winnner Einthoven, who used the *string galvanometer* (originally invented by Ader) to derive the electrical field generated by heart cell excitation. The *electrocardiography* (ECG) was born [30].

Since then many diagnostic and therapeutic methods based on BEM have been developed. Among them are *electroencephalography* (EEG) to measure the electric potentials in the brain, *electromyography* (EMG) to measure the electric excitation of muscles and *magnetic resonance imaging* (MRI) to obtain 3D pictures of body tissues. On the other hand also many therapeutic devices use the knowledge of electric and magnetic influences on the body, as *cardiac pacemakers*, *defibrillators* and *deep brain stimulation* devices to aid people with Parkinson's disease.

4.2 BIOELECTROMAGNETISM

Malmivuo et al. give a general definition of Bioelectromagnetism as follows:

> " Bioelectromagnetism is a discipline that examines the electric, electromagnetic, and magnetic phenomena which arise in biological tissues. " [31]

This quote already reveals that the study of BEM is a vast interdisciplinary field between physics and electrical engineering up to (cell) biology and medicine. But everything starts out at the smallest living unit in the human body – the cell. Every body cell uses its membrane as a controllable barrier to keep a certain equilibrium of ions inside and outside itself. These ions could be charged positively, e.g. *Sodium* (Na^+) or negatively, e.g. *Potassium* (K^-). The difference in electrical charges inside and outside the cell causes an electrical voltage and a corresponding bioelectromagnetic field [31].

But since the large part of biomedical sensors recording bioelectromagnetic body signals are applied on the skin not penetrating into the tissue of interest, i.e. they are non-invasive sensors, they are therefore not able to sense the tiny voltages of single cell potentials, but only of larger united cell structures.

In general the detection quality of the such a body signal depends on:

- The electrical and magnetic field intensity generated by the source
- The conductivity of the tissues between the source and the sensor
- The distance between the source and the sensor
- The amplitude of the signal
- The amount of artifacts caused either by other electric fields in the body or by the sensing equipment

In the following we will go into detail about how to manage such problems, especially for the EOG.

4.3 PHYSIOLOGY OF THE EYE

The electrooculogram is the measurement of the movement of the eyes. The eyes are the visual sensing organ of humans and are located in the *viscerocranium* – the facial part of the skull. The anatomy of the eye is depicted in the schematic figure 4.1. We are only taking interest in the fact, that the *cornea* at the front side of the eye is charged relatively positive, whereas the *retina* on the back side of the eye is charged relatively negative. This originates from the higher metabolism rate due to the hyper-polarizations and depolarizations of the nervous cells in the retina [31, 32].

So the generated electric potential field can be described as a fixed dipole resulting in an easy and robust mathematical model. So when

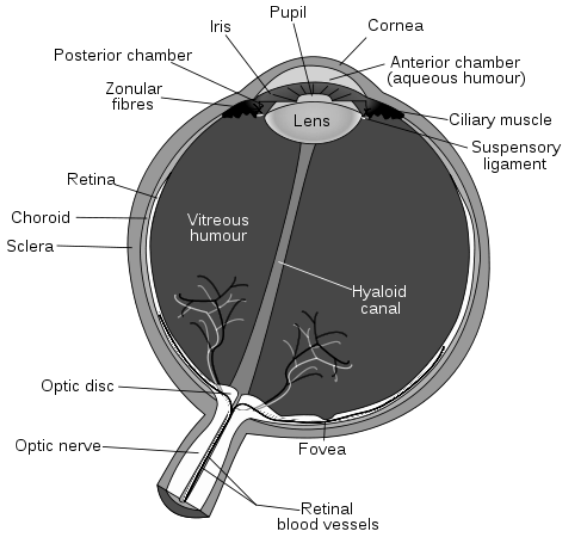


Figure 4.1: Anatomy of the human eye, horizontal section. (Picture taken from *Wikimedia Commons*; "File:Schematic diagram of the human eye en.svg")

the eyeball moves, the dipole field moves along with it, causing different potentials around the eye.

4.4 TECHNICAL REALIZATION

To obtain a usable EOG signal, one needs a setup with electrodes, cables, an amplifier and some signal processing on the computer. We will briefly talk about all that steps and the major obstacles in EOG recording.

4.4.1 *Electrodes*

To record the EOG signal, which is transmitted through the body tissues, a good skin contact with low impedance is necessary. The leading technique is to use Ag/AgCl hydrogel electrodes, which are also standard equipment in EEG recordings [33]. The advantage is the low impedance thanks to the hydrogel. Meanwhile also self adhesive Ag/AgCl electrodes are available, making the process of electrode preparation as easy as possible. According to [34] at least two channels have to be recorded to be able to cancel out noise from other biopotentials like EMG and EEG.

When talking about electrode positioning, there seem to be various methods described in the literature – mainly fitted to the respective purpose [35, 36, 37]. In general one could distinguish between two major classes of electrode positioning (see figure 4.2).

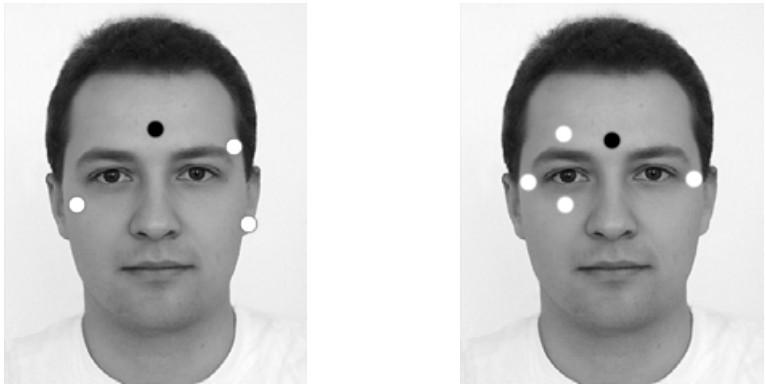


Figure 4.2: Channel electrode positions marked by white dots. Ground electrode marked by black dots. Left picture: positioning according to Rechtschaffen & Kales. Right picture: positioning for awake tasks – seperate horizontal and vertical channels. Portrait picture taken from C. Braun, M. Gründl, C. Marberger, and C. Scherber, "Beautycheck - Ursachen und Folgen von Attraktivität. Projektabschlussbericht," 2001.

The first positioning was introduced by Rechtschaffen and Kales for sleep research:

> "The recommended procedure is to record on one channel the potentials from an electrode approximately 1 cm above and slightly lateral to the canthus of one eye and a reference electrode on either homolateral ear lobe or mastoid. On the second eye movement channel are recorded the potentials from an electrode 1 cm below and slightly lateral to the outer canthus of the eye referred to the contralateral ear or mastoid, i.e. both eyes are referred to the same ear or mastoid electrode. " [34]

When this method is used nowadays, one usually uses also one ground electrode on the forehead for noise cancellation purposes. The electrode placement is depicted in the left picture in figure 4.2.

The second way of electrode positioning is normally used for the recording of eye movements in an awake state of the subject. Therefore it is necessary to be able to distinguish between vertical and horizontal eye movements. This is achieved by placing two electrodes in a vertical line around one (or both) eyes, usually below the eye and above the eyebrow. This could be also done for both eyes seperately. The horizontal signal is recorded from two electrodes either placed outside the lateral canthi of the eyes, or only around one eye with the second electrode being placed on the side of the nose. Since eye movements are coupled, the most common – because most convenient – method is the position depicted in the right picture in figure 4.2. The ground electrode is again placed at the forehead.

4.4.2 *Amplification and Processing*

The electrodes are connected via shielded cables to an appropriate amplifier for biosignals. The input signal frequencies range from DC - 100Hz, the signal amplitudes typically range from 10-1000μV (although information in literature varies [39, 32, 40, 33, 31]). Since the EOG signal is derived from the dipole of the eye, the signum and the amplitude of the signal are proportional to the displacement of the eye ball from the neutral position. For the horizontal channel that means: in the neutral position the system is calibrated to give a (nearly) zero amplitude. The more the eye ball is turned to the right the higher is the signal amplitude, the more it is turned to the left, the smaller (see figure 4.3). The same principle is analogously applicable to the vertical channel.

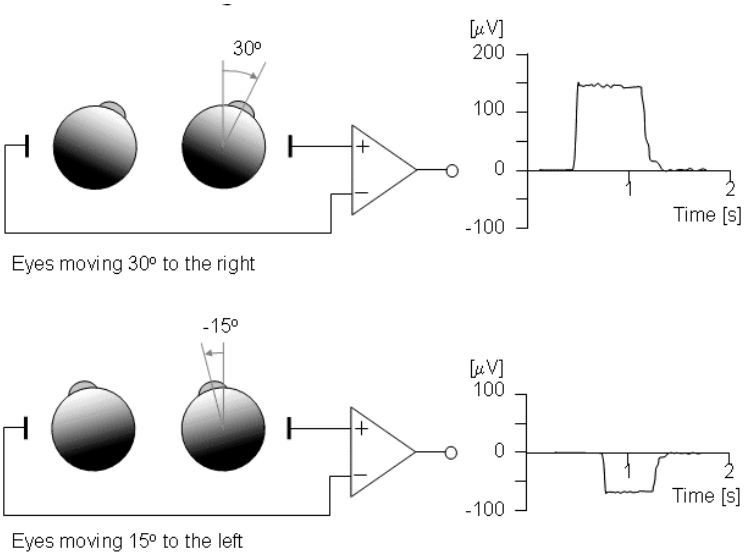


Figure 4.3: Schematic EOG signal generation, taken from J. Malmivuo and R. Plonsey, *Bioelectromagnetism : principles and applications of bioelectric and biomagnetic fields.* New York : Oxford University Press, 1995.

After signal amplification, it is necessary to process the signal in order to obtain the desired frequency range and to remove artifacts. Up to date biosignal amplifier already use hardware-implemented filters to pick the right frequency band of interest, get rid of DC-drifts and 50 Hz power line noise. A block diagram of such an amplifier built by Usakli et al. [36] is shown in figure 4.4. As frequency band of interest, the literature defines frequencies from DC, i.e. 0 Hz, up to 100 Hz, but most of the published articles have declared the range from DC to 30 Hz as feasible window [41].

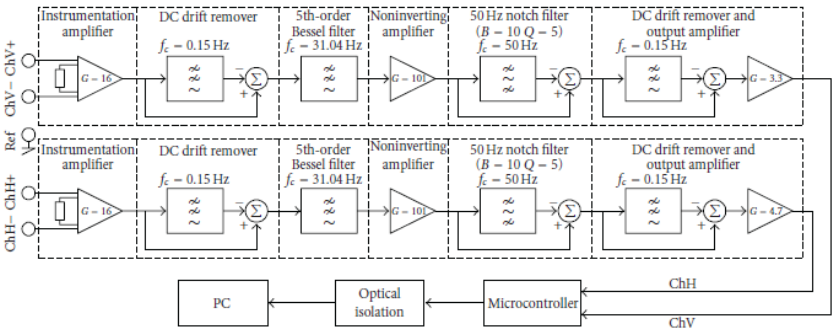


Figure 4.4: Block diagram for EOG data processing, taken from A. B. Usakli, S. Gurkan, F. Aloise, G. Vecchiato, and F. Babiloni, "On the use of electrooculogram for efficient human computer interfaces," *Computational Intelligence and Neuroscience*, vol. 2010, 2010.

4.4.3 *Artifacts*

As all biosignals, also the EOG signal is disturbed by a number of different sources, causing artifacts in the signal, some of them so severe, that one has to take care of them before using the signal for further computation. In regular research conditions one emphasizes *artifact avoidance*, e.g. instructions of the test subjects to not blink, move, etc. But this is not possible in many application conditions, like the one we want to investigate later on. Since we use data from a driving simulator study, it is perfectly normal and necessary for the test persons to blink and move their head while driving. In the following we mention the most common artifact sources, and how to possibly suppress their influence.

POWER LINE INTERFERENCE Electricity is distributed in electrical networks through power lines which we can find everywhere around us. Those power lines create electromagnetic fields which couple into literally everything, e.g. electrodes, cables and even the human body. In addition, all electrical equipment which is plugged in receives the same frequency directly over the power cable. This frequency is either 50 Hz (standard in Europe) or 60 Hz. Since it takes great efforts to cancel out all effects of power line interference, it is more useful to rely on 50/60 Hz notch filters to eliminate the signal distortion. Most of the amplifiers come already equipped with such analog notch filters.

BASE-LINE DRIFT Since the signal frequencies of interest range approximately form DC to 30 Hz [41] it is advisable to use a *direct coupled amplifier* (DC amp) in order not to distort or lose the low frequency components. On the other hand this brings also the disadvantage of a so called *baseline drift*, i.e. the signal voltage changes slowly over time. This can be due to manifold rea-

sons like temperature changes, changes of electrode impedance and/or skin resistance. Unfortunately it is not possible to filter those drifts, so it is left over for software algorithms for baseline drift correction [40, 42]. This has to be done in order to allow threshold based classifiers to perform properly. Another method is introduced by Yagi [43], which uses re-calibration in regular time intervals. This is not applicable to a vigilance surveillance tool, e.g. while driving.

ELECTRODE CROSSTALK The relative proximities of the different electrodes and the two eyes to each other are likely to produce *electrode crosstalk*, i.e. excitation of nearby electrode through electric potentials. The electrode crosstalk can be responsible for up to 54% of the recorded potentials, depending on electrode position, eye rotation angle and the individual [44]. Therefore electrode crosstalk cannot be ignored and has to be encountered with attenuation algorithms. Shinomiya et al. have done basic research on what has been undertaken to solve that problem, and made in-depth studies coming up with a simple yet effective solution [44].

EMG SIGNALS The human body as a connected system produces a lot of different electric biosignals like ECG and EMG. The bigger the muscles involved in the movement, the larger is also the produced bioelectrical field. That means that the potential recorded at the EOG electrodes is influenced by EMG artifacts, depending on their strength and proximity. This means that small facial muscles around the eyes do distort the signal as well as movements of the farther but larger neck muscles. Since it is not applicable in our case to prevent the probands from moving, we have to deal with the artifacts. Fortunately is the main frequency band of EMG signals 20-200 Hz with a peak in frequency power around 60-80 Hz [45, 46]. Since we are only interested in frequencies below 30 Hz for the EOG, we do get rid of the major part of the EMG artifacts by high pass filtering the signal (see figure 4.4).

BLINKS Blinking is a natural reflex behavior of all humans to keep the cornea moist and protect the eye from potential hazards. Adult blinking rate varies between 4.5 to 26 blinks per minute in healthy subjects, depending on the activity, with an average of 17 blinks per minute [47]. When using the EOG as a long term surveillance signal, it is impossible to advise the subjects to not blink, so it is necessary to detect and filter out distortions due to blinking. In offline studies manual blink detection and removal is done, but this is not suitable for online applications. To automatize this procedure many different approaches have been found and implemented. When blinking the eyeballs shoot

upwards, leading to a sharp edged dislocation of the EOG signal. Merino et al. employ simple time thresholding of the whole blink duration for detection [39] whereas Venkataramanan et al. use the derivation of the signal, i.e. steepness of the ascend, to differentiate between blinks and intentional upward movements of the eyes [32]. A quite different approach is described by Reddy et al., using *empirical mode decomposition* [48].

Since it is not possible to perfectly clean the EOG signal from all artifacts, it is advisable to have some artifact rejection mechanisms implemented, which automatically exclude disturbed sections of the signal from the classification process. This is especially important for applications which base decisions on short time periods, e.g. human-computer interfaces.

4.5 APPLICATIONS

The first widespread use of EOG signals, which also made them famous, was in sleep research, where they recorded the *rapid eye movement* (REM) phases while dreaming. Since then the EOG signal has grown more and more popular due to its easy recording and robustness and is used for example in following application fields:

- In sleep labors it is still used for REM-sleep detection and other measures [49].
- Oculomotor abnormalities like nystagmus, strabismus and supranuclear oculomotor dysfunction can be diagnosed [50, 51].
- The already noted linearity of the EOG signal to the angle of rotation of the eyeball makes the EOG, next to camera based systems, suitable for vision tracking systems [52, 53].
- It is frequently recorded alongside with EEG signals, helping to identify blink and eye movement artifacts in the EEG signal [54].
- It has also grown more and more popular as standalone source for *human machine interfaces* (HMI), where the EOG signal is used to control a machine, e.g. a screen keyboard or a wheelchair to aid handicapped people [36].
- It is possible to used several parameters derived from the EOG signal for vigilance classification (see next section 4.6) and even for activity recognition [42].

4.6 VIGILANCE DETECTION

We talk about different systems which are used for vigilance detection in chapter 5. Now we want to concentrate on EOG-based methods to detect phases of drowsiness.

EOG signals recorded from awake subjects normally consist of a horizontal and a vertical channel to be able to differentiate the two axis of movement, as well as blinks. A whole variety of so called *features*, i.e. heuristic measurable properties, can be derived from the two channels, e.g. frequencies, velocities, amplitudes. Some of them have been proven to be suitable for automatic drowsiness detection algorithms, which we want to discuss in the following.

4.6.1 *Blinks*

The by far most used EOG feature in vigilance classification is the blink. This may astonish, since we talked about blinks as mere artifacts in section 4.4.3. But different to applications which use the EOG to track the vision angle or communication with a computer system, blinking – as a natural reflex – reveals also the state of vigilance of a person. Since it is not feasible to manually detect blinks, algorithms have been developed which do that task successfully [55]. Once one has located the blinks, it is easy to derive the:

BLINK RATE As mentioned before, the blink rate in a healthy adult human being ranges between 4.5 to 26 blinks per minute – depending on the task. Mainly visual tasks which afford high concentration as reading or watching a movie highly decrease the blink rate. When the attention fades, i.e. the person gets tired, the blink rate increases [56].

BLINK DURATION The blink duration, i.e. the time the eyelid closes and opens again, is also a meaningful parameter for vigilance classification [56, 55]. Slower eyelid movements are an indicator for fatigue. Apart from not being able to see and therefore react to visual stimuli while the eyelid covers the pupil, long eyelid closure times can lead to microsleep [57] – sometimes with fatal consequences.

BLINK AMPLITUDE Measuring the amplitude of the recorded blink in the EOG signal can also give information on the alertness of the subject. Under alert conditions, blinks are short and forceful. Drowsiness is then detected when blinks have smaller amplitudes, corresponding to partial eye closure [55, 58]. Nevertheless one has to be aware that long blinks frequently have very high amplitudes. So a classification algorithm should label long blinks or small blink amplitudes as a sign of drowsiness.

Please pay attention to the fact that the shape and frequency of blinks are highly individual. A system exploiting the above mentioned features has to be calibrated on the person before use!

4.6.2 *Slow eye movements*

Slow eye movements (SEM) have already been described in [34] during transition between wakefulness and sleep. It has been shown that in phases of diminished vigilance before sleep onset there is a high correlation between typical EEG patterns and SEMs [59], i.e. sleepy subjects have a much higher portion of SEMs than alert subjects.

A frequently used method to extract the low frequency waves of SEMs from the EOG is wavelet transformation, like described from Ma et al. in [60]. Their results are promising so that the detection of SEMs could advance to the second standard in EOG-based vigilance detection (next to blink related algorithms).

4.6.3 *Amplitude, Velocity, Frequency*

Hanke et al. tried a different method for vigilance detection, using EOG amplitude, end velocity – the first derivation of the signal – and the frequency – measured in two bands – to deduce fatigue in test persons [61]. They carried out the *Mackworth clock test* on 10 subjects, showing distinct changes of the selected features over time. Due to the unsure correlation parameters reaction time or mistakes, it is not easy to qualify the results. But there seems to be a certain correlation of the ratio between high and low frequency part of the EOG signal. This could possibly be due to SEMs.

5

OTHER APPROACHES FOR VIGILANCE DETECTION

In the last chapter we learned about the EOG and how we can use its characteristics in vigilance surveillance. But it does not come surprising, that also other parameters have been used for this task. There exist literally hundreds of studies which have investigated the connection between different physiological measures – invasive and non-invasive – and attention. In the following we want to give a quick overview over the most promising vigilance detection systems besides the EOG.

5.1 EEG

The EEG is – as it refers directly to the brain wave patterns – probably the most accurate biosignal attainable when it comes to a classification of vigilance states.

Minkwitz et al. [62] investigated the difference in the EEG-signal between drowsiness and a relaxed state and evaluated it with the reaction time in test settings. One of the main fields of interest in vigilance detection – drowsy driving – has of course also been center of interest, e.g. in [63, 64]. In [63] They used continuous wavelet transform of the signal and a support vector machine classifier to distinguish between three different states of vigilance. Validating the data by video analysis of the drivers in the simulator, they achieved a classification accuracy of 96%.

Another way was gone by Coufal [64], who used fuzzy system models. Those have the advantage of a small parameter set, and also yield accuracies of nearly 80%.

However, one should not forget that the recording of EEG-signals is cumbersome and the devices are considered very inconvenient by the test subjects. So despite all of the qualified work at that field, it does not seem likely for EEG-based systems to find their way in our everyday lives in the near future.

5.2 EMG

The EMG is one of the oldest bioelectrical signals derived in medical science. But it seems difficult to obtain reliable results in vigilance detection with EMG alone. Nonetheless the EMG is widely used as a signal source to filter out artifacts from EEG signals [65, 66], and it has also been shown that implementing the EMG pattern into the

vigilance classification algorithm, the detection accuracy was increased in comparison to EEG alone [67, 68].

5.3 CAMERAS

Since the uprise of pattern recognition in image processing, cameras have been used to fulfill different automatized tasks, e.g. face recognition, eye tracking, activity recognition, etc. [69, 70, 71].

Camera-based systems allow to detect a whole range of visual parameters which can be also used for vigilance detection.Those are for example facial expression, head position, eyelid movement and gaze movement [72]. Most attention in research lies on the eyes, determining features which can be well extracted like eyelid closure time and PERCLOS, the percentage of eyelid closure over the pupil [73, 74]. Actual detection systems yield an accuracy of 90% [74]. For further technical details on this field of vigilance surveillance I strongly recommend the literature of Ji et al. [72].

The advantages of camera based systems are quite obvious: they can track multiple parameters at once which helps to yield better classification results. In addition the systems are contact-free, sparing the users from possible impairments due to the sensors. The disadvantages are the need for a camera mounting allowing to keep track with the subject all the time, dependency on sufficient lighting (which is solved e.g. in cars with active near infrared diodes) and the merely inability to construct a body-mounted device for camera-based vigilance classification.

At the state of the art it seems, that vision based vigilance detection systems is going to be the predominant method in all fixed working environments, e.g. air traffic control towers, operators stands and all sort of drivers cabins.

5.4 PUPILLOGRAPHY

The instrument of pupillography, relies on the visual detection of the pupil with an infrared-based camera system. The testing scenario usually consists of recording infrared images from the eyes for 10 minutes, preferably in a dark, quiet environment [75]. Pupil behavior between alert and sleepy persons differs in this task. Pupils in alert people stay dilated and oscillate with high frequency and low amplitude, whereas these factors are reversed for fatigued people, which results in a very clear distinction between alert and drowsy subjects [75, 76].

So far, pupillography is implemented as a medical test (e.g. for narcolepsy patients) and takes full attention over the whole 10 minutes interval, preferably in a dark surrounding and eyes wide open. Although Deng et al. [77] have overcome the problem of partial eyelid closure with refined estimating algorithms, the whole package still is

not fitted for an online vigilance detection system which can be used to monitor people in their working environment.

5.5 SKIN CONDUCTANCE

Skin conductance is a very old and simple biosignal, measuring the conductivity of the human skin, influenced mostly by sweating which is controlled by the central nervous system. It was and is studied mainly in psychology to measure body responses in different situations. There has been some research to align skin conductance and alertness, i.e. reaction time on tests, but with ambiguous results [78].

Boucsein et al. [79] showed, that when combined with other bioparameters – they chose the heart rate variability – a quite acceptable vigilance detection can be achieved. The huge advantage of such a system is the easy recording of the signals, which can be done with two contact electrodes, e.g. implemented into the yoke of aircrafts.

5.6 NON-BODY-RELATED SENSORS

Nowadays many automotive companies invest in the research of attention classification systems integrated in their products with the aim of advising the driver to take a rest before his drowsiness signs become too alarming.

In [80] Mercedes Benz compared their already developed system based on driving lane data (position of the car relative to the lane, recorded by cameras) to odometric data, i.e. inertial sensors, steering wheel parameters. It has been shown that especially the camera based systems provide a good approximation on the subjectively rated Karolinska Sleepiness Scale.

A similar approach has been used by the Ford company in their fatigued driver detection system *Driver Alert*. They also rely on camera based pattern recognition algorithms to analyze the trajectory the car is taking in its lane [81]. A conspicuous behavior will lead to warnings indicating a rest.

The advantages of such systems are that it feels more comfortable for the user to not be directly cabled into the vigilance measure system. On the other hand, those systems often lack accuracy and can be used only in the machine they are built in.

5.7 INTEGRATED SYSTEMS

Like stated in the above points, some biosignals have been proven to be good estimators of the overall alertness level of humans. But in order to create more robust and exact models for vigilance classification, a multiparametrical approach seems to be most promising, i.e. the system takes different biosignals as input as e.g. EEG, EOG,

camera systems as well as data from optional environment sensors [82]. Nevertheless will a profound research on individual input signals, such as the EOG, increase the accuracy rate of integrated systems too.

Part II

METHODS

6 HIDDEN MARKOV MODELS

The following chapter is dealing with the theoretical background of *Hidden Markov Models* (HMMs) and is based on the work of Rabiner [83] (notably section 6.1 and 6.2) who uses Hidden Markov Models for speech recognition applications. This article is fairly popular for an introduction into the functionality of Hidden Markov Models and the associated *expectation-maximization algorithm* (EM-algorithm) and is therefore the foundation of many research results concerning the application of these mathematical tools.

Since the first description of Hidden Markov Models in the 1960s, this method has spread over many different research areas and is nowadays present in fields like speech recognition [83], activity recognition [84, 85, 86] and gene sequence alignments [87, 88].

Generally speaking, Hidden Markov Models are stochastic (non-deterministic) models for processes which possess different distinctive states which are not known, and must therefore be guessed. How to proceed to make an educated guess, we will see in the following:

6.1 DISCRETE HIDDEN MARKOV MODELS

Discrete HMMs are the most straightforward form of the model and are therefore qualified as a presentation of the basic principles. They are nevertheless indispensable for applications with a narrow set of observable outputs, e.g. genome sequence alignment.

6.1.1 *From Markov Chains to Hidden Markov Models*

Prior to the theory of HMMs it is necessary to clarify the essential components of Hidden Markov Models.

A (discrete-time) *Markov Chain* (MC) is a set of (discrete) random variables $\mathbf{X} = \{X^t | t \in \mathbb{N}\}$ which fulfills the so called *Markov property* (MP):

$$P(X^{t+1}|X^t, \ldots, X^1) = P(X^{t+1}|X^t) \tag{6.1}$$

for all $t \in \mathbb{N}$ i.e. all future events depend only on the present one. To be more precisely, the above stated is called a first order Markov Chain, because the probabilities of future states depend only on one state – the present one.

This can be generalized to a Markov Chain of order q ([89]), which has to fulfill the Markov property

$$P(X^{t+1}|X^t,\dots,X^1) = P(X^{t+1}|X^t,\dots,X^{t-q}). \tag{6.2}$$

Such Markov chains of higher order are used in genomics and bioinformatics as well as in cryptology [90, 91]. Though giving a better connection to the previous time steps, Hidden Markov models which are based on higher order MCs are often complex to handle while the information gain is modest.

Therefore only first order Markov Models are used in the following theoretical discussion of Hidden Markov Models and for the vigilance classification application.

Now back to regarding the Markov Chain $\mathbf{X}$. At every discrete time step t, X^t yields a certain value $S_i \in \{S_1, S_2, \dots, S_n\}$ out of a finite n-set of different states. A fixed sequence of states produced by a Markov Chain $\mathbf{X}$ will be denoted as $\mathbf{S} = \{S^t | t \in \mathbb{N}\}$. Please take notice of the different font types of "S" which are used as well as the sub- and superscript indices to differentiate between a fixed state sequence $\mathbf{S}$, its elements S^t indicating the position in the *time domain* and the actual state symbol S_i indicating the according element of the discrete *state domain*.

Example: A Markov Chain $\mathbf{X} = X^1, X^2, X^3, X^4, X^5$ with the length of five may yield a sequence $\mathbf{S} = S^1, S^2, S^3, S^4, S^5 = S_2, S_2, S_1, S_2, S_1$ consisting of the two different states S_1 and S_2.

Considering the minimal example above it is easy to see, that there has to be a certain possibility for the states to change over time. The so called *state transition probability*

$$a_{ij} = P(X^{t+1} = S_j | X^t = S_i), \quad i, j \in \{1, \dots, n\} \tag{6.3}$$

is the probability to go from state S_i to S_j in one time step. Obviously all a_{ij} have to obey the standard stochastic constraints:

$$a_{ij} \geqslant 0 \quad \forall i, j \tag{6.4}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i. \tag{6.5}$$

To get a compact notation, the state transition probabilities could be written in a single $n \times n$-matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \tag{6.6}$$

If the state transition probabilities do not depend on the time t then the Markov Chain is called *homogeneous*. If not denoted otherwise, all

discussed Hidden Markov Models throughout the next chapters will be based on homogeneous Markov Chains.

To depict a vivid example, which should clarify how the state transitions interact with a Markov chain, let the states S_i correspond to three different weather situations :

$$\begin{aligned} S_1 &= \text{sunny} \\ S_2 &= \text{cloudy} \\ S_3 &= \text{rainy} \end{aligned}$$

Assuming that a meteorological station records the actual weather every day (at a fixed time, e.g. noon), then the emerging data could be interpreted as a three-state time discrete Markov Chain, which could be graphed as in figure 6.1. Hence the state transition matrix **A**

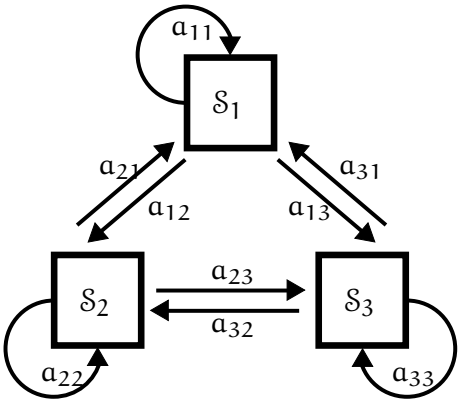


Figure 6.1: Three-state Markov Chain with transition probabilities

specifies the probabilities of a daily change in the weather. Filled with randomly chosen numbers **A** could look like:

$$\mathbf{A} = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.1 & 0.2 & 0.7 \\ 0.4 & 0.2 & 0.4 \end{pmatrix}$$

The state transition a_{11} denotes that, given a sunny day, the probability of an additional sunny day is 0.4. The probability of 3 sunny days in a row plus a rainy day plus a cloudy one would be: $(a_{11})^3 \cdot a_{13} \cdot a_{32} = (0.4)^3 \cdot 0.3 \cdot 0.2 = 0.00384$. So in general the probability of a certain state sequence **S**, generated by a Markov Chain **X**, can be calculated by

$$P(\mathbf{S}) = \prod_{t=1}^{T} a_{ij} \tag{6.7}$$

In the following the theory of Markov Chains is extended step by step to obtain the basics of Hidden Markov Models. These could

be seen as Markov Chains where at every time step t the stochastic variable X^t generates a certain state S^t which, on the other hand, produces an output O^t itself. This output is the only thing which is observable while the states themselves remain hidden! Hence one receives two linked sequences: one state sequence $\mathbf{S} = S^1, S^2, S^3, \ldots$ and an observation sequence $\mathbf{O} = O^1, O^2, O^3, \ldots$ like shown in figure 6.2.

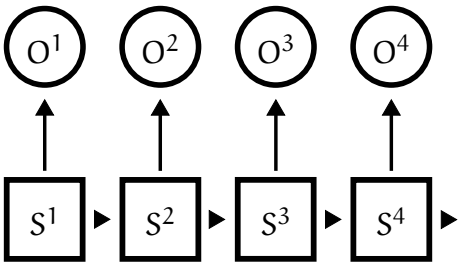


Figure 6.2: The two linked sequences of a HMM

Note that these sequences are finite in any application, so they could also be seen as vectors – and they will be treated as vectors in the modeling process later on.

Every element O^t of the observation sequence $\mathbf{O}$ refers (analogously to the state sequences) to a specific observation $\mathcal{O}_k \in \{\mathcal{O}_l | l = 1, \ldots, m\}$ where $m \in \mathbb{N}$. In a HMM with discrete observations, the number of distinct observations $\mathcal{O}_k$ normally is bounded by $m < \infty$. For further information on that topic see [92]. The different observations $\mathcal{O}_1, \ldots, \mathcal{O}_m$ are called the *discrete alphabet* (of observations) and m the *discrete alphabet size*. On purpose of better understanding, figure 6.2 is shown again with random values filled in. So figure 6.3 shows the distinct states and observations of such two random sequences.

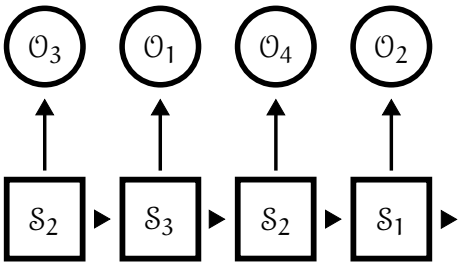


Figure 6.3: HMM with exemplary states and observations filled in

Every observation $\mathcal{O}_k$ occurs with a certain probability $b_i(\mathcal{O}_k)$ depending only on the current state $\mathcal{S}_i$

$$b_i(\mathcal{O}_k) = P(\mathcal{O}_k | \mathcal{S}_i) \tag{6.8}$$

Analogous to the state transition probabilities it comes also handy to write the occurrent $b_i(\mathcal{O}_k)$ in form of a $n \times m$-matrix

$$\mathbf{B} = \begin{pmatrix} b_1(\mathcal{O}_1) & \dots & b_1(\mathcal{O}_m) \\ \vdots & \ddots & \vdots \\ b_n(\mathcal{O}_1) & \dots & b_n(\mathcal{O}_m) \end{pmatrix} \tag{6.9}$$

Like for the a_{ij} there hold also the standard stochastic constraints:

$$b_i(\mathcal{O}_k) \geqslant 0 \quad \forall i, k \tag{6.10}$$

$$\sum_{k=1}^{n} b_i(\mathcal{O}_k) = 1 \quad \forall i. \tag{6.11}$$

Note: like the a_{ij} also the $b_i(\mathcal{O}_k)$ are regarded stationary, i.e. the observation probabilities do not change with time t but depend only on the actual state $\mathcal{S}_i$!

Now that we have defined that a HMM consists of a hidden state sequence and a visible observation sequence it is obvious that we do not know at which state to start the model. That is the reason why the *initial distribution vector* $\mathbf{I}$ is introduced, which indicates the probabilities of starting in a certain state.

$$\mathbf{I} = \begin{pmatrix} i_1 \\ \vdots \\ i_n \end{pmatrix} \tag{6.12}$$

Of course the initial distribution vector satisfies also the standard stochastic constraints

$$i_k \geqslant 0 \quad \forall k \tag{6.13}$$

$$\sum_{k=1}^{n} i_k = 1. \tag{6.14}$$

To get an idea of how HMMs work and why they have such a vast field of application we are going to take a look at the three models from [83], page 259, which are perfectly suited for that case: they are simple enough to be understood right away, but are nonetheless able to explain the main working principles and aims of Hidden Markov Models:

Two people, locally separated, are part of the same process. Person 1 performs a series of coin tossing experiments and Person 2 is told only the results (i.e. observations) – namely *heads* or *tails* – of each round. For example

$$\mathbf{O} = O^1, O^2, \dots, O^T = \mathrm{HHTTTTHTHH} \dots \mathrm{T}$$

Note that in this case we identified $\mathcal{O}_1 = \mathrm{H}$ and $\mathcal{O}_2 = \mathrm{T}$. Given this specific scenario above, the question arises how to explain respectively

model the observation sequence. Merely the first problem to encounter is to decide on the number of different states in the HMM. One has to keep in mind that the model complexity as well as the computing time increases dramatically with every new state added. So an additional state in a HMM has to bring considerable advantages in the model preciseness to be of any practical use!

In the following three different models (with one, two and three states) are discussed and additionally depicted in table 6.1.

ONE. A one state model would yield a single state transition probability $\mathbf{A} = a_{11} = 1$, because in every time step there will take place only the transition from the single state to itself. Additionally there are two different observations possible (alphabet size $= 2$), namely heads *(H)* $= \mathcal{O}_1$ and tails *(T)* $= \mathcal{O}_2$. Taken a "fair coin" the probability for each observation to occur is $b_1(\mathcal{O}_1) = b_1(\mathcal{O}_2) = 0.5$. On the other hand it could also be thinkable, that the coin is "unfair", i.e. biased, so that the probability distribution for heads/tails is for example $0.4/0.6$. Rabiner calls this one-state model "degenerated" because there is no real need for a concept of states and transition probabilities (there is nothing hidden, in this HMM). All results could also be obtained with traditional stochastic calculation.

TWO. A two state model consists of the states $\mathcal{S}_1$ and $\mathcal{S}_2$, a 2×2 state transition matrix $\mathbf{A}$ and a 2×2 observation probability matrix $\mathbf{B}$. Note, that the two states, which symbolize two different coins, could have two quite different biases – in fact, this is the only sensible reason for adding a new state. The state transition probabilities are another stochastic process which could be imagined for example as another, unrelated, coin tossing experiment.

THREE. Here the two-state model from above is again expanded by a new state. This yields the states $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ and the possible observations $\mathcal{O}_1, \mathcal{O}_2$ along with the 3×3-matrix $\mathbf{A}$ and the 3×2-matrix $\mathbf{B}$. Of course also this model is more reasonable with three quite differently biased coins.

Given this three different Hidden Markov Models for the coin tossing experiment one is naturally interested in the question, which one of these matches best the underlying unknown conditions. Gut feeling may tend to a model with more states because such models have plenty of undefined parameters, also called *degrees of freedom*, which can be tweaked and adjusted. In the one-state model there is only one parameter to determine: the observation probability (or bias) for heads $b_1(H)$ (because the observation probability of tails can be easily obtained through $b_1(T) = 1 - b_1(H)$), because they have to fulfill the stochastic constraints 6.11, i.e. they must add up to 1. In the three-state model on the other hand there are already nine

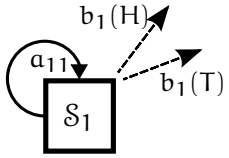
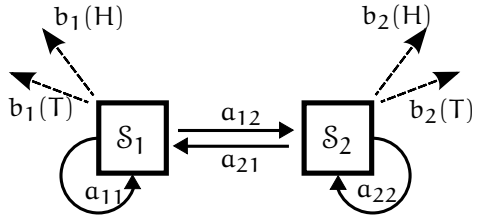
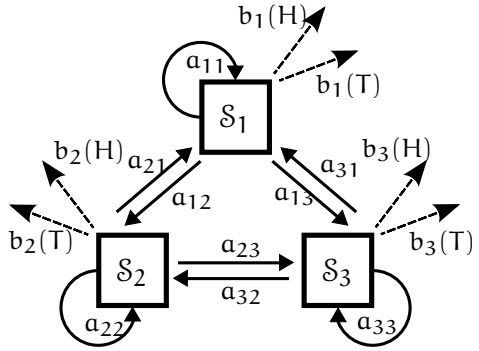
States	Graph of HMM	Exemplary Sequences
i		$\mathbf{O} = HHTTTTHTHH\ldots$ $\mathbf{S} = 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\ldots$
ii		$\mathbf{O} = HHTTTTHTHH\ldots$ $\mathbf{S} = 2\,2\,2\,1\,1\,2\,1\,1\,1\,2\ldots$
iii		$\mathbf{O} = HHTTTTHTHH\ldots$ $\mathbf{S} = 1\,3\,2\,2\,2\,1\,1\,3\,1\,3\ldots$

Table 6.1: Different Hidden Markov Models for the coin tossing experiment

unknown parameters to calculate – six transition probabilities and three observation probabilities. (The missing parameters result again from the stochastic constraints 6.5 and 6.11.)

Although there lies a truth in the way of thinking that complex models yield better results, this can quickly turn out to be a curse. Not only does the computational effort increase quadratically with every state added, but assuming that the real experiment is a one-coin-flip, a three (or more) state model will be objectively not accurate. A model which is more complicated than the real system is called *underspecified*. It will yield indefinitely many best solutions because there exist parameters which are not needed, and therefore their value does not matter.

So in the end, the aspired Hidden Markov Model is the one which will give the best explanation for the observed data using the least variables to determine!

6.1.2 *The Three Basic Problems for HMMs*

All elements needed to form a proper Hidden Markov Model were introduced throughout the last section. These are:

S_i	...	states of the model, $i \in [1, n]$
n	...	number of distinct states
$\mathbf{A}$	...	$n \times n$—matrix of state transition probabilities
$\mathbf{I}$	...	$n \times 1$—vector of the initial state distribution
O_k	...	observations of the model, $k \in [1, m]$
m	...	number of distinct observations
$\mathbf{B}$	...	$n \times m$—matrix of observation probabilities

The parameters needed to determine a Hidden Markov Model are therefore $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{I}$ and will be referred to as the parameter set $\Lambda = (\mathbf{A}, \mathbf{B}, \mathbf{I})$. Given such a set of parameters, the resulting model could work as a generator of possible observation sequences (of lenght T) in a very simple way:

1. Decide on an initial state $S^1 = S_i$ according to the initial state distribution $\mathbf{I}$
2. set time $t = 1$
3. Choose an observation $O^1 = O_k$ according to the observation probabilities for the state S_i, i.e. according to $b_i(O_k)$
4. Determine the next state $S^2 = S_j$ according to the state transition probabilities a_{ij}
5. increase time step t++
6. return to point 3 if $t < T$; else terminate sequence

This shows, that it is very simple to generate training sequences for well known models. Unfortunately that is not the case in everyday applications. In the by far most common case one gets one (or more) observation sequences but no hints whatsoever concerning the number of states or the parameters involved in the *true* underlying model. That leads to the *Three Basic Problems for HMMs*:

PROBLEM 1 How to compute the probability $P(\mathbf{O}|\Lambda)$ of a given observation sequence $\mathbf{O} = O^1, O^2, \ldots, O^T$ under a fixed parameter set $\Lambda = (\mathbf{A}, \mathbf{B}, \mathbf{I})$?

PROBLEM 2 How to chose the "best" state sequence $\mathbf{S} = S^1, S^2, \ldots, S^T$ for the given observation sequence $\mathbf{O}$ and the parameter set Λ, i.e. which state sequence does the best job in explaining the model regarding probability and simplicity?

PROBLEM 3 How to maximize the probability $P(\mathbf{O}|\Lambda)$ by adjusting the parameter set Λ?

The three problems mentioned are tightly woven into each other, which will be clear by the moment that each of the problems has to be solved repeatedly to gain an optimal solution. The next sections will provide the formal solutions.

PROBLEM 1 The aim is to compute the probability $P(\mathbf{O}|\Lambda)$, given an observation sequence $\mathbf{O}$ and fixed parameters Λ. A very straightforward way of doing this, would be to sum up the probabilities $P(\mathbf{O}|\mathbf{S}_z;\Lambda)$ of $\mathbf{O}$ under every possible state sequence $\mathbf{S}_z$, $z = 1 \ldots n^T$ with respect to the parameter set Λ.

Please behold for the following short section that the index notation slightly differs from what was introduced before. That is done for reasons of better understanding of the underlying principles. Afterwards the original notation will be used due to its shorter form.

A *single* state sequence $\mathbf{S} = S^1, S^2, \ldots, S^T$ occurs with the probability

$$P(\mathbf{S}|\Lambda) = i_{S^1} \cdot a_{S^1S^2} \cdot a_{S^2S^3} \cdots a_{S^{T-1}S^T} \tag{6.15}$$

where i_{S^1} denotes the initial probability of the state S^1 and $a_{S^iS^j}$ is the probability of a transition from state S^i to state S^j. The probability of a given observation sequence $\mathbf{O}$ under $\mathbf{S}$ and Λ is calculated by

$$P(\mathbf{O}|\mathbf{S};\Lambda) = \prod_{t=1}^{T} P(O^t|S^t;\Lambda) \tag{6.16}$$

Formula 6.16 can also be written as

$$P(\mathbf{O}|\mathbf{S};\Lambda) = b_{S^1}(O^1) \cdot b_{S^2}(O^2) \cdots b_{S^T}(O^T) \tag{6.17}$$

Please note, that these two results only hold under the condition of stochastically independent observations!

The probability of a given observation sequence $\mathbf{O}$ under a certain state sequence $\mathbf{S}$ (and a fixed parameter set Λ) is simply the joint probability of the appearance of $\mathbf{O}$ and $\mathbf{S}$, i.e. the product of the single probabilities:

$$P(\mathbf{O}, \mathbf{S}|\Lambda) = P(\mathbf{O}|\mathbf{S}; \Lambda) \cdot P(\mathbf{S}|\Lambda) \tag{6.18}$$

To yield the probability of $\mathbf{O}$ over *all* possible state sequences $\mathbf{S}_z$ it is necessary to sum up the term 6.18 over all state sequences. This gives the formula

$$\begin{aligned} P(\mathbf{O}|\Lambda) &= \sum_{z=1}^{n^T} P(\mathbf{O}|\mathbf{S}_z; \Lambda) \cdot P(\mathbf{S}_z|; \Lambda) = \\ &= \sum_{S^1, S^2, \ldots, S^T} i_{S^1} b_{S^1}(O^1) \cdot a_{S^1 S^2} b_{S^2}(O^2) \cdots a_{S^{T-1} S^T} b_{S^T}(O^T) \end{aligned} \tag{6.19}$$

Those products can be interpreted in a very straightforward way: starting at time $t = 1$ the system is in state S^1 with the initial probability i_{S^1} and generates the observation O^1 with probability $b_{S^1}(O^1)$. With the transition form $t = 1$ to $t = 2$ the state S^1 changes to S^2 (or stays the same) according to the transition probability $a_{S^1 S^2}$. The system in state S^2 generates again an observation O^2 with probability $b_{S^2}(O^2)$, and so on ... This process ends at the time $t = T$, i.e. the length of the observation sequence $\mathbf{O}$, when no more output is generated.

As comfortable this approach to the solution of problem 1 is, a closer look at the computational costs shows a disadvantage too important to neglect. A quick glance confirms a total of n^T sums, each consisting of $2 \cdot T$ products. Even a relatively small amount of states, e.g. $n = 4$, and a very short observation sequence, e.g. $T = 100$ would need $2 \cdot 100 \cdot 4^{100} \approx 3.5 \cdot 10^{62}$ computations, which is totally unacceptable for any practical application (since one usually operates with far greater numbers of states and observations).

The actual holder of the title of the fastest supercomputer on earth – the K computer, 20th February 2012 – can do over 10 petaFLOPS, i.e. $10 \cdot 10^{15}$ floating point operations per second [93]. For a solution of the task above even this computer would need $1.1 \cdot 10^{39}$ years, or roughly 10^{29} times the estimated age of the universe!

A by far not so obvious solution, gives us the so called forward-procedure which is the first part of the well known *Forward-Backward-Algorithm* (see [83]) – an algorithm developed for an efficient calculation of this problem.

FORWARD-PROCEDURE: The *forward variable* α_i^t is defined as

$$\alpha_i^t := P(O^1 \ \ldots \ O^t, S^t = S_i|\Lambda) \tag{6.20}$$

i.e. α_i^t is the probability of the system (with parameters Λ) being in state $\mathcal{S}_i$ at time t, having generated the partial observation sequence $O^1 \ldots O^t$. To obtain α_i^t for all times $t = 1, \ldots, T$ an inductive approach is used:

INITIALIZATION:

$$\alpha_j^1 = i_j \, b_j(O^1) \qquad \forall j = 1, \ldots, n \tag{6.21}$$

INDUCTION:

$$\alpha_j^t = \left(\sum_{i=1}^{n} \alpha_i^{t-1} \, a_{ij} \right) \cdot b_j(O^t) \qquad \forall j = 1, \ldots, n$$
$$t = 2, \ldots, T \tag{6.22}$$

TERMINATION:

$$P(\mathbf{O}|\Lambda) = \sum_{j=1}^{n} \alpha_j^T \tag{6.23}$$

The initialization step declares the forward probabilities as the joint probability of observing O^1 while being in state $S^1 = \mathcal{S}_j$. Like previously mentioned, α_j^t denotes the probability that the system is in state $S^t = \mathcal{S}_j$ while the output observed was $O^1, O^2, \ldots, O^t$.

The induction step calculates the forward probability as the joint probability of additionally observing O^{t+1} while now being in state $S^{t+1} = \mathcal{S}_j$, coming from all possible states $S^t = \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ (see figure 6.4). Mathematically this is done by multiplying α_i^t with the matching state transition probability a_{ij}. To obtain the overall probability one needs to sum up over all i. At last, to get the new forward probability α_j^{t+1} it is necessary to multiply again with the probability of observing O^{t+1} in the actual state.

Finally the termination step yields the desired value of $P(\mathbf{O}|\Lambda)$ by summation of the n forward variables α_j^T, $j = 1, \ldots, n$ at time step T, because $\alpha_j^T = P(O^1 \ldots O^T, S^T = \mathcal{S}_j|\Lambda) = P(\mathbf{O}, S^T = \mathcal{S}_j|\Lambda)$.

Remember, the horrid computational costs of the earlier approach showed the need for a more efficient calculation. But is the forward-procedure a decent solution to that problem?

To calculate the results for problem 1, i.e. $P(\mathbf{O}|\Lambda)$, the forward-procedure roughly needs T steps with $n \cdot n$ computations each, which results in $n^2 \cdot T$ operations. Given the afore-mentioned example ($n = 4$, $T = 100$), there will be an overall need of $4^2 \cdot 100 = 16 \cdot 100 = 1.6 \cdot 10^3$ computation steps. This shows a reduction of computational effort by 59 orders of magnitude!

The second part of the forward-backward-algorithm is the backward-procedure which will not be used until the solution to problem 3. But because forward- and backward- procedure belong to the same algorithm and are structured very similarly, it is alright to introduce it at this place.

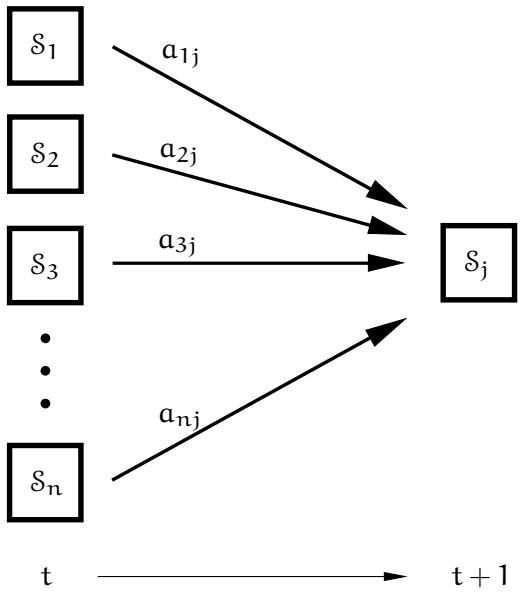


Figure 6.4: Computation of the forward variable α_j^{t+1}

BACKWARD-PROCEDURE: The *backward variable* β_i^t is defined as

$$\beta_i^t := P(O^{t+1} \ \ldots \ O^T | S^t = S_i, \Lambda) \tag{6.24}$$

i.e. β_i^t is the probability of the system (with parameters Λ) being in state S_i at time t, that will generate the partial observation sequence $O^{t+1} \ \ldots \ O^T$, which is the complementary partial state sequence to $O^1, O^2, \ldots, O^t$ used for the computation of α_i^t. Analogously to the Forward-Procedure β_i^t is obtained by a iterative approach:

INITIALIZATION:

$$\beta_j^T = 1 \qquad \forall j = 1, \ldots, n \tag{6.25}$$

INDUCTION:

$$\beta_j^t = \sum_{i=1}^{n} a_{ij} b_i(O^{t+1}) \beta_i^{t+1} \qquad \forall j = 1, \ldots, n$$
$$t = T-1, \ldots, 1 \tag{6.26}$$

TERMINATION:

$$P(\mathbf{O}|\Lambda) = \sum_{j=1}^{n} \beta_j^1 i_j b_j(O^1) \tag{6.27}$$

The initialization step sets the starting values (which are the β_j^T) arbitrarily to 1. The induction step can be seen as the opposite of the

induction step of the Forward-Algorithm, i.e. instead of going from many states at time t to a specific one at time $t+1$, the algorithm calculates going from a single state at t to multiple states at $t+1$. But please note that the time path runs the reverse way (see figure 6.5)! The termination step shows that the backward-variable also determines the same quantity as the forward variable, i.e. $P(\mathbf{O}|\Lambda)$.

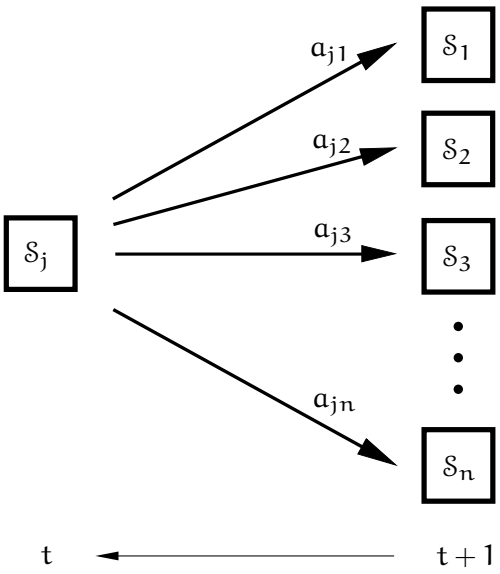


Figure 6.5: Computation of the backward variable β_j^t

Completely analogous to the Forward-Algorithm, the acquired order of computations is also $n^2 \cdot T$.

Problem 2

Remember, at the definition of problem 2 there was the wish to determine the "best" state sequence - but how to define "best"? There are many ways of doing this, but unfortunately not all of them are useful for further computation.

One of the easiest methods would be to search for those states, which maximize the given observation sequence. In other words, that would afford to maximize the observation probability for each time step t individually. The probability to be in state S_i at time t with given observation sequence $\mathbf{O}$ and model parameters Λ can be defined as a new variable

$$\gamma_i^t = P(S^t = S_i|\mathbf{O}, \Lambda). \tag{6.28}$$

Luckily it is possible to use the already known forward- and backward-variables α_i^t and β_i^t to rewrite the definition of γ_i^t, because they stand

for the system being in state $\mathcal{S}_i$ given the whole observation sequence ($O^1 \ldots O^t$ used by α_i^t and $O^{t+1} \ldots O^T$ used by β_i^t):

$$\gamma_i^t = \frac{\alpha_i^t \beta_i^t}{\sum_{i=1}^{n} \alpha_i^t \beta_i^t} \tag{6.29}$$

The normalization factor $\sum_{i=1}^{n} \alpha_i^t \beta_i^t$ makes sure that γ_i^t is a probability measure:

$$\sum_{i=1}^{n} \gamma_i^t = 1. \tag{6.30}$$

Then the *individually* most likely state $\mathcal{S}_i$ at time t is given by

$$S^t = \arg\max_i(\gamma_i^t). \tag{6.31}$$

But looking a bit deeper into this method, one can encounter severe problems very quickly! Imagine that the just found optimal state sequence includes invalid state transitions (e.g. $\mathcal{S}_1$ to $\mathcal{S}_1$ but the state transition probability from $\mathcal{S}_1$ to itself is zero)!

To cope with that problem, different solutions can be presented, but one of the most efficient ways is the widely used *Viterbi Algorithm*. The key aim of the Viterbi Algorithm is to find the full-length state sequence with the highest overall probability.

To be more concrete, the Viterbi Algorithm yields the state sequence $\mathbf{S} = S^1, S^2, \ldots, S^T$ which has the highest overall probability of the state transitions as well as of explaining the observation sequence $\mathbf{O} = O^1, O^2, \ldots, O^T$.

VITERBI ALGORITHM: For this algorithm a new variable δ_j^t is defined:

$$\delta_j^t = \max_{\{S^1, S^2, \ldots, S^{t-1}\}} P(S^1, S^2, \ldots, S^{t-1}, S^t = \mathcal{S}_j, O^1, O^2, \ldots, O^t | \Lambda) \tag{6.32}$$

where $\{S^1, S^2, \ldots, S^{t-1}\}$ has to be understood as the set of all possible state sequences up to the time $t-1$. If the iteration of the forward variable α is considered, it is easy to establish an iteration for δ by:

$$\delta_j^{t+1} = \max_i \; \delta_i^t \cdot a_{ij} \cdot b_j(O^{t+1}) \qquad \forall i = 1, \ldots, n \tag{6.33}$$

Finally it is necessary to also introduce a variable ψ which keeps track of the state $\mathcal{S}_i$ which actually maximizes the term 6.33. The whole procedure is given by

INITIALIZATION:

$$\delta_j^1 = i_j b_j(O^1) \tag{6.34}$$

$$\psi_j^1 = 0 \tag{6.35}$$

INDUCTION:

$$\delta_j^{t+1} = \max_i \ \delta_i^t \cdot a_{ij} \cdot b_j(O^{t+1}) \qquad \begin{matrix}\forall i,j=1,\dots,n \\ t=1,\dots,T-1\end{matrix} \tag{6.36}$$

$$\psi_j^{t+1} = \arg\max_i \ \delta_i^t \cdot a_{ij} \qquad \begin{matrix}\forall i,j=1,\dots,n \\ t=1,\dots,T-1\end{matrix} \tag{6.37}$$

TERMINATION:

$$P(\mathbf{S},\mathbf{O}|\Lambda) = \max_i \ \delta_i^T \tag{6.38}$$

$$S^T = \arg\max_i \ \delta_i^T \tag{6.39}$$

BACKTRACKING:

$$S^t = \psi_{S^{t+1}}^{t+1} \qquad t = T-1,\dots,1 \tag{6.40}$$

The computational effort for the Viterbi Algorithm is thereby comparable to those of forward- and backward-procedure.

Problem 3

The third problem deals with the fitting of the model parameters $\Lambda = (A, B, I)$ in order to maximize the overall probability of the observation sequence, regarding the given model. Unfortunately there exists no possibility to do this analytically, like for the other two problems before. Even worse, there is no way known to solve this problem by any means in a global context.

The most popular solution for this is to use well established iterative methods to find at least a local maximum of the probability function $P(\mathbf{O}|\Lambda)$. This can be done, for example, with standard and advanced gradient descent techniques [94] or with the *Expectation-Maximization-Algorithm*, referred to also in short as *EM-Algorithm*. Here we will use the *Baum-Welch-algorithm* which is a special case of the EM-algorithm, using posterior probabilities and a maximum likelihood estimator [95].

The EM-ALGORITHM is a two-step iterative algorithm which consists of

EXPECTATION STEP: It uses the forward-backward procedure which was discussed previously to estimate the expected state transitions under different conditions.

Therefore a new variable $\xi_{(i,j)}^t$ is introduced, describing the probability of being in state $\mathcal{S}_i$ at time t and in state $\mathcal{S}_j$ at time $t+1$, regarding the model Λ and the observation sequence $\mathbf{O}$:

$$\xi_{(i,j)}^t = P(S^t = \mathcal{S}_i, S^{t+1} = \mathcal{S}_j|\mathbf{O},\Lambda). \tag{6.41}$$

It is possible to take advantage of the previous defined variables α_i^t and β_i^t and write $\xi_{(i,j)}^t$ in the form

$$\xi_{(i,j)}^t = \frac{\alpha_i^t \cdot a_{ij} b_j(O^{t+1}) \cdot \beta_j^{t+1}}{P(\mathbf{O}|\Lambda)} = \tag{6.42}$$

$$= \frac{\alpha_i^t \cdot a_{ij} b_j(O^{t+1}) \cdot \beta_j^{t+1}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\alpha_i^t \cdot a_{ij} b_j(O^{t+1}) \cdot \beta_j^{t+1} \right)} \tag{6.43}$$

which can be understood easier by consideration of figure 6.6. The fracture consists of $P(S^t = \mathcal{S}_i, S^{t+1} = \mathcal{S}_j, \mathbf{O}|\Lambda)$ divided by the overall probability $P(\mathbf{O}|\Lambda)$. This ensures that $\xi_{(i,j)}^t$ is a probability measure.

There exists a direct connection between $\xi_{(i,j)}^t$ and γ_i^t, which is

$$\gamma_i^t = \sum_{j=1}^{n} \xi_{(i,j)}^t. \tag{6.44}$$

This fact can be easily understood when looking at the signification of the two variables:

$$\sum_{t=1}^{T-1} \xi_{(i,j)}^t = \text{expected number of transitions from state } \mathcal{S}_i \text{ to } \mathcal{S}_j \tag{6.45}$$

$$\sum_{t=1}^{T-1} \gamma_i^t = \text{expected number of all transitions from state } \mathcal{S}_i \tag{6.46}$$

Note that due to the construction of the variable $\xi_{(i,j)}^t$, which contains the term β^{t+1}, it is necessary to limit the sum over all time points to $T-1$!

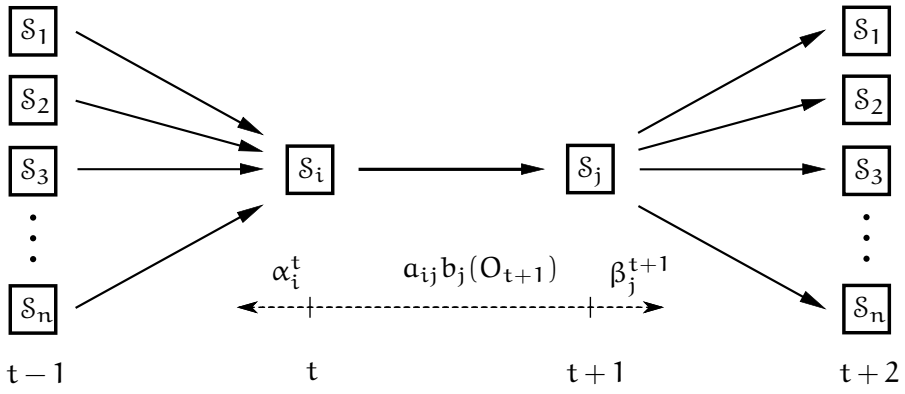


Figure 6.6: Computation of $\xi_{(i,j)}^t$

MAXIMIZATION STEP: Based on this descriptive understanding of these two variables, it is straight forward to derive formulas for the re-estimation of the model parameters of the HMM, which are $\Lambda = (\mathbf{A}, \mathbf{B}, \mathbf{I})$:

$$\begin{aligned}\hat{a}_{i,j} &= \frac{\sum_{t=1}^{T-1} \xi^t_{(i,j)}}{\sum_{t=1}^{T-1} \gamma^t_i} \\ &= \frac{\text{expected number of transitions from state } \mathcal{S}_i \text{ to } \mathcal{S}_j}{\text{expected number of all transitions from state } \mathcal{S}_i}\end{aligned} \tag{6.47}$$

$$\begin{aligned}\hat{b}_i(\mathcal{O}_k) &= \frac{\sum_{t=1}^{T} \gamma^t_i}{\sum_{t=1}^{T} \gamma^t_i} \\ &= \frac{\text{expected number of observations } \mathcal{O}_k \text{ in } \mathcal{S}_i}{\text{expected number of visits of state } \mathcal{S}_i}\end{aligned} \tag{6.48}$$

$$\begin{aligned}\hat{\imath}_i &= \gamma^1_i \\ &= \text{expected number of visits of state } \mathcal{S}_i \text{ at time } t = 1\end{aligned} \tag{6.49}$$

These formulas are the likelihood estimators for the parameters. Please note that the stochastic constraints

$$\sum_{i=1}^{n} \hat{\imath}_i = 1 \tag{6.50}$$

$$\sum_{j=1}^{n} \hat{a}_{i,j} = 1 \qquad \forall i = 1, \ldots, n \tag{6.51}$$

$$\sum_{i=k}^{m} \hat{b}_i(\mathcal{O}_k) = 1 \qquad \forall i = 1, \ldots, n \tag{6.52}$$

$$\tag{6.53}$$

are all satisfied for all iterations per definition of the update step.

If we denote the actual model parameter set as Λ and the re-estimated parameter set as $\hat{\Lambda}$ it can be shown that there holds

$$P(\mathbf{O}|\hat{\Lambda}) \geqslant P(\mathbf{O}|\Lambda). \tag{6.54}$$

According to the work of Baum et. al the initial model is either at a (local) optimum, i.e $P(\mathbf{O}|\hat{\Lambda}) = P(\mathbf{O}|\Lambda)$ or the improved model after one Expectation-Maximization-step has an improved posterior probability $P(\mathbf{O}|\hat{\Lambda}) > P(\mathbf{O}|\Lambda)$ [96].

Going on iteration the Expectation and Maximization steps will lead to a (local) optimum of the parameter set in terms of posterior probability. The result we obtain is therefore called *Maximum-Likelihood-Estimate* of the HMM.

6.1.3 *Scaling*

The above described algorithm is neat and easy to understand but bears a major problem when implemented on any computer. When taking a look at how we introduced the forward variable α^t (see 6.21 and 6.22) it becomes clear that it consists of the sum of terms of the form

$$\prod_{r=1}^{t-1} a_{S^r S^{r+1}} \prod_{r=1}^{t} b_r(O^r) \tag{6.55}$$

where the probabilities $a_{i,j}$ and b_j are smaller than one, in the most cases indeed remarkably smaller than one! This leads to numerical instabilities due to round-off errors at the limitation of the machine accuracy, even for relatively small numbers of time points t!

The same problem we encounter at the computation of the backward variable β^t and subsequently at all further calculations in the EM-algorithm.

Therefore it is necessary for computer implementation to counteract this problem with a scaling procedure. Luckily forward and backward variable are set up in similar orders of magnitude. This allows us to use an easy way of scaling, which is to scale by multiplication with a factor only depending on t, i.e. it is chosen constant for all α_i as well as for all β_i at every time step. This helps to achieve the goal of keeping the algorithm in the dynamic range of our computation system with the positive side effect of having this factor cancelled out when doing the update step in the EM-algorithm, yielding the exact same results as without scaling.

Since the scaling procedure is adapted with every time step t it is necessary to include it in the forward and backward algorithm. Exemplarily we want to take a look at the SCALED FORWARD ALGORITHM:

INITIALIZATION:

$$\alpha_j^1 = i_j \, b_j(O^1) \qquad \forall j = 1, \ldots, n \tag{6.56}$$

$$z_1 = 1 \tag{6.57}$$

$$\tilde{\alpha}_j^1 = \alpha_j^1 \tag{6.58}$$

INDUCTION:

$$\alpha_j^t = \left(\sum_{i=1}^{n} \tilde{\alpha}_i^{t-1}\, a_{ij}\right) \cdot b_j(O^t) \qquad \begin{matrix}\forall j=1,\ldots,n \\ t=2,\ldots,T\end{matrix} \tag{6.59}$$

$$z_t = \frac{1}{\sum_{j=1}^{n} \alpha_j^t} \tag{6.60}$$

$$\tilde{\alpha}_j^t = \left(\prod_{\tau=1}^{t} z_\tau\right) \alpha_j^t \tag{6.61}$$

TERMINATION:

$$P(\mathbf{O}|\Lambda) = \sum_{j=1}^{n} \tilde{\alpha}_j^T \tag{6.62}$$

To prove that the scaling factors z_t eventually cancel out, we will take a look at the re-estimation formula for $a_{i,j}$:

$$\begin{aligned}
\hat{\hat{a}}_{i,j} &= \frac{\sum_{t=1}^{T-1} \tilde{\alpha}_i^t a_{i,j} b_j(O^{t+1}) \tilde{\beta}_j^{t+1}}{\sum_{t=1}^{T}\sum_{t=1}^{T-1} \tilde{\alpha}_i^t a_{i,j} b_j(O^{t+1}) \tilde{\beta}_j^{t+1}} = \\
&= \frac{\sum_{t=1}^{T-1} \left(\prod_{\tau=1}^{t} z_\tau\right) \alpha_i^t a_{i,j} b_j(O^{t+1}) \left(\prod_{\tau=t+1}^{T} z_\tau\right) \beta_j^{t+1}}{\sum_{t=1}^{T}\sum_{t=1}^{T-1} \left(\prod_{\tau=1}^{t} z_\tau\right) \alpha_i^t a_{i,j} b_j(O^{t+1}) \left(\prod_{\tau=t+1}^{T} z_\tau\right) \beta_j^{t+1}} = \\
&= \frac{\sum_{t=1}^{T-1} \left(\prod_{\tau=1}^{T} z_\tau\right) \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}}{\sum_{t=1}^{T}\sum_{t=1}^{T-1} \left(\prod_{\tau=1}^{T} z_\tau\right) \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}} = \\
&= \frac{\left(\prod_{\tau=1}^{T} z_\tau\right) \sum_{t=1}^{T-1} \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}}{\left(\prod_{\tau=1}^{T} z_\tau\right) \sum_{t=1}^{T}\sum_{t=1}^{T-1} \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}} \\
&= \frac{\sum_{t=1}^{T-1} \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}}{\sum_{t=1}^{T}\sum_{t=1}^{T-1} \alpha_i^t a_{i,j} b_j(O^{t+1}) \beta_j^{t+1}} = \frac{\sum_{t=1}^{T-1} \xi_{(i,j)}^t}{\sum_{t=1}^{T-1} \gamma_i^t} = \hat{a}_{i,j}
\end{aligned} \tag{6.63}$$

As this shows, the re-estimated values of the parameter sets (b_i and i_i behave identically) are not affected by the scaling procedure. However, this does not hold for the computation of the posterior probability of

the model $P(\mathbf{O}|\lambda) = \sum_{j=1}^{n} \tilde{\alpha}_j^T$, because the sum of the already scaled forward variables does not meet the stochastic constraints! But we can exploit the fact that there must hold

$$1 = \left(\prod_{t=1}^{T} z_t\right) \cdot \sum_{j=1}^{n} \alpha_j^T = \left(\prod_{t=1}^{T} z_t\right) \cdot P(\mathbf{O}|\lambda) \tag{6.64}$$

$$\tag{6.65}$$

This leads to

$$P(\mathbf{O}|\lambda) = \frac{1}{\left(\prod_{t=1}^{T} z_t\right)} \tag{6.66}$$

According to the fact, that the posterior probability will be lower than the machine accuracy, we are going to compute the logarithm of the probability as a measurement for model quality:

$$\log P(\mathbf{O}|\lambda) = -\sum_{t=1}^{T} \log z_t \tag{6.67}$$

The Viterbi algorithm (for computing the most likely state sequence) has to deal with the same problem due to scaling. Therefore one evades those difficulties again by using the logarithm. This gives the SCALED VITERBI ALGORITHM:

Instead of the previous variable δ_j^t we define analogously

$$\phi_j^t = \max_{\{S^1, S^2, \ldots, S^{t-1}\}} \log P(S^1, S^2, \ldots, S^{t-1}, S^t = s_j, O^1, O^2, \ldots, O^t|\lambda) \tag{6.68}$$

Then the algorithm changes to

INITIALIZATION:

$$\phi_j^1 = \log i_j + \log b_j(O^1) \tag{6.69}$$

$$\psi_j^1 = 0 \tag{6.70}$$

INDUCTION:

$$\phi_j^{t+1} = \max_i \ \phi_i^t + \log a_{ij} + \log b_j(O^{t+1}) \qquad \begin{matrix}\forall i,j=1,\ldots,n \\ t=1,\ldots,T-1\end{matrix} \tag{6.71}$$

$$\psi_j^{t+1} = \arg\max_i \ \phi_i^t \cdot a_{ij} \qquad \begin{matrix}\forall i,j=1,\ldots,n \\ t=1,\ldots,T-1\end{matrix} \tag{6.72}$$

TERMINATION:

$$\log P(\mathbf{S}, \mathbf{O}|\lambda) = \max_i \ \phi_i^T \tag{6.73}$$

$$S^T = \arg\max_i \ \phi_i^T \tag{6.74}$$

BACKTRACKING:

$$S^t = \psi_{S^{t+1}}^{t+1} \qquad t = T-1, \ldots, 1 \tag{6.75}$$

This improved algorithm helps to avoid numerical problems with no extra computational costs. The logarithmic posterior probabilities are nonetheless comparable to each other due to the fact that the logarithm is a monotonous augmenting function. So now the EM-algorithm will yield the parameter set which accounts for the highest logarithmic posterior probability, which is the parameter set which accounts for the highest non-logarithmic probability.

6.1.4 *Model topology*

It should be mentioned with a few words that so far we have always depicted fully connected ergodic models. In general holds that a model is called *ergodic* if every state can be reached from every other state in a limited amount of time steps. The specification "fully connected" reduces this amount to one, i.e. it is possible to reach every state in one time step, regardless where one starts.

Note that the smaller a transition probability gets, the more those two states get disconnected. This means that a diagonal transition matrix will reflect a model with completely independent states, while an upper triangular matrix will reflect a so called left-right model, where it is only possible to migrate in one direction, but not back. As this shows, the model topology reflects in the structure of the transition matrix $\mathbf{A}$ and vice versa.

Such models find their field of application in speech and handwriting recognition as well as in genomics and proteomics [87, 97, 98]. For further information on this topic please read into [83], for hierarchical HMMs have a look at [99].

In the following application of HMMs to vigilance detection we will limit ourselves to such fully connected ergodic models and try to adapt to the reality by adapting the parameters, i.e. the transition probabilities.

6.2 CONTINUOUS HIDDEN MARKOV MODELS

Above we have described discrete Hidden Markov Models. Those are discrete in their states as well as in their observations. These models are well suited for applications where the model is based on a finite set of observations, like for example the matching of genetic sequences, where one observation can only be one of the four nucleobases (Adenine, Cytosine, Guanine and Thymine). Proteins for example, consist of amino acids which are encoded by triplets of nucleobases. So a model for protein decoding sites in human DNA has

to use observations out of the observation alphabet $m = 4$ to adapt to the $n = 22$ different states which stand for the 22 different amino acids. (Please note that this model description is heavily simplified for demonstration purposes - for further information on that topic see [100].

But when looking further, one sees that many real world systems cannot be sufficiently described in a discrete way because the observations that one gets are e.g. measurements on a continuous scale. Unfortunately discretization is not always a good idea since every discretization scheme bears the danger of concealing underlying dependencies. The same problem appears also when using EOG-data for vigilance classification. The measurements of the EOG-channels are continuous and higher dimensional, which hinders a meaningful discretization scheme. For that reason Hidden Markov Models can be generalized to match the need of such conditions.

There exist two major generalization steps to eliminate the strong limitations of discreteness. These are:

A. discrete states and continuous observations

B. continuous states and continuous observations

A. The first approach – *discrete states and continuous observations* – adapts the model to observations on a continuous scale without touching the underlying (discrete) structure of states. That means that instead of using discrete probability densities we move on to a finite mixture of probability density functions – in the simplest case Gaussian densities. In that case the correct denomination of such a continuous HMM would be *Gaussian Hidden Markov Model* (GHMM). A mixture probability for an observation O^t has the form

$$b_i(O^t) = \sum_{v=1}^{V} c_{iv}\, \mathcal{N}_{iv}(O^t) = \sum_{v=1}^{V} c_{iv}\, \mathcal{N}(O^t; \mu_{iv}, \Sigma_{iv}) \tag{6.76}$$

where $\mathcal{N}(O^t; \mu, \Sigma)$ stands for the Gaussian probability density $\mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ at the point of observation O^t. We use $\mathcal{N}_{iv}(O^t)$ as a shortened version for simpler notation. The observations per se can be scalars or vectors, i.e. the probability densities dealt with are univariate or multivariate. Since this does not change anything in the process, we are using the more general case of vector observations (still notated as O^t) and multivariate distributions throughout this chapter. V is the number of mixed distributions and the c_{iv} are their respective *mixture coefficients* which have to fulfill

$$\sum_{v=1}^{V} c_{iv} = 1 \qquad \forall i \tag{6.77}$$

$$c_{iv} \geqslant 0 \qquad \forall i, v \quad . \tag{6.78}$$

This ensures that the $b_i(.)$ are adequate probability density functions.

In the discrete case we used the EM-algorithm to compute the maximum likelihood estimators for the initial distribution i_i state transitions a_{ij} and the observation distribution b_i. Now this observation distribution is of the form 6.76 which depends on the mixture coefficients, means and covariances. Working with the continuous probability densities, these three variables can be estimated through:

$$\begin{aligned}\hat{c}_{iv} &= \frac{\sum_{t=1}^{T} \gamma_{iv}^t}{\sum_{t=1}^{T}\sum_{v=1}^{V} \gamma_{iv}^t} \\ &= \frac{\text{expected number of times in } \mathcal{S}_i \text{ regarding } \mathcal{N}_{iv}}{\text{expected number of times in } \mathcal{S}_i}\end{aligned} \tag{6.79}$$

$$\begin{aligned}\hat{\mu}_{iv} &= \frac{\sum_{t=1}^{T} \gamma_{iv}^t \cdot O^t}{\sum_{t=1}^{T} \gamma_{iv}^t} \\ &= \frac{\text{weighted sample mean regarding } \mathcal{S}_i \text{ and } \mathcal{N}_{iv}}{\text{expected number of times in } \mathcal{S}_i}\end{aligned} \tag{6.80}$$

$$\begin{aligned}\hat{\Sigma}_{iv} &= \frac{\sum_{t=1}^{T} \gamma_{iv}^t \cdot (O^t - \mu_{iv})(O^t - \mu_{iv})^\intercal}{\sum_{t=1}^{T} \gamma_{iv}^t} \\ &= \frac{\text{weighted sample covariance regarding } \mathcal{S}_i \text{ and } \mathcal{N}_{iv}}{\text{expected number of times in } \mathcal{S}_i}\end{aligned} \tag{6.81}$$

Here the symbol $^\intercal$ indicates the transpose operation. The term γ_{iv}^t is a multivariate generalization of the in 6.29 and is defined as

$$\gamma_{iv}^t = \left(\frac{\alpha_i^t \beta_i^t}{\sum_{i=1}^{n} \alpha_i^t \beta_i^t}\right) \cdot \left(\frac{c_{iv}\, \mathcal{N}(O^t; \mu_{iv}, \Sigma_{iv})}{\sum_{v=1}^{V} c_{iv}\, \mathcal{N}(O^t; \mu_{iv}, \Sigma_{iv})}\right) \tag{6.82}$$

It can be easily seen that this term transforms to γ_i^t in the case of a trivial mixture ($V = 1$) or a discrete density.

Note: The above described mechanisms of continuous observation mixture models can be easily modified to all other log-concave or elliptically symmetric densities [101], as shown for student-t distributions in the following section.

B. The second approach – *continuous states and continuous observations* – is better known under the name *Kalman filter* and is the border case of a HMM with n states where $n \to \infty$. The mathematics behind this algorithm correspond also very closely to that of digital filtering methods what explains also its name and origin. A good introduction and more details on Kalman filters are given by [102, 103].

6.3 EXPANSION TO STUDENT'S-T DISTRIBUTIONS

Gaussian Mixture Models (GMM), i.e. a mixture model based on normal distributions, is the most used variant for continuous Hidden Markov Models since the *central limit theorem* states that a large number of independent random variables drawn from the same underlying distribution is approximately normally distributed. So it is only logical to use GMMs to model the unknown distributions, but outliers can severely distort the convergence of the HMM.

To come by such situations, which are likely to happen when working with real world data, many attempts have been undertaken to avoid the pitfalls [104, 105]. Most of them try to improve the training or classification pattern in order to succeed in making the HMM less vulnerable, what are mostly heuristic approaches tailored to very special applications, e.g. speech recognition.

A much more universal method for an outlier resistant Hidden Markov Model is described in [106]. Instead of normal distributions, one makes use of the heavy-tailed nature of Student's-t distributions. This means that the boundary areas of the distributions are attributed much more weight so that extreme outliers still have higher probabilities than in the Gaussian case. This helps in several situations:

- The mixture distributions are not as much dragged towards the outliers.
- Outlier probabilities do not fall below machine accuracy, causing numerical troubles.
- Less mixture probabilities are necessary to describe the data.
- Extenuated requirements for the training data.

The *probability density function* (pdf) of the multivariate Student's-t distribution is characterized by

$$\mathcal{T}(O^t; \mu, \Sigma, \nu) = \frac{\Gamma(\frac{\nu+p}{2})|\Sigma|^{-\frac{1}{2}}(\pi\nu)^{-\frac{p}{2}}}{\Gamma(\frac{\nu}{2})(1 + \frac{1}{\nu}(O^t - \mu)^T\Sigma^{-1}(O^t - \mu))} \tag{6.83}$$

with

$\mathcal{T}$	...	Student's-t probability density
O^t	...	vector of observations at time point t
μ	...	vector of means
Σ	...	covariance matrix
ν	...	degrees of freedom
p	...	dimensionality of the observations
Γ	...	Gamma function
$\lvert . \rvert$	...	determinant
$(O^t - \mu)^\tau \Sigma^{-1} (O^t - \mu)$	...	Mahalanobis distance
τ	...	matrix transpose.

In [106] it is explained how the Student's-t distribution and the normal distribution interact with each other and why it is possible to use the Student's-t distributions for the HMM. One important result is, that the degrees of freedom ν is the parameter which influences the shape of the distribution, i.e. it defines how much weight there is on the tails of the distribution. For $\nu \to \infty$ the Student's-t distribution converges to a Gaussian distribution, leaving proportionally less weight on the tails.

Since the EM-algorithm for Student's-t HMMs is also defining the optimal ν, this measure – for a converged model – gives also inference of how big is the influence of the outliers, and if a Gaussian Mixture Model would have problems with that dataset.

6.3.1 *The Student's-t Hidden Markov Model (SHMM)*

We can use a SHMM only when we consider the hidden observation distributions being mixtures of Student's-t distributions. For the sake of simplicity we assume – as above for the GHMM – that the number of mixtures is the same for every state. The parameters for the SHMM are now defined analogously to the Gaussian Hidden Markov Model:

$\mathcal{S}_i$	...	states of the model
n	...	number of distinct states
$\mathbf{A}$	...	$n \times n$—matrix of state transition probabilities
$\mathbf{I}$	...	$n \times 1$—vector of the initial state distribution
$\mathbf{O}$	...	observation sequence (multidimensional)
V	...	number of mixtures

The probability density of a single observation O^t coming from the i-th model state $\mathcal{S}_i$ is given by

$$b_i(O^t) = \sum_{\nu=1}^{V} c_{i\nu} \cdot \mathcal{T}(O^t; \mu_{i\nu}, \Sigma_{i\nu}, \nu_{i\nu}) \tag{6.84}$$

with the mixture weights $c_{i\nu}$ which again fulfill the restraints 6.77 and 6.78.

We will directly step into the EM-part of the Baum-Welch algorithm, implied that the forward, backward and Viterbi algorithms have been already computed, switching the normal distributions to the Student's-t distributions. Using the forward and backward variables α_i^t and β_i^t we then yield the update variables

$$\xi_{(i,j)}^t = \frac{\alpha_i^t \cdot a_{ij} b_j(O^{t+1}) \cdot \beta_j^{t+1}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\alpha_i^t \cdot a_{ij} b_j(O^{t+1}) \cdot \beta_j^{t+1} \right)} \tag{6.85}$$

$$\gamma_{i\nu}^t = \left(\frac{\alpha_i^t \beta_i^t}{\sum_{i=1}^{n} \alpha_i^t \beta_i^t} \right) \cdot \left(\frac{c_{i\nu}\, \mathcal{T}(O^t; \mu_{i\nu}, \Sigma_{i\nu})}{\sum_{\nu=1}^{V} c_{i\nu}\, \mathcal{T}(O^t; \mu_{i\nu}, \Sigma_{i\nu})} \right) \tag{6.86}$$

THE EM-ALGORITHM For the Expectation-Maximization algorithm we are going to compute the updates for every iteration of the algorithm, denoted by the symbol $\hat{}$, like in the case of the GHMM. The EM-updates of the initial probabilities and the state transition probabilities, i.e. $\hat{i}_i$ and $\hat{a}_{i,j}$ are calculated the exact same way as in 6.49 and 6.47.

The updated model parameters are

$$\hat{c}_{i\nu} = \frac{\sum_{t=1}^{T} \gamma_{i\nu}^t}{\sum_{t=1}^{T} \sum_{\nu=1}^{V} \gamma_{i\nu}^t} \tag{6.87}$$

$$\hat{\mu}_{i\nu} = \frac{\sum_{t=1}^{T} \gamma_{i\nu}^t \cdot \hat{u}_{i\nu} \cdot O^t}{\sum_{t=1}^{T} \gamma_{i\nu}^t \cdot \hat{u}_{i\nu}} \tag{6.88}$$

$$\hat{\Sigma}_{i\nu} = \frac{\sum_{t=1}^{T} \gamma_{i\nu}^t \cdot \hat{u}_{i\nu} \cdot (O^t - \mu_{i\nu})(O^t - \mu_{i\nu})^\intercal}{\sum_{t=1}^{T} \gamma_{i\nu}^t} \tag{6.89}$$

$$\tag{6.90}$$

where $\hat{u}_{i\nu}^t$ are the updated precision scalars (see [106]) which are obtained through

$$\hat{u}_{i\nu}^t = \frac{\nu_{i\nu} + p}{\nu_{i\nu} + (O^t - \mu_{i\nu})^\intercal \Sigma_{i\nu}^{-1} (O^t - \mu_i)} \tag{6.91}$$

We yield the update of the degrees of freedom $\hat{\nu}_{i\nu}$ for the Student's-t mixtures by solving the following implicit equation for $\hat{\nu}_{i\nu}$:

$$1 - \psi\left(\frac{\hat{\nu}_{i\nu}}{2}\right) + \log\psi\left(\frac{\hat{\nu}_{i\nu}}{2}\right) + \psi\left(\frac{\hat{\nu}_{i\nu}+p}{2}\right) + \log\psi\left(\frac{\hat{\nu}_{i\nu}+p}{2}\right) + \frac{\sum_{t=1}^{T}\gamma_{i\nu}^{t}\cdot(\log\hat{u}_{i\nu}-\hat{u}_{i\nu})}{\sum_{t=1}^{T}\gamma_{i\nu}^{t}} = 0 \tag{6.92}$$

where $\psi(.)$ is the *digamma function*. This gives us all needed EM-updates, so that the modified Baum-Welch algorithm also works on this problem.

Regarding the computational efficiency of the operations, [106] shows that the complexity of the SHMM lies only marginally higher than the one for GHMMs. That is soothing since this would allow also a future application in online systems for vigilance surveillance.

6.4 MODEL ORDER SELECTION AND VALIDATION

When developing a Hidden Markov Model for a certain application, one of the most crucial choices is the one of the number of different states n, which is also called the *order* of the model. The order does not only heavily influence the efficiency of the model, but should at best also aid to the interpretation of a matching model theory.

In some cases the number of states is determined by existing knowledge of the process, e.g. in the case of genetic sequence matching one is aware that there could only be four different states, which represent the four nucleobases A, G, C, T or A, G, C, U, depending if DNA or RNA is examined.

But most of the time, the exact number of states is not known – not for nothing are we working with *Hidden* Markov Models. To make an educated guess is not as simple as it may seem. Like we discussed above, every new state raises the number of parameters to estimate quadratically. On the other hand yield too simple models a bad fit to the data. And last but not least could be a wrong number of states a hurdle in finding an appropriate theory to explain the observations.

6.4.1 *Likelihood-based criteria*

The most obvious criterion to qualify a HMM is the (log-)likelihood, given by the EM-algorithm. This value measures the fit of the model on the data, and is therefore prone to overfitting, i.e. the more states we add, the better will be the fit and the according model (log-)likelihood. So this measure will be maximized when every different observation is accorded its own state.

It is clear, that this is not at all what we want a model order validation criterion to do. Several criteria have been developed which are based on the log-likelihood of the model, but which do also penalize the number of parameters to determine.

AIC The Akaike information criterion (AIC), developed by Hirotugo Akaike [107], is a very simple yet effective measure, which combines the log-likelihood and the number of parameters to determined in a simple formula:

$$\mathtt{AIC} = 2(p - \ln L) \tag{6.93}$$

where p is the number of independent parameters and L is the likelihood of the model.

The AIC is *the* widespread order selection criterion. But due to the fact that it is based on the large-sample properties of maximum likelihood estimators, the application of AIC on small-sample observations results in overfitting [108].

AICC Therefore a corrected version of the AIC criterion was developed (AICc), which was designed to yield more accurate results in the case of small sample sizes [108]. For small observation sizes, the AICc gives more weight to the model complexity, whereas it converges asymptotically to the AIC criterion for increasing sample size.

$$\mathtt{AICc} = \mathtt{AIC} + \frac{2p(p+1)}{l-p-1} \tag{6.94}$$

Here p stands again for the number of undefined parameters in the model, and l is the length of the observation sequence. For our models with rather short observation sequences, we will put most confidence in the AICc value.

BIC Another measure is the Bayesian information criterion (BIC), which penalizes the number of parameters additionally with the logarithm of the observation sequence length:

$$\mathtt{BIC} = 2\left(p\frac{\ln l}{2} - \ln L\right) \tag{6.95}$$

where again p is the number of parameters, l is the length of the observation sequence and L is the model likelihood [89].

For all three criteria discussed above, the optimal number of states, i.e. the model order, is the one where the criterion reaches the minimum. Of course the model order will also be dependent on the underlying theory of the sequence generation, i.e. that not always the order with the minimal validation criterion explains the data best. But in all cases, a large difference of model orders between theory and criterion should be questioned and investigated.

6.4.2 *Ordinary pseudo-residuals*

When we decided on a number of states which seems to be suited according to some criterion from above, there is always the question left how well the fit is qualitatively and if we could possibly identify outliers in the data which hinder the fitting process. Derived from the residual-based model checking in the theory of regression models it is possible to define so called pseudo-residuals for our HMM [89]. These are able to perform the same role as traditional residuals.

To motivate the following definitions we want to keep in mind the following fact:

Be X a stochastic variable with continuous distribution function $F(.)$ then $Y = F(X)$ is uniformly distributed on the interval $(0, 1)$, i.e.

$$Y \sim U(0, 1). \tag{6.96}$$

The proof is easily constructed from the properties of the inverse cumulative distribution function, which is also called the quantile function.

According to [89] we can then define the *uniform pseudo-residuals* of an observation O^t coming from a continuous stochastic variable X^t as

$$u^t = P(X^t \leqslant O^t) = F_{X^t}(O^t). \tag{6.97}$$

If we chose the right type of model (regarding parameters, distributions, number of states) u^t should be uniformly distributed.

One big advantage of using the concept of pseudo-residuals is the comparability of results. Even when we are confronted with different probability functions at every time point t, where it is impossible to directly compare the observations, the pseudo-residuals are always uniformly distributed and thus well comparable.

When considering extreme observations, the related uniform pseudo-residuals would lie near zero and one and it is – especially in histogram plots with medium bin size – very difficult to distinguish outliers from reasonable observations at the margins of the distribution. To circumnavigate this problem it is advisable to transform the pseudo-residuals to a standard-normal shape.

This can be achieved by making use of the cumulative distribution function Φ of the standard normal distribution. Be X a stochastic variable with continuous distribution function $F(.)$ then $Z = \Phi^{-1}F(X)$ is distributed standard normal, i.e.

$$Z \sim \mathcal{N}(0, 1). \tag{6.98}$$

Analogously to above, we define the *normal pseudo-residuals* as

$$z^t = \Phi^{-1}(u^t) = \Phi^{-1}F_{X^t}(O^t). \tag{6.99}$$

Now, the validity of the model is checked, by comparing the resulting normal pseudo-residuals to a standard normal distribution. In this

case it is easier to identify outliers at the margins of the probability density function because the values of the residuals increase with increasing deviation form the distribution median.

When it comes down to calculating the pseudo-residuals Zucchini and MacDonald [89] distinguish between two methods:

ORDINARY PSEUDO-RESIDUALS In the offline case, where we can access the whole time series, the pseudo-residuals are calculated sample by sample in respect to *all* other observations. Written formally this gives us

$$z^t = \Phi^{-1}\left(P(X^t \leqslant O^t | \mathbf{O}^{-t})\right) \tag{6.100}$$

where $\mathbf{O}^{-t}$ is the full observation sequence missing only the element O^t:

$$\mathbf{O}^{-t} = O^1, \ldots, O^{t-1}, O^{t+1}, \ldots, O^T. \tag{6.101}$$

For a correct model, the pseudo-residuals z^t should be distributed standard normal for all $t \in 1, \ldots, T$.

FORECAST PSEUDO-RESIDUALS In an online environment, i.e. when used in a real time application, one has to settle on the already received observations to construct the pseudo-residuals. That leaves us with the formula

$$z^t = \Phi^{-1}\left(P(X^t \leqslant O^t | \mathbf{O}^{t-1})\right) \tag{6.102}$$

where

$$\mathbf{O}^{t-1} = O^1, \ldots, O^{t-1}. \tag{6.103}$$

Since this thesis deals only with the offline use of Hidden Markov Models, we will restrict ourselves to the ordinary pseudo-residuals in the chapters to come. In order to check the model validity, we will use plots of the uniform and standard pseudo-residuals together with their respective target values, as well as quantile-quantile plots (Q-Q plots) and the autocorrelation function (ACF) of the normal pseudo-residuals (see chapter 8).

Please note that the concepts of pseudo-residuals rely on continuous distribution functions and have to be changed in order to be suitable for discrete distributions. See [89] for details.

6.4.3 *Order estimation*

We have now spoken about how one can validate a choice of model order by comparing certain criteria to models with different order. With the vigilance detection models we have developed, we will get by with a trial-and-error method. This of course not completely random,

but based on the fact, that models should be as simple as possible – especially for relatively short observation sequences.

For the sake of completeness we nevertheless want to point out, that there exist more sophisticated approaches to estimate the model order, before or even instead of using the above discussed criteria, e.g. [109, 110].

7

DATA ACQUISITION AND PROCESSING

The data used in the following for model building and verification originate from the SENSATION project ("Advanced Sensor Development for Attention, Stress, Vigilance & Sleep/Wakefulness Monitoring"), a project funded by IST (Information Society Technologies) – the framework program for research and development of the European Union. The project description quotes their aims as follows:

> "This aim is targeted through a number of intermediate objectives and achievements with the scope of hypovigilance detection, prediction and management, as well as, diagnosis, treatment and remote monitoring of sleep disorders that will provide a safeguard for promoting peoples' health and safety, as well as environmental protection, in a variety of application fields such as medical, industrial and transportation."

In the following a brief description is given, which experiments have been done, which measurements have been undertaken and how we processed the EOG signal and derived the relevant features from it. For a more detailed description of the work done by any partner in the SENSATION project, please visit the website *www.sensation-eu.org*.

7.1 EXPERIMENT

The data we use was recorded in the course of the SENSATION subproject SP1, work package WP 1.7, pilot 2.5 under the objective "The Design and Placement of milled rumble strips on Swedish rural roads". The project served as a data basis for two different aspects – the effect of different rumble strips on fatigued drivers and the collection of physiological data of the aforementioned in a fatigue driving situation. The pilot was carried out using a third generation moving base driving simulator to obtain results in a realistic driving environment without any safety concern.

The according experiments have taken place in the Swedish National Road and Transport Research Institute *VTI (Statens väg- och transportforskningsinstitut)* in collaboration with the leading Swedish medical university *Karolinska institutet*.

The study features 38 subjects which fulfilled the inclusion criteria, 19 females and 19 males respectively. Since it was necessary for the experiment that all subjects are sufficiently sleepy, all persons were shift workers, coming directly for the testing after a full night shift

without sleep. No professional drivers have been included in the driving simulator study. The main task was then to drive between 45 and 90 minutes without any communication or distraction.

7.2 RECORDING

Upon arrival at VTI all subjects had to undergo a pretest session to get basic information on their overall vigilance, including:

- biocalibration
- reaction time test
- pupillometry
- Karolinska Sleepiness Scale
- Epworth Sleepiness Scale

In the driving phase, there has been recorded:

- EEG, EOG and EMG
- Karolinska Sleepiness Scale
- driving behavior
- eye gaze and blinks (with *Smarteye* detection system)
- camera DVD recording

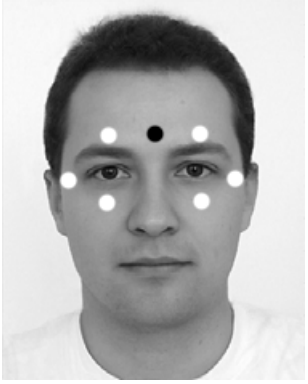


Figure 7.1: EOG electrode position in driving simulator experiments, black dot marks ground electrode, portrait taken from C. Braun, M. Gründl, C. Marberger, and C. Scherber, "Beautycheck - Ursachen und Folgen von Attraktivität. Projektabschlussbericht," 2001

The electrode position used for the EOG recording was a three channel setting with one horizontal and two vertical channels, as depicted in figure 7.1. It has been worked with silver cup electrodes together with a portable digital recorder of the type *Vitaport 2* from *TEMEC Instruments BV*. The sampling rate was chosen 512 Hz with an amplification factor of 1000 and cutoff frequencies of DC and 70.1 Hz.

To compensate for baseline drifts, who are an effect of the DC-recording of the EOG signal, the signal was nulled manually about every five minutes. This can be seen especially in the horizontal channel – see figure 7.2. In the following 6.3 seconds (on average) after every nulling, there was no recording taking place, so for further calculations this has to be kept in mind!

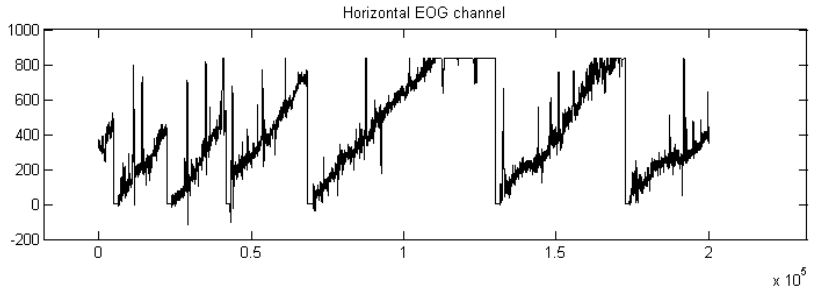


Figure 7.2: fragment of the raw-EOG-signal of dataset `fp01` which shows the baseline drift of the signal and the nulling process which sets the signal back to zero

7.2.1 *The KDS*

In contrast to the self-rated Karolinska Sleepiness Scale (KSS) [111, 112], there exists another externally rated measurement derived from the driving data, the *Karolinska Drowsiness Score* (KDS). Therefor the signal is divided in (non overlapping) 20 second windows, which are divided again in 10 2-second epochs. Each of these epochs has been scored visually to decide if it shows signs of sleepiness – slow eye blinks, small amplitude blinks, blink frequency, alpha and theta activity. Every window is now labeled a number between 0 and 10, indicating the percentage of drowsy epochs in one window.

But Jammes et al. have discovered on the exact same dataset that the visual scoring done by the Karolinska Institutet did not work as accurate as their specially developed detection algorithm [55]. This was mostly due to the differentiation between normal and long blinks. The algorithm classified them by a fixed threshold, whereas the physicians did that by intuition.

Alarmed by that fact and the circumstance that the original KDS scale did not at all agree (even with the overall trend) of our EOG-features, we decided to calculate the KDS anew, using an automatic blink detection algorithm. Our choice fell upon the MATLAB® toolbox *eogui* [113].

The toolbox is straightforward to use. First one hast to define the test setting to define the viewing angle. Due to a lack of knowledge we decided on values which are similar to driving simulator settings. The toolbox also needs signal characteristics like one model saccade, one model blink and the background noise, which are all calculated after marking according positions in the signal by hand. As blink defining parameters we settled for

minimal amplitude	2
threshold softening	0.3
maximal delay	180 ms

These values give us a number of detected long blinks which are comparable to those of [55], as can be seen in table 7.1. So we calculated again the KDS values for the obtained number of long blinks, which yield better visual correlation to the EOG-data. The KDS, as a not-subjective indicator of sleepiness, will be our number one reference for the Hidden Markov Model we develop.

	Jammes et al.	eogui	difference
`fp_01`	1828	1807	21
`fp_02`	2313	2296	17
`fp_03`	1615	1631	16

Table 7.1: comparison between the results of Jammes et al. and our detection method based on eogui for the first three test subjects

7.2.2 *Rumble strip hits*

Another measurement from the driving simulator experiment – which was also part of a study for different rumble strip designs – was a marker, when a driver hit the rumble strip at the curbside. In that case we can strongly assume that the driver was inattentive if not falling asleep, but at least with reduced vigilance. We will display that marker too in our plots of the final results.

7.3 FEATURE SELECTION

We are provided with MATLAB files containing the EEG and EOG data from the driving episodes of all subjects. All subjects (Swedish: försöksperson) have been assigned a serial number, which leads to the file denomination "`fpXX`", where `XX` stands for a number of 01 to 44.

For the signal processing part we used MATLAB® because the original data was in their proprietary file format.

7.3.1 *Amplitude, Velocity, Frequency*

The first three features we calculate are the amplitude, velocity and frequency of the eye movements. Like stated above, sleepiness has some effects on the amplitude and velocity of the eye movements. Hanke et al. also showed the correlation between the actual vigilance state of a subject and those features [61]. It is stated that the amplitude and velocity of the EOG recordings is negatively correlated (sinking when less vigilant) while the frequency spectrum shows the exact inverse behavior.

As stated above, the horizontal EOG channel is unfortunately not well suited for such long term observation (figure 7.2), which leaves

all EOG-based features being derived by the vertical channels. Since in healthy subjects the eyes move in accordance of one another, the two vertical channels should give approximately the same values (thus often only one vertical channel is recorded). We decided to use the signal of the vertical channel of the left eye for all future calculations.

AMPLITUDE The amplitude is simply the recorded current fluctuation.

VELOCITY The signal velocity is computed by simple numerical differentiation.

FREQUENCY In order to obtain the signal frequency, the Fast Fourier Transformation (FFT) was applied to overlapping signal windows in the time domain.

Unfortunately these three features are highly correlated. This is shown in the correlation coefficients between the different features in table 7.2. Such dependent input vectors would limit the discriminability of the different model states in the Hidden Markov Model. For that reason we are going to implement some external measures collected from sensors in the simulated car, which we call "driving data".

`fp01`	amplitude	frequency	velocity
amplitude	1	0.8932	0.6078
frequency	0.8932	1	0.5358
velocity	0.6078	0.5358	1

Table 7.2: Correlation matrix of the three feature vectors of dataset `fp01`.

7.3.2 *Driving data*

In the driving simulator study there have also been recorded car-based signals from sensors in the drivers cabin and outside. Two often used signals in automotive on-board hypovigilance detection systems are the steering wheel angle and the actual position of the car between the lane markings [80, 81]. Those signals have been recorded with 25 Hz.

CAR POSITION To hold the car in the same relative position to the lane markings needs a certain amount of vigilance (negative examples: fatigue, drunk driving). So the measure we are looking for is the variance of the car position in a certain (moving) time window. We used a window length of 100 and applied additional moving window smoothing to the resulting time series (window length = 100).

STEERING WHEEL ANGLE The most revealing part here is the velocity of turning on the steering wheel since abrupt movements

correspond to potentially hazardous situations, causes mostly by inattentiveness. As input parameter we take therefore the smoothed (window length = 30) numerical differentiated signal of the steering wheel angle.

For being able to compare our HMM with the KDS-values we calculated, we have been forced to resample the recorded data to 20 second windows, which makes the model rather rough. Out of cosmetic reasons, we also rescale the features that we pass on to the Hidden Markov Model to fit the interval $[0, 1]$ before saving.

8 IMPLEMENTATION

We now have so far gathered all theoretical tools we have to know in order to be able to create a program which uses afore mentioned algorithms to fit a Hidden Markov Model to our data. But as everywhere, there are some hurdles to jump when advancing from the theoretical to the practical side. Here we want to point out the main points and pitfalls when implementing the algorithms.

For reasons of traceability and clarity the entire source code is enclosed in appendix A.

8.1 THE PROGRAMMING LANGUAGE: R

R is an open source statistical computing language created by Ihaka and Gentleman on the base of the statistical programming language S. The program source code as well as all available toolboxes underly the GNU General Public License and are therefore freely available and customizable. The built-in function's source code is mostly written in Fortran and C, making it fast and memory-saving. One of the upsides of R is also the graphics engine, capable of creating high-quality fully customizable plots.

Despite MATLAB® being the number one software in the field of mathematical computing and machine-learning (quasi-standard), we decided on building our program in R due to the advantages of open source software like the free availability.

8.2 INPUT FORMAT

Many different factors could play a role in favor for a certain data exchange format. In our case it was necessary to transfer the processed data from MATLAB® to R where we run the HMM code. The choice fell upon the simple `.txt` format, because it is platform independent, lightweight and easily writable/readable by both programs.

After data processing as described in chapter 7, we saved the obtained features in files with the name `input_HMM_fpXX.txt`, where `XX` again stands for the subject serial number. Every feature – all of them have been brought to the length of the KDS-vector – is saved as one column in the input file, resulting in a structure described in table 8.1.

	Amplitude	Velocity	Frequency	Position	Steering wheel	KDS	Hits
O^1							
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
O^T							

Table 8.1: Schematic arrangement of the input format for the program. Rows correspond to observations (same length as KDS) and columns 1-5 correspond to the feature vectors. The last two columns are the reference data.

8.3 R-PACKAGE: RHMM

One of the advantages of R is the possibility to easily add new functions by installing R-packages. At this place I want to express my gratitude to all the people who share their work generously with the whole world. This is a huge help for many of us!

We decided that the package *RHmm*, written by Taramasco and Bauer, would fit our needs best because it is able to work with multivariate Gaussian Mixture Models and is coded mainly in c, which gives it an edge regarding calculating speed [114]. For our purposes we only need the two central functions of the package: `HMMFit` and `viterbi`.

HMMFIT This function takes the input data and runs the Baum-Welch algorithm on it for optimizing the model parameters. The type of distribution, the number of mixtures and the number of states must be set as input. It returns the fitted parameters (means and covariances, transition probability matrix, initial probability) along with the log-likelihood and the information criteria AIC and BIC. Model initialization is done – unless otherwise specified – by a (randomly initialized) k-means algorithm on the input data.

VITERBI This function takes the output of `HMMFit` and uses the fitted model parameters to calculate the most probable state sequence using the Viterbi algorithm. It returns the state sequence alongside with two other probability measures.

All in all this package proved to be *much* faster than the self implemented code in R. The fact, that this package is only able to use mixture models with Gaussian distributions is not disadvantageous in our case, since our data does not necessarily support the presumption of non-Gaussian distributions.

Detailed information about the package, its functions and examples can be found in the documentation of the package (see [114])

8.4 MODEL INITIALIZATION

After deciding on the type of the model (discrete/continuous, distribution, mixtures) and the number of states involved one has to focus on how to initialize the algorithm. It lies in the nature of the EM-algorithm, that there exists the probability to get stuck in a local maximum of the likelihood function.

Zucchini and MacDonald indicate in [89] two different methods of initialization:

A. A costly way of eliminating local maximums in favor of the (presumably) global one is to randomly initialize at many different points and compare the model results. The best one is kept as the global maximum. A more elaborate way to do this is to use a clustering algorithm with random seeds to find neuralgic starting points. So the number of different initial points can be reduced to a sensible figure. We are following that procedure with the RHmm package by recomputing the whole model including the random k-means initialization 50 times and keeping the best model.

B. The more efficient way is to set the off-diagonal transition probabilities to very small values (e.g. 0.01) and arrange the state means equally spaced about nearly the range of the observation data.

In our proper code, we used the optimal results coming from the RHmm package as initialization values to finally compare the fit of the Gaussian model to the Student-t model.

8.5 MISSING VALUES

As in all practical recording situations, not all the data is usable. This may be due to electrode detachment, large artifacts, instrument failure or data corruption. Since such situations can not be avoided at all cost, it is necessary to implement precaution measures which deal with missing or corrupted data.

In our case, we got already checked data from the VTI, so that we had not directly to deal with that matter. Nevertheless it is strongly advisable to cover that cases – and it is absolutely essential when programming for online applications!

One easy way to deal with singular defects is to interpolate the signal in order to restore values for the NaN-entries. For larger defect ranges, since it would not be possible to get interpretable results, the program should be halted and an error message should be displayed. After checking the equipment the program could be resumed.

8.6 NUMERICAL STABILITY

When inventing an algorithm, one of the first tests it must pass is the one of theoretical stability (e.g. Lyapunov). Of course the algorithms for HMMs outlined in chapter 6 fulfill this standard. But when it comes to implement an algorithm on a computer a new problem can rise very easily – numerical instability. Stoer defined numerical stability with following words:

> "An algorithm is numerically stable on some input set if the effect of round-off errors in the computations is comparable to the effect of round-off errors in the input data and the output data, independent of the particular input element." [115]

Dealing with algorithms which are working with products of probabilities, it is clear that one has to counteract a steady decrease of values down to a range of machine accuracy, where roundoff errors set all future results equal to zero. To avoid that, we already spoke about the concept of scaling and log-likelihood in the theoretical chapter 6. The scaling addendum keeps the probability at the range of the initialization, whereas the logarithm of the likelihood enables to evaluate the likelihood of a computed model.

Another pitfall is, that the mathematically easy concept of an inverse matrix entails a few problems when done numerically. This is all the more grave, when the matrix is ill-conditioned, i.e. nearly singular. We cushioned that issue by using the pseudoinverse matrix, which is already a built-in function in R, relying on *singular value decomposition* (SVD).

We also encountered difficulties with the adapted EM-algorithm for Student-t distributions described in [106]. There we had the problem that the estimations of the covariance matrices have sometimes been singular matrices, i.e. non-invertible. Therefore we decided to use a method of covariance regularization which is also known under the name of *covariance shrinkage*. The method is depicted in [116].

For a more detailed view on numerical problems and a more formal definition of numerical stability, see [117, 115].

8.7 OVERFITTING

In all machine-learning tasks one has to be wary of overfitting, i.e. adapting too much to the training data (and also their measurement errors) so that the classification of new samples gets worse. This happens frequently when only small sample sizes are available – like it is in our case, since we had to downsample our data in order to match the reference data. In the case of HMMs the most influential parameter on the model fit is the number of states (see section 6.4).

Keeping the number of states low also keeps the number of parameters low which antagonizes the overfitting. In addition we use only one distribution per state, which also lowers the parameters (see section 9.1). For that simple model the number of observations is sufficient to fit the distributions.

Of course there are also other reasons how and why the model could overfit, e.g. unfortunate initialization could lead to one state overfitting some extreme data points. Whenever the norm of one of the covariance matrices is very small compared to the others one should investigate that process.

8.8 REFERENCE DATA

As already stated in chapter 7, we are using the KDS and rumble strip hits as reference data to evaluate the model. For measures as the KDS, which are based on eye blink behavior, it is advised to use intervals of at least 60 seconds length in order to obtain an interpretable and not erratic signal. It was shown by Sandberg that the 1 minute window gave significantly better results as shorter ones (10 - 30 s), whereas the difference to longer ones (120 - 900 s) seemed small [118].

We did follow this rule of thumb by computing the mean of the KDS score in a moving window approach (window length = 60 s) over the last five 20-second bins. This function is then displayed over the model states, all confined to the unit interval.

When it comes to the rumble strip hits, which are binary data – True or False – we cannot use that sensibly for constructing a function. But we will use that knowledge to mark the time points in the model where the drivers hit the rumble strip. As this is clearly a sign of (preceded) hypovigilance, the model should also indicate a high fatigue level.

Part III

RESULTS AND DISCUSSION

9 RESULTS

This chapter presents and discusses interim and final results we yielded in the process of this work. Detailed explanations are provided to explain why we set our assumptions the way they are. The field of applied biomedical engineering is not at all easy to master, and practically never universal solutions can be found. So the intention is to clearly delimit what our model is made for, and what it is capable to achieve.

Like shown in section 7.2.1, we have been able to reproduce sensible KDS values for the three datasets `fp01`, `fp02` and `fp03`. Due to this reliable reference data, those datasets are used to calculate the following results.

9.1 MODEL CHOICE

Even with the concepts perfectly understood and the program running, one has nonetheless to start with physiological characteristics, mathematical reasoning and not at least a good measure of intuition to come up with an effective and sensible model.

9.1.1 *Distribution*

The first question to answer is the one of the distributions used in the Hidden Markov Model. Since we are dealing with continuous measurement data it is only logical to make use of continuous HMMs with continuous densities. In the following we want to go through all input features, study their histogram and discuss why we finally decided on a Gaussian HMM.

EOG DATA When taking a look at the histograms of the EOG-based features in figure 9.1 one remarks the heavier tail on the right side of the distribution. But we must not be mislead by the heavy tail on the right side of the data distribution, since those histograms have been created using all recorded data, independent from their assigned state. That means that the histogram is a mixture density from whatever how many states we assign the model.

DRIVING DATA The lowest two histograms in figure 9.1 look more like they would have been derived from beta-distributed data. This comes because the normal values of both the variance of the car in the lane and the steering wheel velocity are rather low and steady. Only when getting so tired that it needs a quick

reaction to stay at the street, the values of those two measures rise abruptly. That means – since such different samples will not belong to the same state – that we can again model those distributions as a mixture of normal distributions

But what we cannot deduce so easily from the histograms, is the answer to the question if we would benefit from a Student-t Hidden Markov Model. To find out, we ran the Gaussian HMM on all three datasets (using a three-state model) and took the respectively optimized model parameters to feed the SHMM (one example is located in the source code section A.2).

As can be seen in table 9.1, the Student-t HMM has a slightly decreased logarithmic likelihood for all three test subjects. This does not give us a 100% certainty that the SHMM might not perform superior for a specific parameter set, but all in all the GHMM seems to be the better solution for our problem.

	`fp01`	`fp02`	`fp03`
GHMM	758.55	993.93	672.67
SHMM	679.54	895.66	615.43

Table 9.1: Comparison between the logarithmic likelihood produced by a three-state GHMM and SHMM.

In the context of online application of EOG-based devices on the other hand, SHMMs could prove to be a valuable extension of GHMM, because they are much more resistant to outliers and bad initialization [106].

9.1.2 *Mixtures*

Now that we settled for GHMMS, the second question is how many Gaussian mixture densities should represent each state. Our decision to use only one normal distribution per stage – making the mixture model a simple Gaussian HMM – is based on following considerations:

A. From a physiological viewpoint, the transition between different vigilance levels should transfer (roughly) linearly into the data, leaving simple distributions as the easiest way to model that behavior. We do not expect one state to have two different maximums, e.g. the EOG frequency being very high or very low for a certain state, but not average. The same reasoning applies to the driving data. Also the lane variance rises with rising tiredness and abrupt steering wheel motions indicate fighting a high level of fatigue.

B. When taking the maths into account, one sees in figure 9.1 that it should be possible to model the densities outlined by

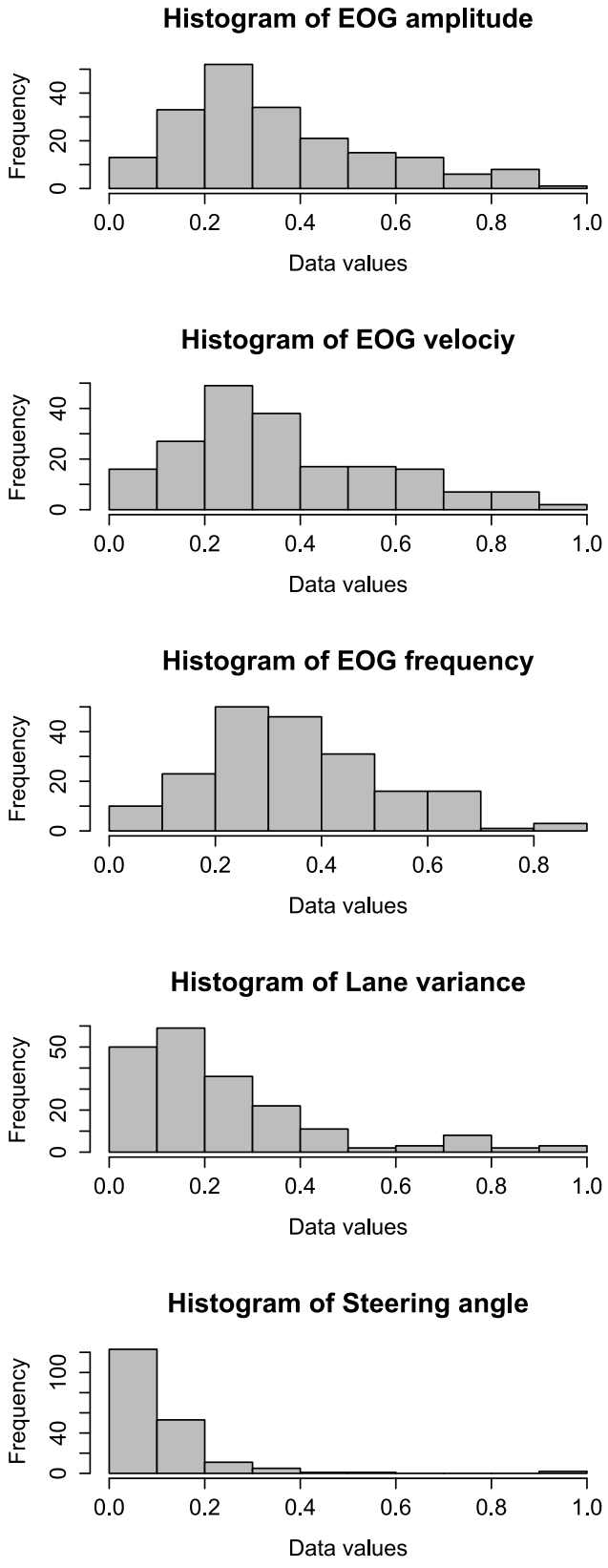
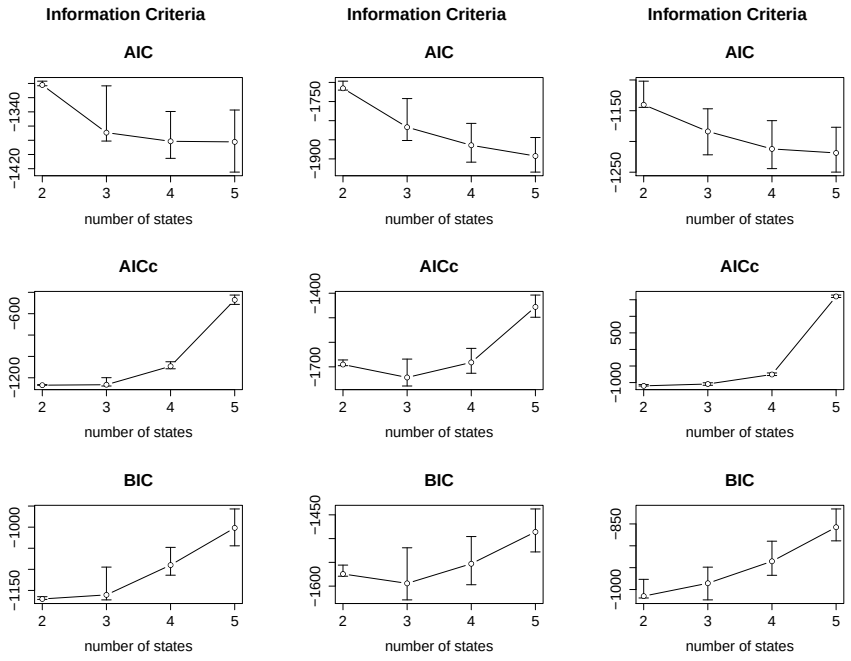


Figure 9.1: Histograms of the five input features over all data

the histograms with a simple GHMM. This train of thought is supported by the figures 9.4, 9.9, 9.8, 9.9, 9.12, 9.13, which show the fit of the model to the estimated densities for each feature. On has to bear in mind, that although the fit is far from perfect, that it will never be when dealing with real-world data (especially with so small sample sizes). In addition, a raise in mixtures per state will also increase the number of independent parameters to estimate, which is unfavorable!

9.1.3 *States*

Now we have fixed the model distributions and restricted them to only one per state. The next question we have to face is, how many states we should use to get the best model for our purposes? In section 6.4 we have already explained the method of information criteria and pseudo-residuals in order to pick the optimal number of states. The results for all three test subjects can be seen in figure 9.2 and the table 9.2.



(a) `fp01` (b) `fp02` (c) `fp03`

Figure 9.2: Information criteria AIC, AICc and BIC for all three datasets used. The whiskers indicate the variability of the measure for models with different starting values. It can be clearly seen that the AIC favors models with a higher number of states while AICc and BIC do not. Best compromise alongside with the physiological interpretation is a three-state HMM.

One can see that the AIC values keep dropping with rising number of features while AICc and BIC have their minimum either for the

2-state or 3-state model. Thinking about the robustness of the AICc measure for small observation sets, we will base our decision mostly on that criterion. While for `fp01` and `fp03` the AICc value increases slightly when going from two to three states, for `fp02` it has its minimum at three states. The three-state model also allows us to introduce a transition state between "rather tired" and "rather alert", which helps in terms of usability and interpretability.

`fp01`	# parameters	-LLH	AIC	AICc	BIC
2 states	43	694.09	−1302.18	−1269.28	−1170.22
3 states	68	785.73	−1369.22	−1264.95	−1160.53
4 states	95	785.73	−1381.45	−1091.93	−1089.91
5 states	124	815.13	−1382.26	−470.49	−1001.71

`fp02`	# parameters	-LLH	AIC	AICc	BIC
2 states	43	901.22	−1716.43	−1691.54	−1575.47
3 states	68	975.54	−1815.08	−1741.19	−1592.17
4 states	95	1031.63	−1873.26	−1690.86	−1561.83
5 states	124	1072.09	−1896.18	−1459.56	−1489.7

`fp03`	# parameters	-LLH	AIC	AICc	BIC
2 states	43	613.15	−1140.31	−1099.17	−1015.06
3 states	68	659.85	−1183.71	−1043.65	−985.64
4 states	95	701.07	−1212.14	−756.14	−935.44
5 states	124	733.34	−1218.68	1599.5	−857.51

Table 9.2: Information criteria, negative log-likelihood and number of independent model parameters for the Gaussian Hidden Markov Model for all three subjects

The interaction between the states, which is described by the state transition probability matrix $\mathbf{A}$ has no real constraints, leaving it open to reach every state at every time point, i.e. ergodic model. This is not contrasting reality, since it is for example possible to go from nearly asleep to highly alert in parts of a second due to a sudden event. Of course, we expect most of the changes to be much smoother.

Before presenting the final results, we have to point out, that it is not at all easy to assign a level of fatigue to every model state. Especially in our case, where we take advantage of letting a k-means algorithm find the best initial values, state labels are not always the same, e.g. "state1", "state2", etc. depend on a (random) k-means seed.

It appears that the safest way to order the states by vigilance is to use the driving data. The higher the lane variance and the steering wheel angle velocity, the higher is the hypovigilance. States are then ordered by the vector norm of that two features and plotted in relative distance to one another.

9.2 MODEL RESULTS

Like mentioned in the chapters before, we are using the feature sets of three test subjects `fp01`, `fp02`, `fp03`, from the driving simulator study to test if we are able to model sensible vigilance states with a Gaussian Hidden Markov Model. Summing up the previous sections and pre-results we launched our try-outs with following meta-parameters:

- Gaussian distributions
- No mixtures (one distribution per state)
- 2 to 5 States
- Ergodic
- Randomized k-means initialization (best out of 50)

The structure of the results pages is:

TEST SUBJECT

A. Fitted state sequence with reference data
B. Distribution comparison – part 1
C. Distribution comparison – part 2
D. Pseudo-residuals and Q-Q plot

Those results will be topic of discussion in the subsequent chapter 10.

9.2.1 *Subject fp01*

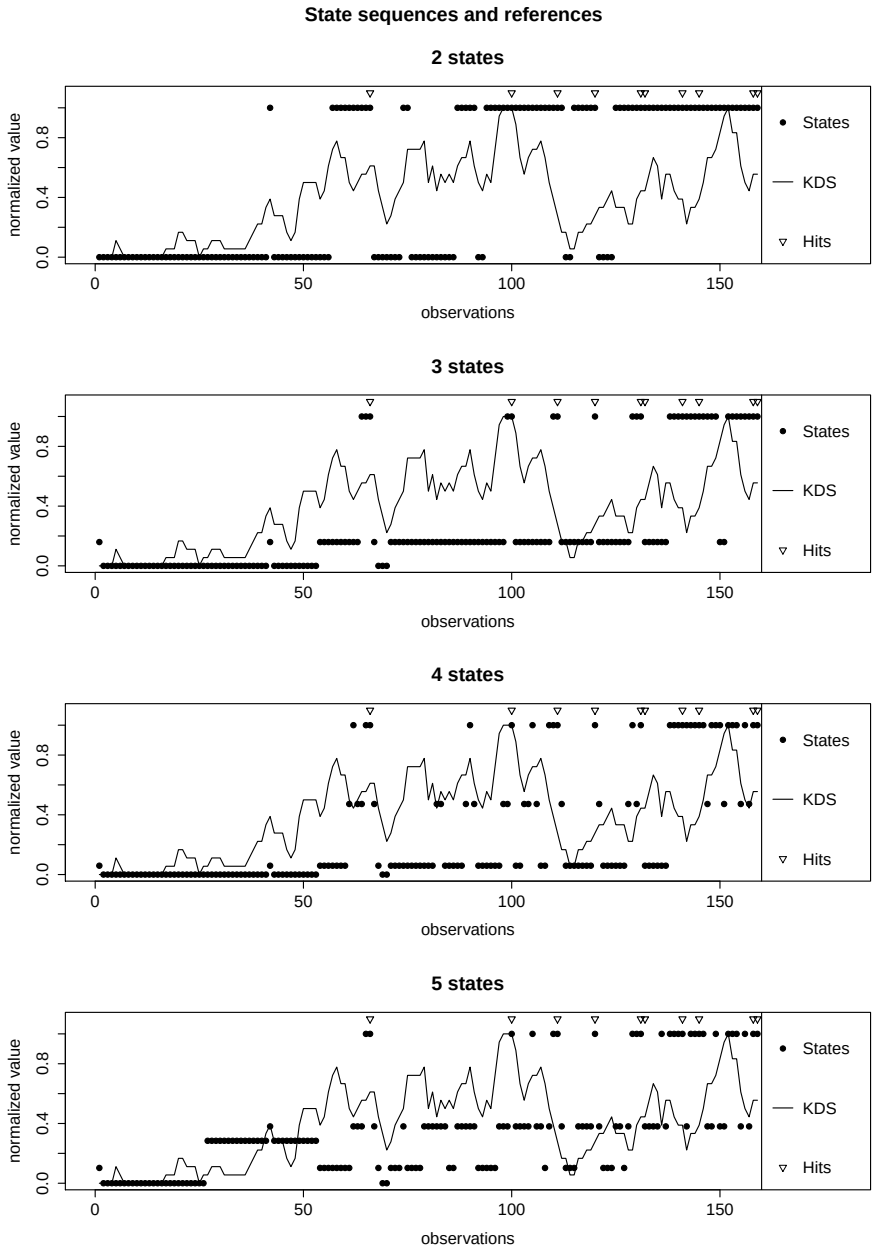


Figure 9.3: (fp01) Fitted models for 2 (top) to 5 (bottom) states. All data (states and smoothed KDS values) was fitted to the unit interval. Black dots are the states (with relative distances to each other), continuous line illustrates the KDS-value and the top-down triangles indicate time points when a rumble strip was hit.

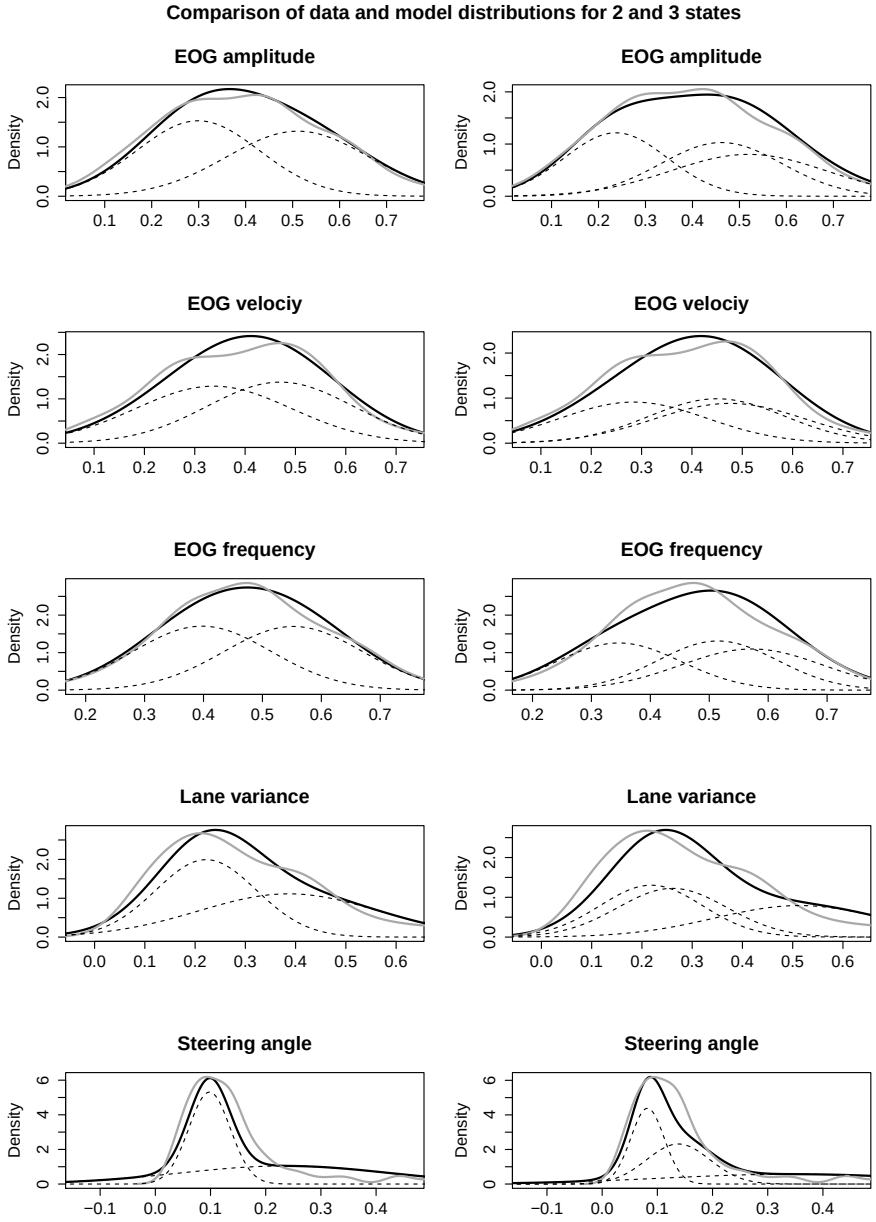


Figure 9.4: (fp01) Comparison of model and data distributions for all five input features for the 2-state model (left) and the 3-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

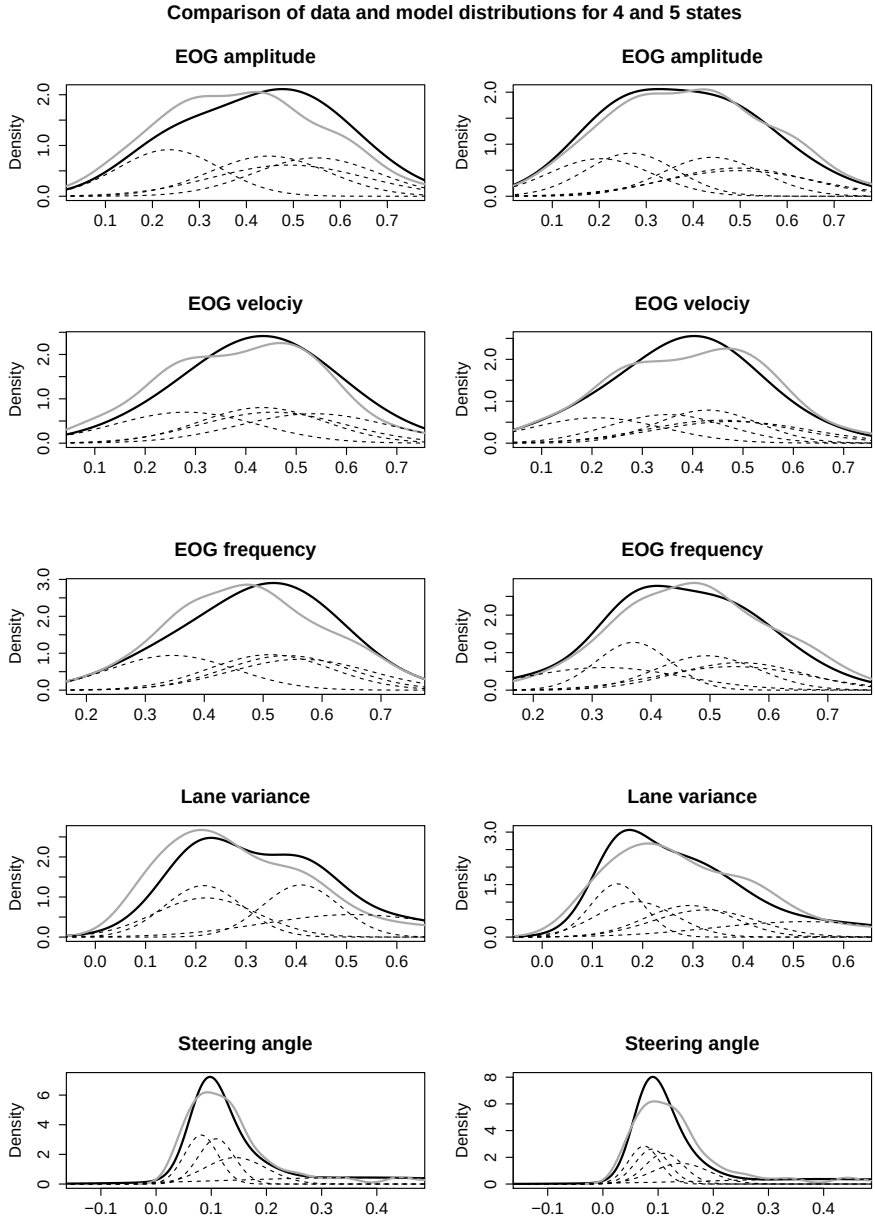


Figure 9.5: (fp01) Comparison of model and data distributions for all five input features for the 4-state model (left) and the 5-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

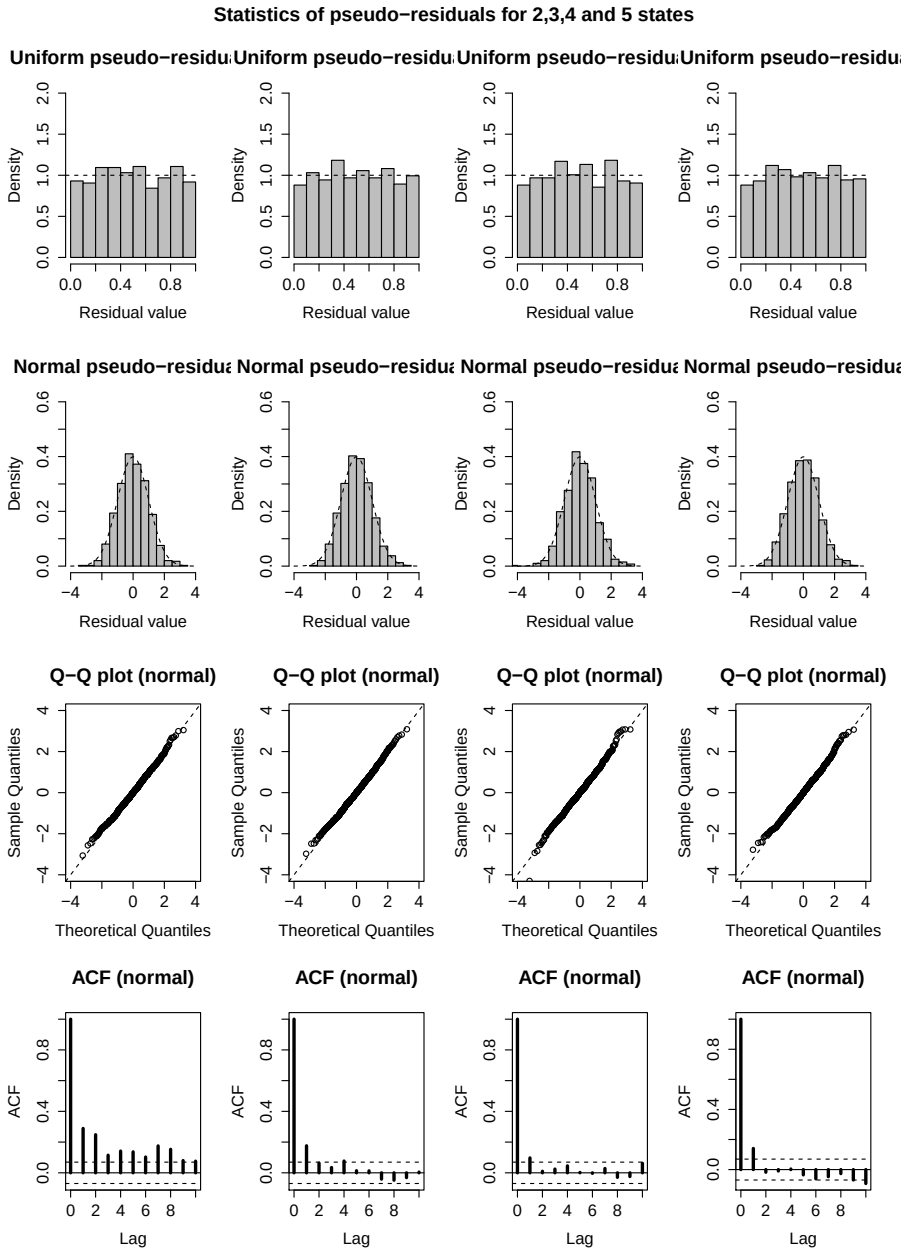


Figure 9.6: (fp01) Pseudo-residuals for models with 2 - 5 states for visual inspection. *First line:* uniform pseudo-residuals *Second line:* normal pseudo-residuals *Third line:* quantile-quantile plot of the normal pseudo-residuals *Forth line:* autoregressive function of the residuals.

9.2.2 *Subject fpo2*

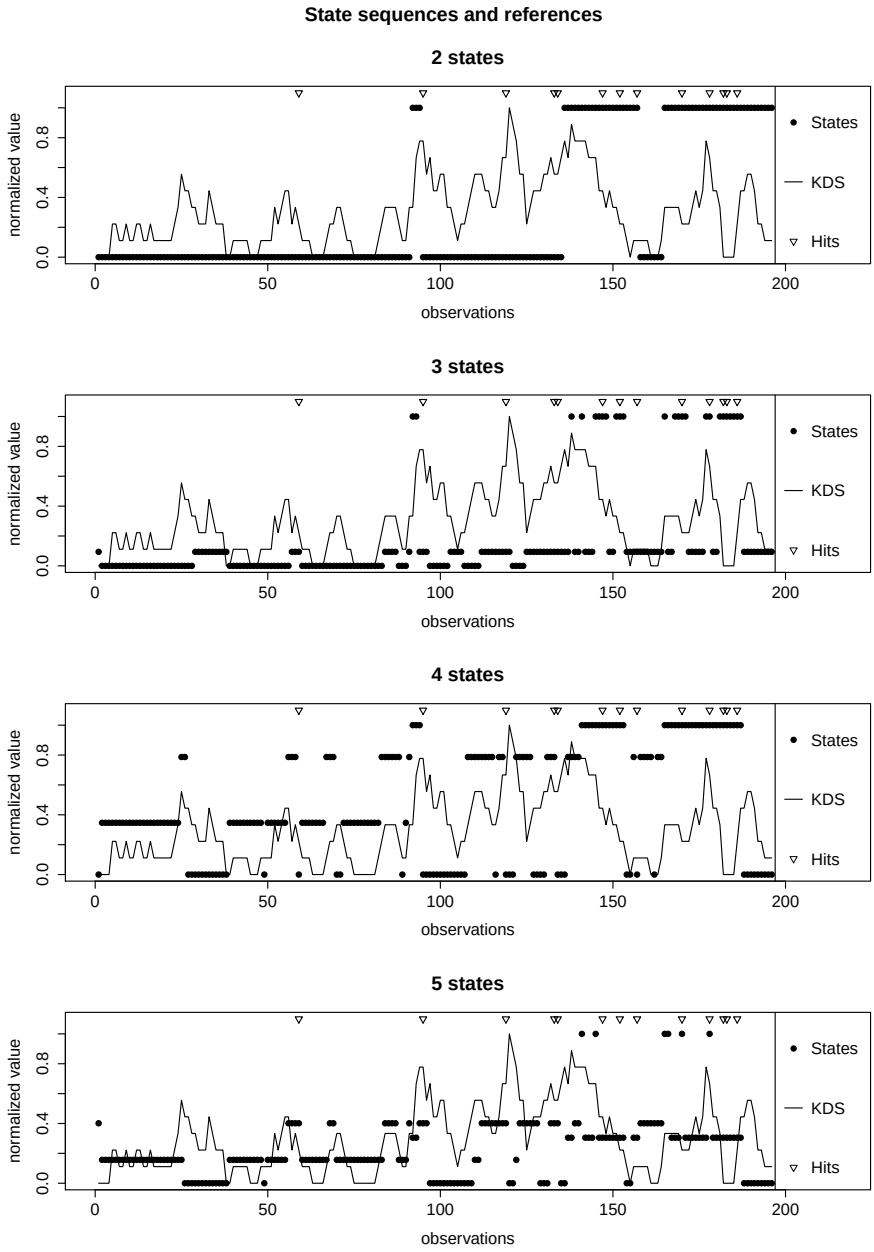


Figure 9.7: (fp02) Fitted models for 2 (top) to 5 (bottom) states. All data (states and smoothed KDS values) was fitted to the unit interval. Black dots are the states (with relative distances to each other), continuous line illustrates the KDS-value and the top-down triangles indicate time points when a rumble strip was hit.

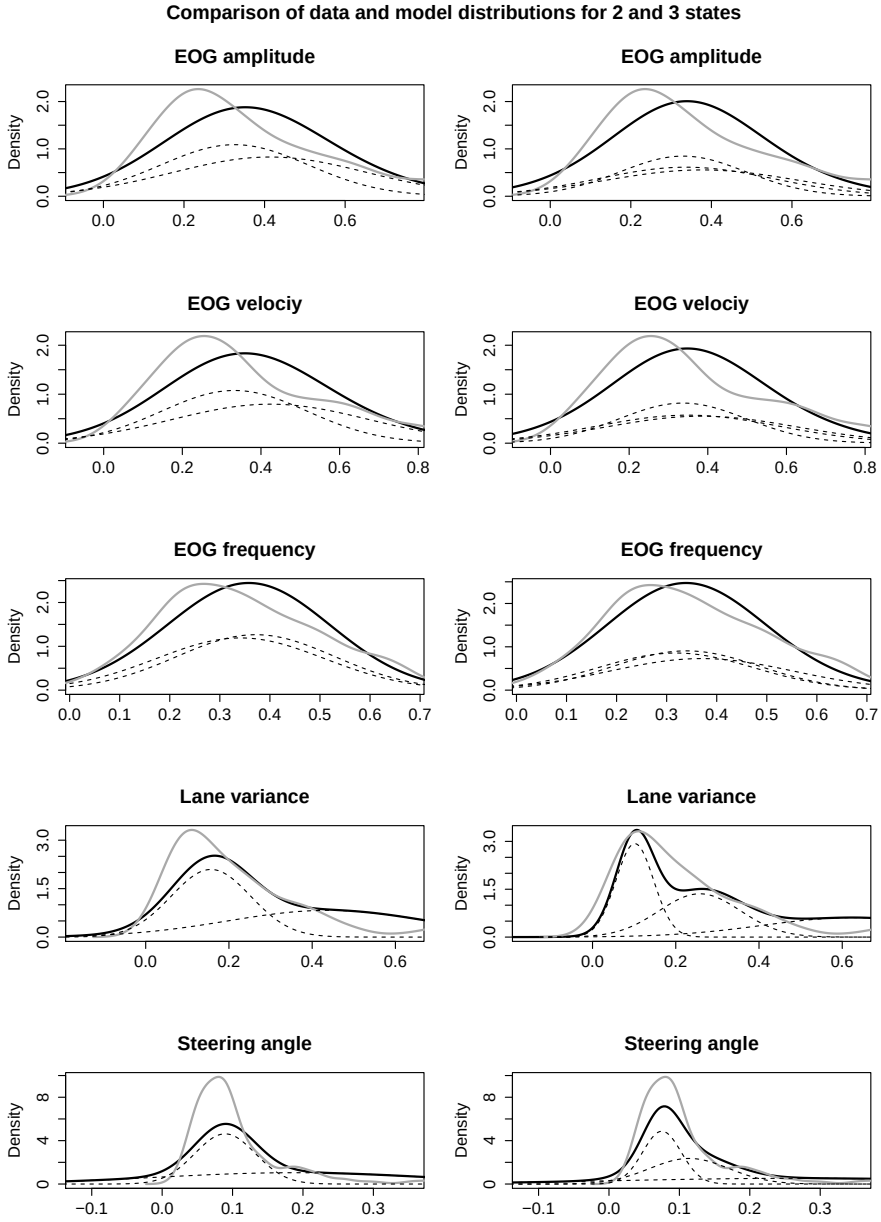


Figure 9.8: (fp02) Comparison of model and data distributions for all five input features for the 2-state model (left) and the 3-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

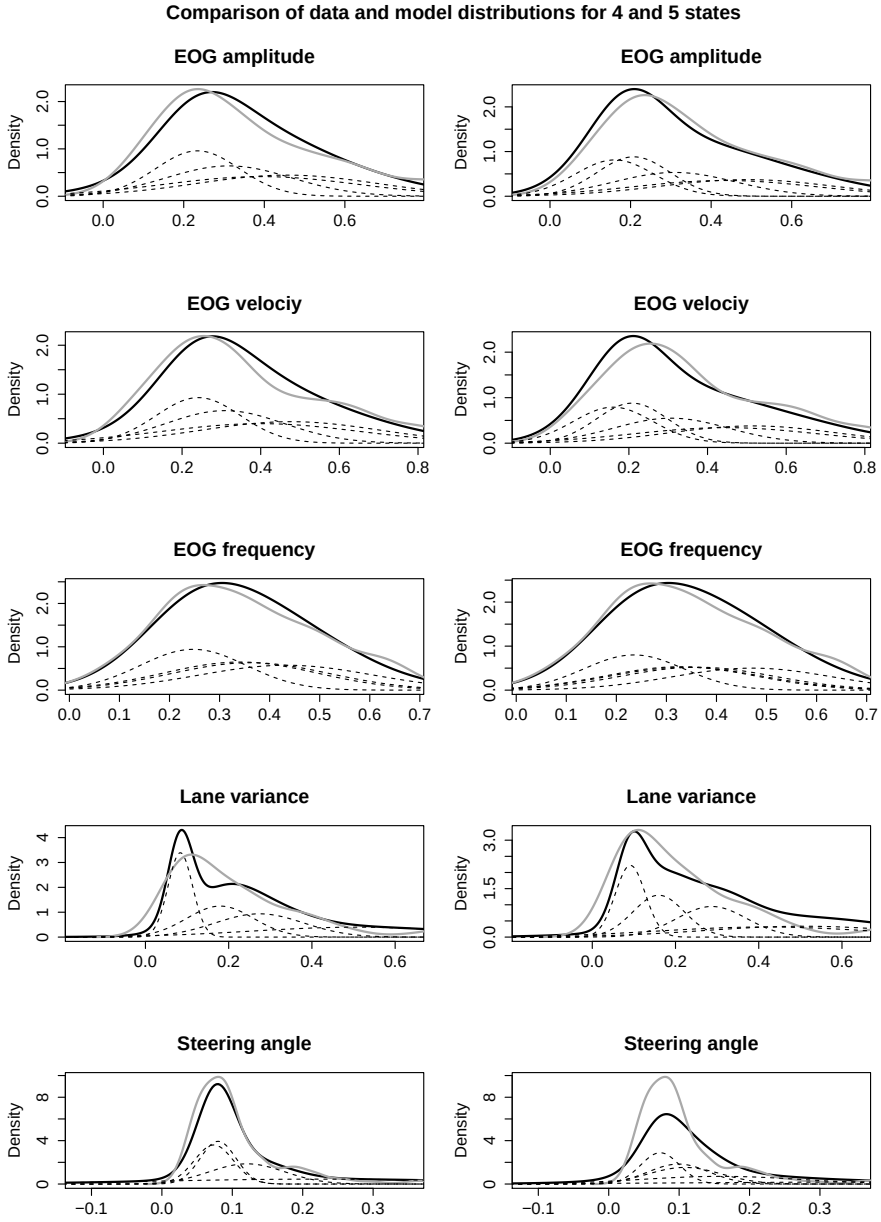


Figure 9.9: (fp02) Comparison of model and data distributions for all five input features for the 4-state model (left) and the 5-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

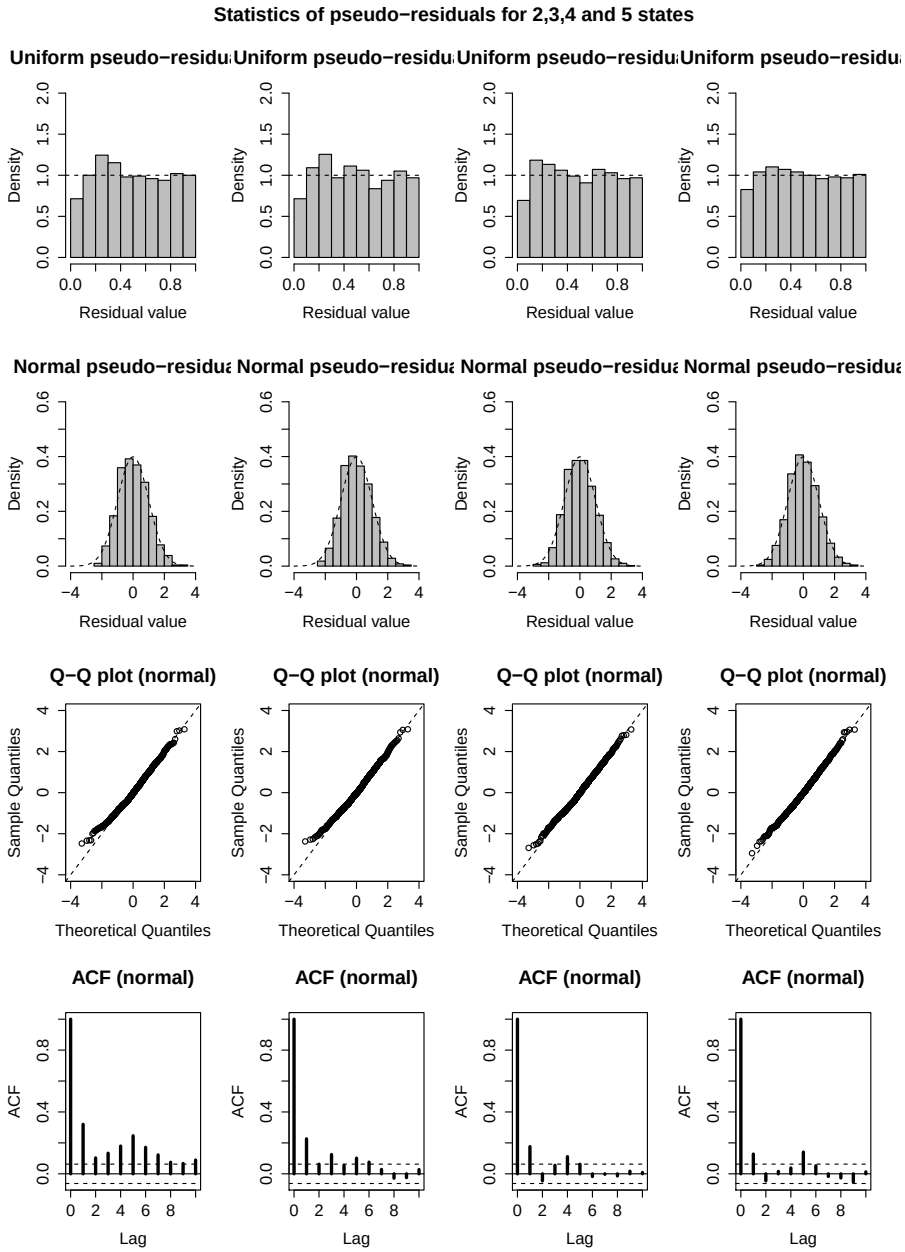


Figure 9.10: (fp02) Pseudo-residuals for models with 2 - 5 states for visual inspection. *First line:* uniform pseudo-residuals *Second line:* normal pseudo-residuals *Third line:* quantile-quantile plot of the normal pseudo-residuals *Forth line:* autoregressive function of the residuals.

9.2.3 *Subject fp03*

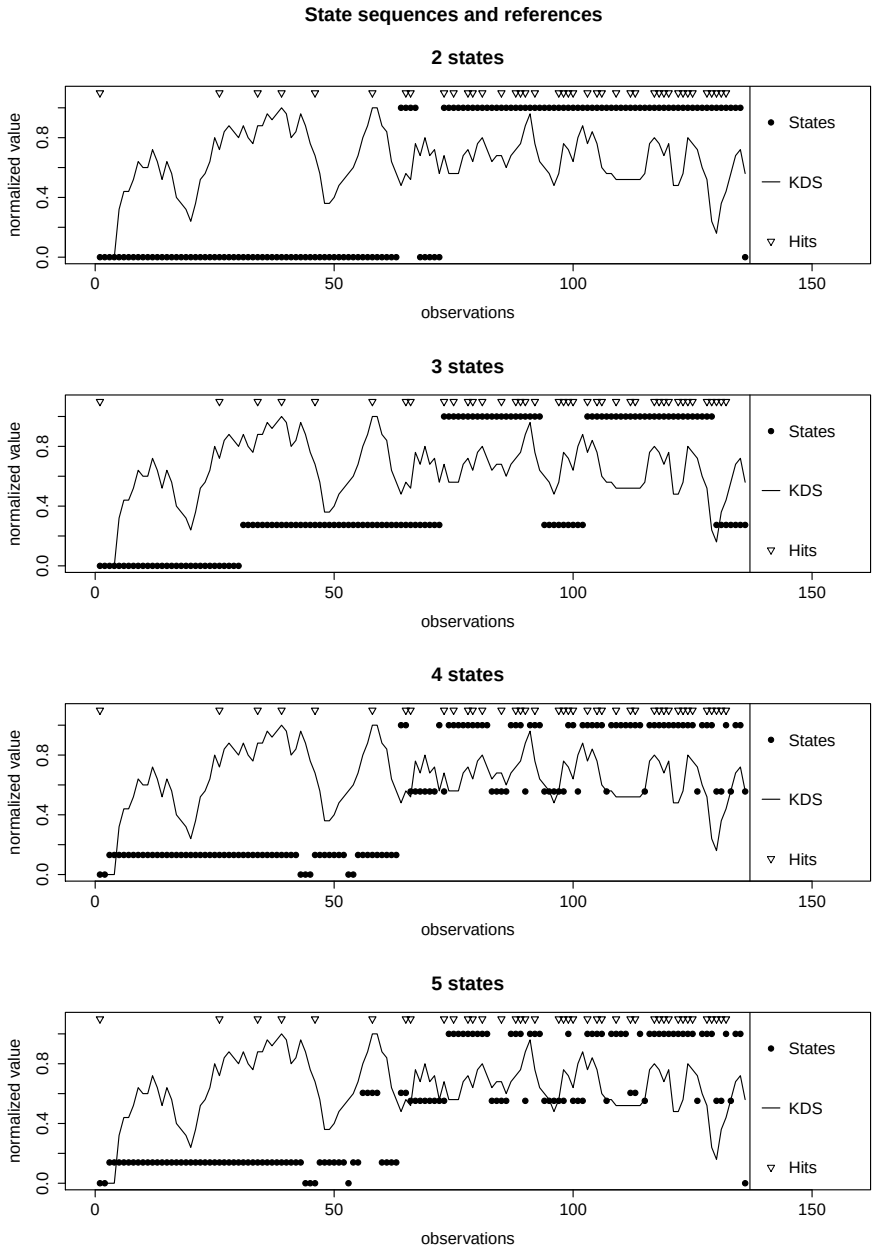


Figure 9.11: (fp03) Fitted models for 2 (top) to 5 (bottom) states. All data (states and smoothed KDS values) was fitted to the unit interval. Black dots are the states (with relative distances to each other), continuous line illustrates the KDS-value and the top-down triangles indicate time points when a rumble strip was hit.

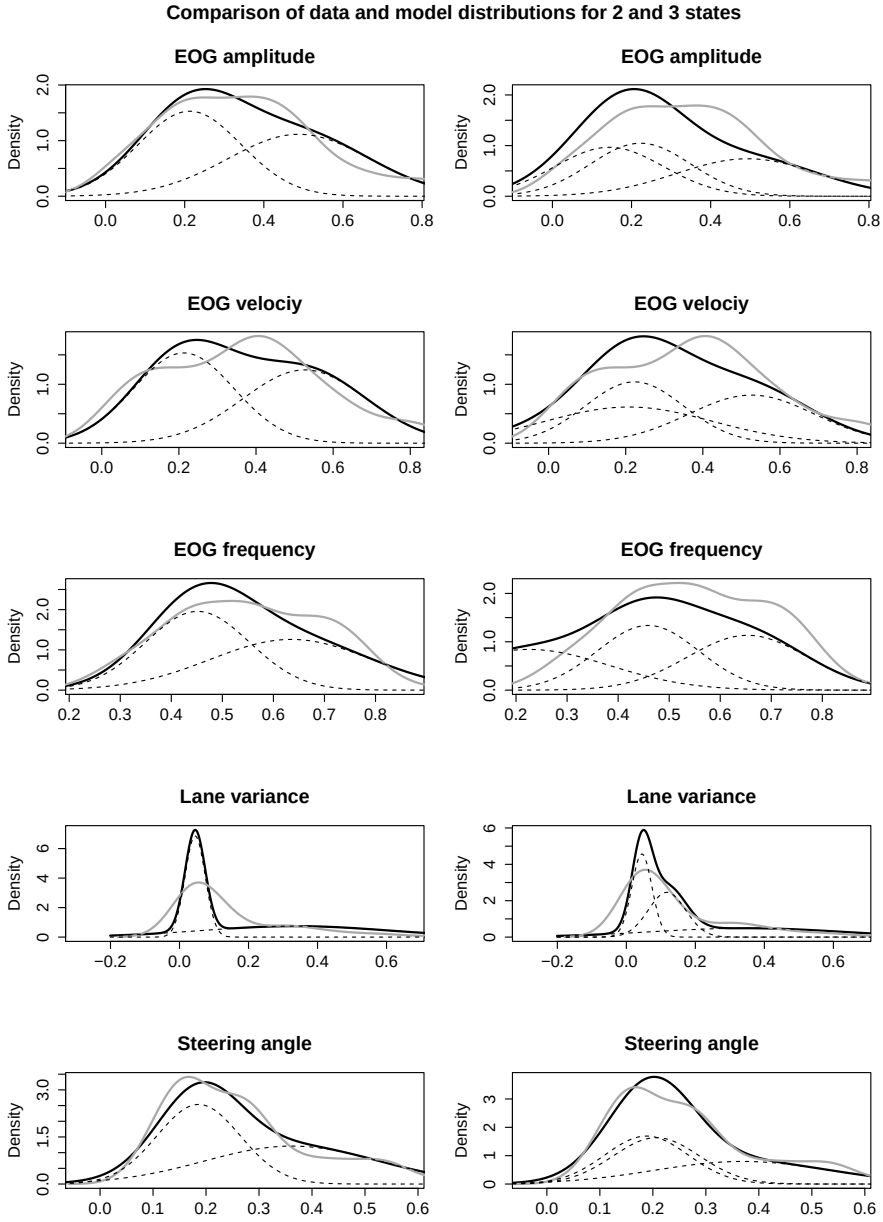


Figure 9.12: (fp03) Comparison of model and data distributions for all five input features for the 2-state model (left) and the 3-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

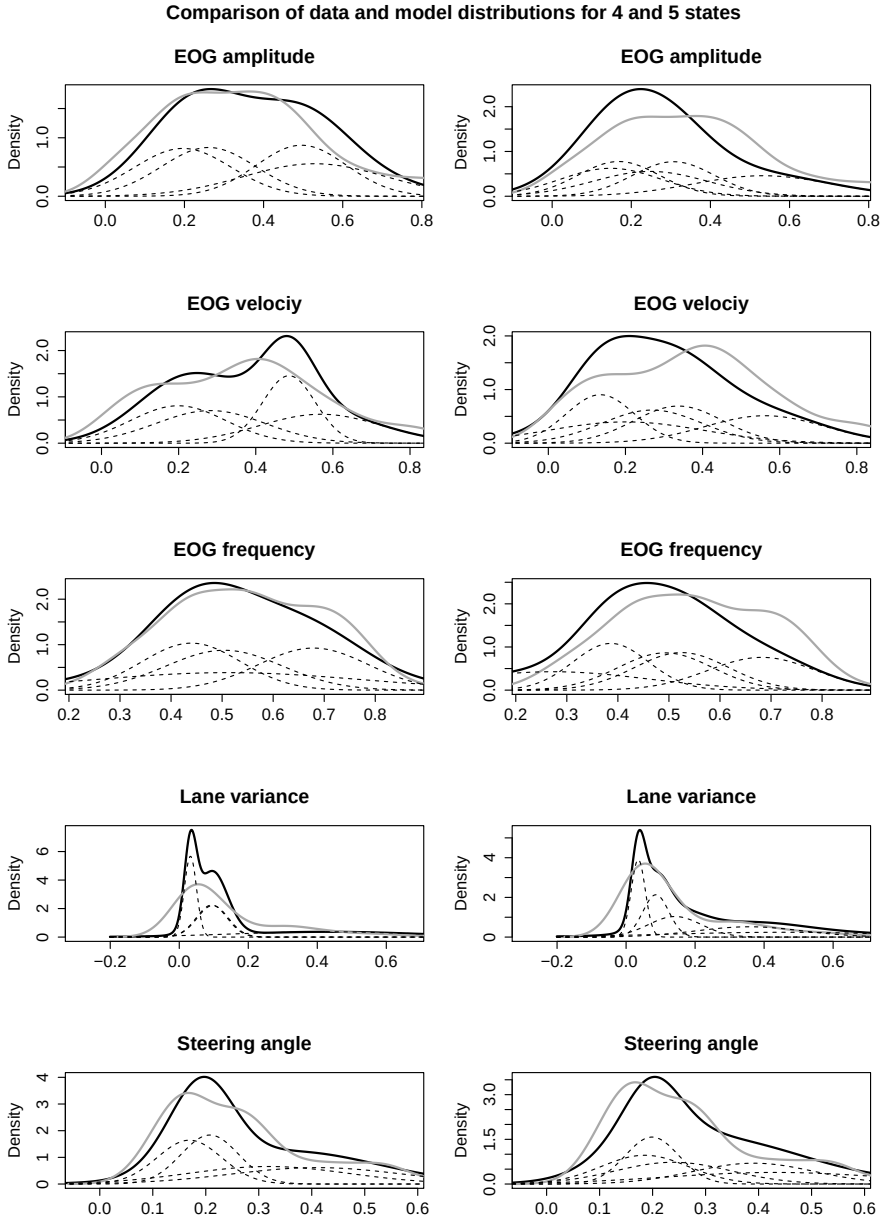


Figure 9.13: (fp03) Comparison of model and data distributions for all five input features for the 4-state model (left) and the 5-state model (right). The thick gray line indicates the estimated overall density of the observations. The thick black line is the fitted overall density of the model and is the sum of the individual state densities depicted by the dotted black lines.

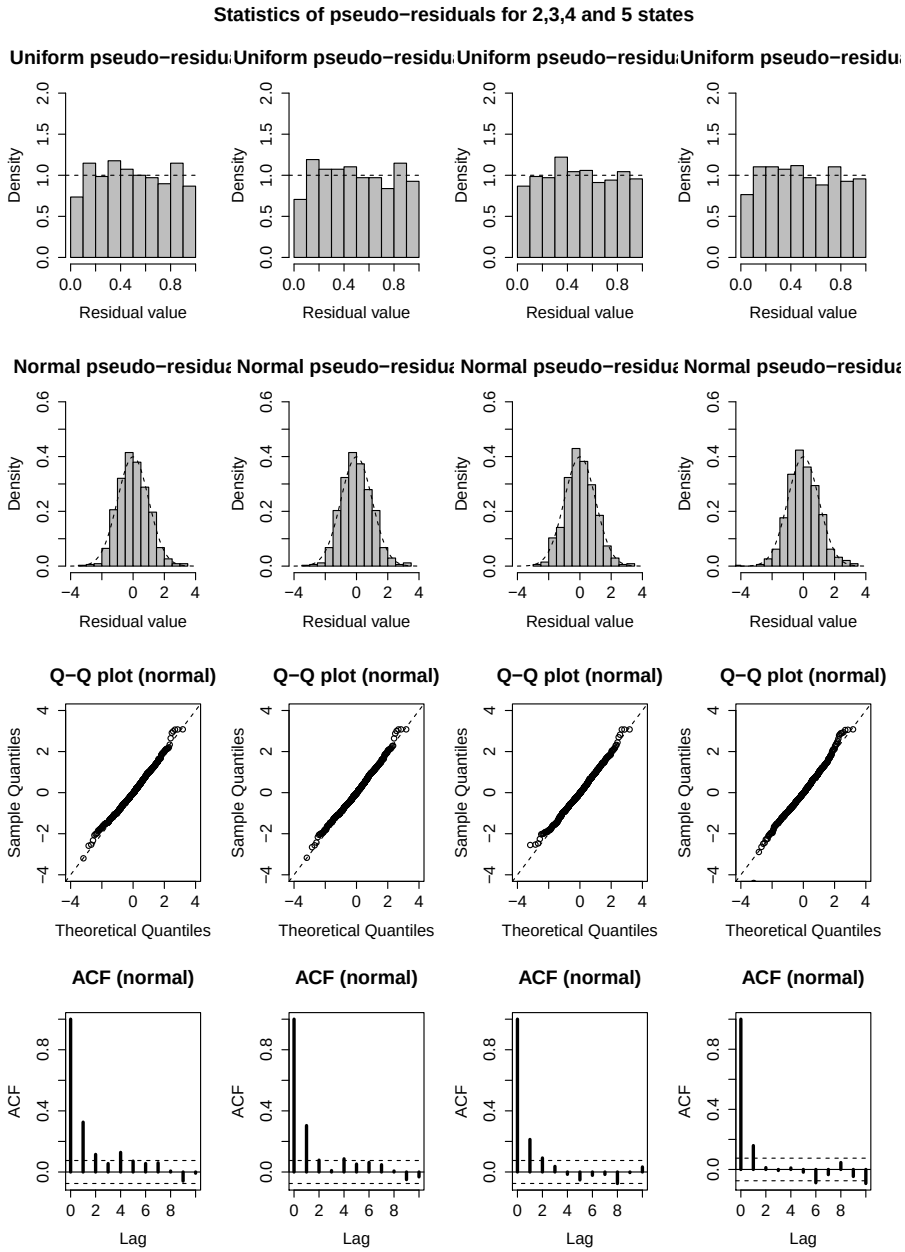


Figure 9.14: (fp03) Pseudo-residuals for models with 2 - 5 states for visual inspection. *First line:* uniform pseudo-residuals *Second line:* normal pseudo-residuals *Third line:* quantile-quantile plot of the normal pseudo-residuals *Forth line:* autoregressive function of the residuals.

10 DISCUSSION

NUMBER OF STATES The fist part of the state-discussion was already held in the section 9.1.3. There we concluded, that regarding the information criteria as well as the physiological interpretation, a three-state GHMM seems to be best suited. Now, is this confirmed in any way by the model results we have got in the last chapter?

The answer is "yes". When examining the plots with the relative positions of the state sequences to each other (figures 9.3, 9.7 and 9.11) one discovers that in many cases the means of the four respectively five states in the higher order models nearly coincide with one another. This is impressively shown in the case of the first test subject `fp01` in figure 9.3.

For the last part, we can now also check the model pseudo-residuals which can be found as figures 9.6, 9.10 and 9.14. Especially the Q-Q plots are very revealing concerning the fit of the model. There we note that the normal pseudo-residuals of the three-state model fits the standard normal distribution better (`fp01`) or equally good (`fp02`, `fp03`) as the higher-order models.

All this together is proves the three-state model the robustest concept for our input data – and this with a not expected clarity.

RUNTIME The work in this diploma thesis is only done with recorded offline data, but the ulterior motive is of course to advance the technique to make it applicable in online environments. Thus it has to be ensured that the underlying algorithms are capable of a fast, real-time model fitting. Since the algorithms for online applications are designed a little bit differently (for details see [119]), our experiences with the offline algorithms cannot be counted on 100%, but they give a first impression of what is possible.

The Gaussian HMM fitting algorithm of the R-package *RHmm* is definitely fast enough, requiring approximately 50 miliseconds to fit a three-state model on 159 observations. One important fact is also, that the code of the RHmm package is written to great parts in C, making it a lot faster than the self-written version entirely in R. This one takes approximately 2 minutes to fit a three-state Student-t Hidden Markov Model, which is not acceptable in any online setting.

FATIGUE DETECTION QUALITY The main interest lies definitely on the question if our model is able to sufficiently detect the vigilance state of the subjects. Before trying to answer that, we have to recall that the whole problem is ill-posed, i.e. we have no precise indicator

of which level of fatigue the test subject is experiencing at the moment. That is also the reason why the SENSATION project tried to collect as many data as possible in order to detect similarities between them and use that as reliable base sleepiness indicator.

From what the team of VTI and Karolinska Institutet recommend, the KDS scoring (based on the appearance of long blinks) is one the best and most used EOG-based vigilance identifier. In this diploma thesis we focused on the performance of other EOG features, e.g. amplitude, velocity and frequency in order to build our model.

First interim results of models using only the three EOG-based features showed, that the assumption that low values in amplitude and velocity correspond with fatigue does not always seem to hold. See therefor figure 10.1, which shows that the state distributions itself are still reasonable, but the ordering of the states cannot be done sensibly. What gives the car-based features the edge for this match, is that there is no physiology involved in their recording, but only the standardized sensors in the car. This makes them reliable and is also one of the reasons why so many car companies are investing a big part of their research and development budget in car-based fatigue warning systems – they can be kept generalized and so be mass-produced.

When the state ordering is done with the driving features alone, the state sequences are matching the Karolinska Drowsiness Score (KDS) much better. This gives hope, that by including the KDS as a HMM input feature (and discarding the car-based ones), we can achieve to build models depending merely on EOG-data.

GENERALITY The topic of generality in biomedical models has always to be approached very carefully. The main reason for that is the often not negligible difference in physiological parameters between people, even though we all follow the same basic principles. So when speaking of a generally applicable biomedical model, this is meant with an individual training phase for every subject (sometimes even for every new session).

Accepting the fact that the distributions of the HMM have to be determined individually at least for every test subject, there is still hope for the meta-parameters of the model. As the results for three different subjects lead to the same conclusions in the face of number of states, kind of distribution, ergodicity and mixtures we will definitely follow this path. Still one has to keep in mind, that a higher number of observations can open the opportunity to introduce more states, thus refining the steps between them.

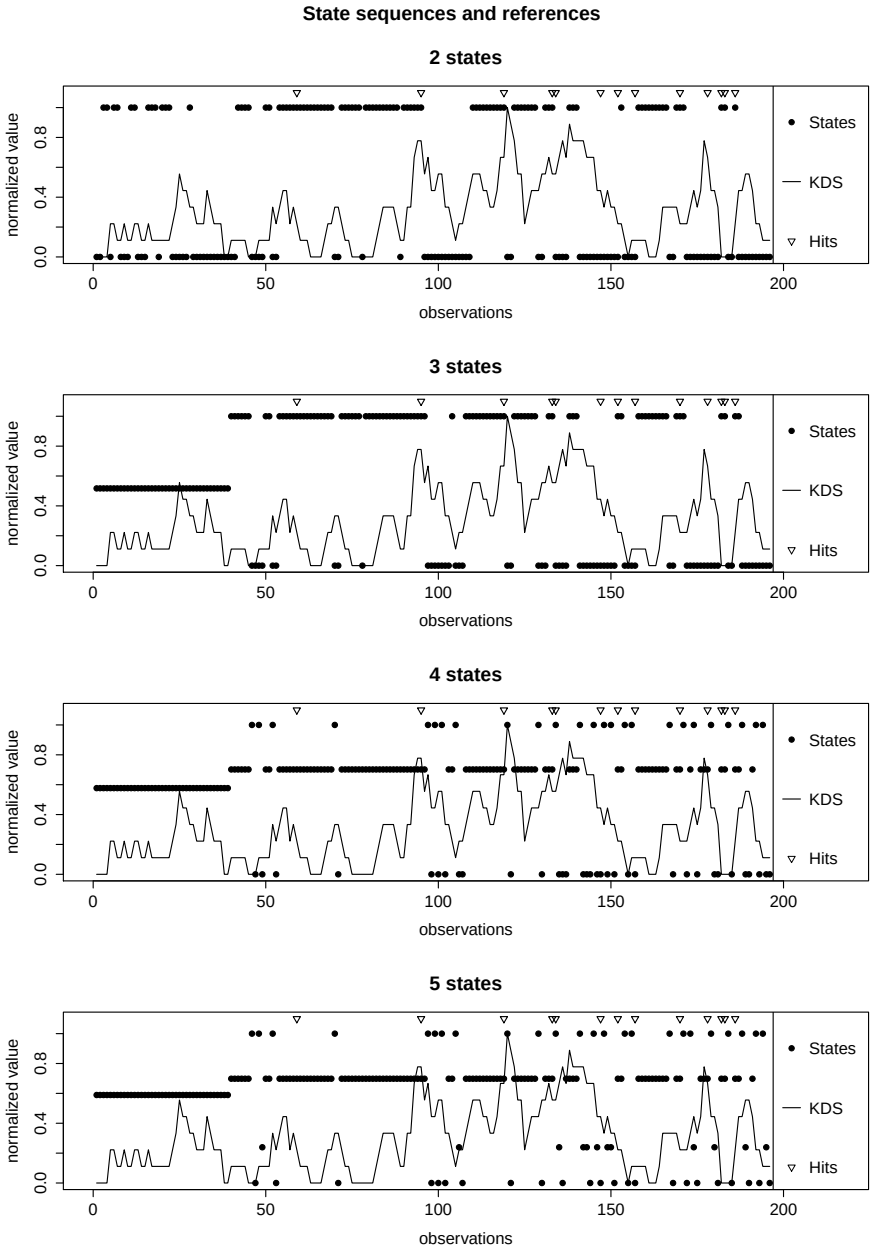


Figure 10.1: (fp03) This graphs show the fit of the model, when only EOG-based features are used for the state ordering process. It can be seen that the characteristic of the so found model states do not necessarily reflect the trend of the KDS values. Including the EOG-based KDS feature in the model seems therefore interesting for the future, since it is more similar to the subject-invariant car-based features.

11 CONCLUSION AND OUTLOOK

11.1 THE FUTURE OF OUR PROJECT

Now the time of truth has come: time to summarize our achievements and drawbacks. We started out with the goal of developing an EOG-based model which should be able to give sensible state estimations, where the states indicate the vigilance level of the test subject. For evaluation purposes we used primarily the – also EOG-based – KDS rate and secondarily the accordance with the rumble strip hits.

In general we can say that we succeeded in finding a suited model with the meta-parameters shown in chapter 9. We get a good and reasonable state estimation for the joint input data (EOG and driving features). Using only the driving features for state ordering, we can observe that the state sequences do finally yield a good representation of the reference data, which are the Karolinska Drowsiness Score (physiological measure) and the rumble strip hits (recorded event data).

These results suggest, that by joining an automatically computed KDS score to the three other EOG-based features, it could be possible to adequately model vigilance states using only EOG-data. When finding a different (independent) reference signal, it would be interesting to rerun the tests with the KDS value made one of the input features for the GHMM. This would pave the way for future applications which rely solely on EOG-data and would be therefore fully mobile and independent.

In order to achieve that, it is also necessary to adapt the Hidden Markov Model algorithms (especially the EM-algorithm) to online specifications, i.e. real-time iterative computation [119]. Not until then we can be sure of the possibility to feasibly detect vigilance states within a realistic setting.

The last step – and by far the most challenging – would be to extend the model to be able to predict future states, relying on the initialization and training data as well as the overall trends. This would give us the possibility to design warnings more precisely in the hope of preventing further drift towards fatigue.

11.2 THE FUTURE OF EOG-BASED DEVICES

The EOG has made its way in the scientific world where its standalone value is shown by many eye-controlled human-machine interfaces (HMI). These are already very capable of determining the single fea-

tures derived from the signal, e.g. blinks, saccades, velocity, frequency, angle, etc., and using them to control computers [120, 121]. With rising accuracy and velocity of the developed devices and programs they become also interesting solutions for handicapped people and therapeutic use.

But the reason why EOG-based devices are not similarly widespread in the non-scientific world is their still mediocre usability and user-friendliness. The best results in terms of signal quality for example are achieved with wet gel electrodes attached to the skin. The procedure of preparing the electrodes for several minutes in advance to every use makes it not favorable for frequent use.

Metal plate dry electrodes, as intended to use in prototypes by our partners, have a significantly reduced signal quality but have the big advantage of (fairly) quick and easy attachment. Still, for purposes of permanent wear, a lot of work has to be put into refining the wearable EOG-devices. They should be lightweight and non-disturbing, because only then a broad acceptance can be reached in the group of people who would profit from such systems, as pilots, drivers and shift workers.

Actual designs of EOG-devices (which have to be positioned somewhere around the eyes) resemble often glasses [53]. One way of raising the acceptance of the wearers would be to join forces with the research in augmented reality, which would turn such eyeglasses into multifunctional devices providing additional benefit to its users.

Apart from the usability issues to overcome, there is another catch in the precision and reliability of the models. Since the underlying motives of the whole research field of vigilance detection and classification are to prevent hazards, such models, in order to get very few false negative responses, have to be biased towards the more tired side. This naturally increases the false positives, i.e. the false alarms of the system, which annoys users and dramatically decreases the acceptance of such a "helping hand", as it is intended to be. The exact thing can be nowadays experienced by listening to the opinion of people, who have bought cars with a (first generation) safety assistance module – they complain about the system proposing to take a rest, even though they do not *feel* tired. But maybe they are?

This project is a small step into the future, a step into a time when technologies as the present one will be fully developed and serve their purpose of helping the mankind preventing severe accidents and make the earth a saver place. And even when we know that this will not be achieved in a couple of years, we should always believe that one day, it will.

Be not afraid of growing slowly, be only afraid of standing still.

— Chinese proverb

Part IV

APPENDIX

A

PROGRAM SOURCE CODE IN R

A.1 GAUSSIAN HMM; USING PACKAGE RHMM

For the case of Gaussian Hidden Markov Models there exist very elaborate packages for R. The optimized code of the RHmm package made it very convenient to work with it, due to its user friendliness and velocity. The following script is built around that package and automatizes the testing of models with different number of states (here 2-5). In addition it calculates various information criteria and the pseudo-residuals (see section 6.4). Finally it displays the results alongside with the reference data, the pseudo-residuals and the comparison of original data and fitted model distributions.

```
##############################################################################
##############################################################################

  # Hidden Markov evaluation of EOG features for vigilance detection #

  # author: Christoph Schneider
  # university: TU Wien
  # date: May, 2012

##############################################################################
##############################################################################

# header

rm(list = ls())           # clear workspace
library("RHmm")           # load Hiddden Markov package 'RHmm'
library("splus2R")        # load function 'vecnorm'


file <- "input_HMM_fp01" # load features for HMM
data <- as.matrix(read.table(file, header=TRUE, sep='\t'))
labels <- c("EOG amplitude","EOG velociy","EOG frequency",
            "Lane variance","Steering angle")

ABA <- NULL               # initialize array for information criteria
sequence <- NULL          # initialize state sequences for HMMs
RG <- NULL                # initialize residuals (for all number of states)

teststat <- c(2,3,4,5)    # quantity of states of interest
restarts <- 50            # number of restarts of RHmm/k-means algorithm

features <- data[,1:5]        # input features for the HMM
KDS <- data[,6]/max(data[,6]) # Karolinska Drowsiness Score (normalized)
Hits <- data[,7]*2 - 1 + 0.1  # rumble strip hits (on top of plot)

numfeats <- dim(features)[2]    # number of input features
end <- length(KDS)              # number of observations

KDS_smoothed <- KDS * 0 # smoothing KDS over 1-min windows (5*20 sec)
for (i in 5:end){KDS_smoothed[i] = sum(KDS[(i-4):i])/5}
KDS_smoothed <- KDS_smoothed / max(KDS_smoothed) # normalize
```

```
##############################################################################
# loop for evaluation of different state quantities

for(h in teststat){

  AIC_global <- NULL      # initialize information criteria
  BIC_global <- NULL
  AICc_global <- NULL
  LLH_global <- NULL

  low <- 0                # auxiliary variable: initialize minimum LLH

  ############################################################################
  # loop for iterating through different starting conditions

  for (o in 1:restarts){

    numstates <-  h                # number of states for HMM

    ##########################################################################
    # Fitting of the Hidden Markov model

    Res <- HMMFit(features,dis='NORMAL',nStates=numstates)
    vit <- viterbi(Res,features)

    # save result when best
    if (is.na(Res$LLH) == FALSE){ # discard NA results
      if (Res$LLH > low){
        Res_min <- Res
        vit_min <- vit
        low     <- Res$LLH
      }
    }

    # number of independent paramerters in the HMM
    nP = (numstates^2 - 1) + numstates * ((numfeats^2+3*numfeats)/2)

    # calculation of information criteria
    AIC_global  <- rbind(AIC_global,Res$AIC)
    AICc_global <- rbind(AICc_global,Res$AIC + 2*nP*(nP+1)/(end - nP -1))
    BIC_global  <- rbind(BIC_global,Res$BIC)
    LLH_global  <- rbind(LLH_global,Res$LLH)

  }

  Res <- Res_min  # shorten name for future use
  vit <- vit_min
  print(Res)      # display best result
  print(vit)

  # save statistics of information criteria
  ABA <- cbind(ABA,c(mean(AIC_global,na.rm=TRUE),
                     mean(AICc_global,na.rm=TRUE),
                     mean(BIC_global,na.rm=TRUE),
                     mean(LLH_global,na.rm=TRUE),
                     max(AIC_global,na.rm=TRUE) - mean(AIC_global,na.rm=TRUE),
                     mean(AIC_global,na.rm=TRUE) - min(AIC_global,na.rm=TRUE)))

  ############################################################################
  # saving the best result

  # reorder HMM-states in the most sensible way
  norm = array(0,c(1,numstates))

  for (i in 1:numstates){
```

```
  x <- unlist(Res$HMM$distribution$mean[i])

  # EOG features normally decay with rising fatigue, the driving features rise
  # => building a measure to order the states

  norm[i] <- vecnorm(x[4:5],2)
}
nnorm <- (norm-min(norm))/(max(norm)-min(norm)) # span over [0,1]
sequence[[h]] <- nnorm[vit$states]

#############################################################################
# calculate ordinary pseudo-residuals

residuals <- NULL

x <- density(features[,1])$x    # span of plot of distributions
y <- list()
length(y) <- numfeats  # list for values of every feature and state

for (i in 1:numstates){

  measurements <- NULL

  measurements <- features[vit$states == i,]  # get according measurements ...
  mean <- Res$HMM$distribution$mean[[i]]      # ... means and ...
  covs <- Res$HMM$distribution$cov[[i]]       # ... covariances
  r <- NULL

  for (j in 1:numfeats){

    for (c in 1:length(measurements[,j])){

                        # density estimation without measurement[c,j]
      est_dens <- density(measurements[-c,j])

      e <- est_dens$x   # points on x-axis of estimated distribution

                        # find nearest point to measurement[c,j] on e
      pos <- which.min(abs(e - measurements[c,j]))

                        # stepwidth between two points on e
      stepwidth <- (range(e)[2] - range(e)[1])/which.max(e)

                        # calculate and write pseudo-residual to r
      new_res <- sum(est_dens$y[1:pos])*stepwidth
      r <- rbind(r,new_res)
    }

    m <- mean[j]
    s <- sqrt(covs[j,j])
    y[[j]] <- rbind(y[[j]],dnorm(x,m,s)/numstates)
  }

  residuals <- append(residuals,r)

}

RG[[h]] <- residuals

#############################################################################
# plot data and model distributions

# set up new plot environment/ 2 pages
if (h==2){par(mfcol=c(numfeats,length(teststat)/2),oma=c(0,0,2,0))}
if (h==4){par(mfcol=c(numfeats,length(teststat)/2),oma=c(0,0,2,0))}
```

```
  for (i in 1:numfeats){    #  plot densities of model (black) and data (gray)
    model_dist <- apply(y[[i]],2,sum)
    mf <- mean(features[,i])
    sf <- sd(features[,i])
    feat_dist <- density(features[,i])
    plot(NULL,NULL,xlim=c(mf-2*sf,mf+2*sf),ylim=c(0,max(model_dist,feat_dist$y)),
         xlab="",ylab="Density")
    title(labels[i])
    lines(x,model_dist,lwd=2,lty=1)
    lines(feat_dist,col="darkgray",lwd=2)

    for (j in 1:numstates){       # plot single state densities (dashed lines)
      lines(x,y[[i]][j,],lty=2)
    }
  }

  # writes titles to the 2 pages
  if (h==2){
    title("Comparison of data and model distributions for 2 and 3 states",
          outer=TRUE)
  }

  if (h==4){
    title("Comparison of data and model distributions for 4 and 5 states",
          outer=TRUE)
  }

}

#############################################################################
# plot best models for each number of states h

par(mfrow=c(length(teststat),1),oma=c(0,0,2,0)) # number of plots on page

for (j in teststat){                        # plotting best results
  plot(Hits,pch=25,main=paste(j,"states"),  # rumble strip hits (triangles)
       xlim=c(0,end+20),ylim = c(0,max(Hits)),
       xlab="observations",ylab="normalized value")
  lines(KDS_smoothed,pch=4)                 # KDS (continuous line)
  points(sequence[[j]],pch=19)              # state sequence (solid dots)
  abline(v=end+1)                           # border to legend
  legend(c(end+3,end+20),y=c(0,max(Hits))+0.2,
         bty="n",legend=c("States","KDS","Hits"),
         pch=c(19,NA,25),lty = c(0,1,0), lwd = c(0,1,0))
}
title(main="State sequences and references",outer=TRUE)

#############################################################################
# plot ordinary pseudo-residuals

par(mfcol=c(4,4),oma=c(0,0,2,0))   # four plots on one page

for (j in teststat){

  # 1st plot: histogram of uniform pseudo-residuals
  hist(RG[[j]],freq=FALSE,col="gray",ylim=c(0,2),xlab="Residual value"
       ,ylab="Density",main="Uniform pseudo-residuals")
  abline(h=1,lty=2)

  # 2nd plot: histogram of normal pseudo-residuals
  hist(qnorm(RG[[j]]),freq=FALSE,col="gray",xlim=c(-4,4),ylim=c(0,0.6),
       main="Normal pseudo-residuals",xlab="Residual value")
  z=seq(-4,4,length=500)
  lines(z,dnorm(z),lty=2)

  # 3rd plot: quantile-quantile plot of normal pseudo-residuals
```

```
  qqnorm(qnorm(RG[[j]]), main="Q-Q plot (normal)",
         xlim=c(-4,4),ylim=c(-4,4))
  abline(a=0, b=1, lty=2)

  # 4th plot: autocorrelation function of the normal pseudo-residuals
  acf(qnorm(RG[[j]]),lwd=3,lag.max=10,ci.col="black",
      main="ACF (normal)")

}

title("Statistics of pseudo-residuals for 2,3,4 and 5 states",outer=TRUE)


##############################################################################
# save and display the information criteria

par(mfrow=c(3,1),oma=c(0,0,2,0))     # three plots on the page

colnames(ABA) <- paste(teststat,"states")
rownames(ABA) <- c("AIC","Aicc","BIC","LLH","top whisker","bottom whisker")
write.table(round(ABA,2),file=paste("ABA_",file,".txt",sep = ""),sep="\t")

tit = c("AIC","AICc","BIC") # titles for plots

for (i in 1:3){

  zu <- ABA[i,]+ABA[5,]   # upper boundary
  zm <- ABA[i,]           # mean value
  zl <- ABA[i,]-ABA[6,]   # lower boundary

  yrange <- c(min(zl),max(zu))

  # plot values with customized x-axis = number of states used
  plot(zm,type="b",ylim=yrange,xaxt="n"
       ,xlab="number of states",ylab="")
  axis(1, at=1:length(teststat), labels=as.character(teststat))
  title(tit[i])


  for (j in 1:length(teststat)){        # add minima and maxima
    arrows(j,zm[j],j,zu[j],length=0.05,angle=90)
    arrows(j,zm[j],j,zl[j],length=0.05,angle=90)
    points(j,zm[j],col="white",pch=20)  # erase lines through circles
  }

}

title(main="Information Criteria",outer=TRUE)
```

A.2 STUDENT-T HMM

To that point of time there exists no available R-package which is capable of dealing with Student-t mixture distributions. The following code is my humble implementation of the concepts of continuous Student-t HMMs, based on the work of Rabiner [83] and Chatzis et al. [106]. The notation borrowed from that two sources and should be able to be understood when familiar with them.

Although fully operational (and hopefully bug-free) the code is written solely in R and is not runtime-optimized. Due to that it needs approximately 10^3-times longer than the RHmm package.

The starting values in the `Run.R` are the optimized results of the Gaussian HMM for subject `fp01`. They should lie close to the optimal values for the Student-t distributions.

RUN.R

```
###############################################################################
###############################################################################

                # Student-t Hidden Markov Models #


       # author: Christoph Schneider
       # university: TU Wien
       # date: May, 2012

###############################################################################
###############################################################################

rm(list = ls()) # clear workspace

# load libraries
library("corpcor")    # for pseudoinverse

# including external functions
source("Forward.r")
source("Viterbi.r")
source("Backward.r")
source("EM.r")
source("Mixprob.r")
source("TDistribution.r")

###############################################################################
# Initialization:

file <- "input_HMM_fp01" # load features for HMM
data <- as.matrix(read.table(file, header=TRUE, sep='\t'))
labels <- c("EOG amplitude","EOG velociy","EOG frequency",
            "Lane variance","Steering angle")

observation <- data[,1:5]
print(list('observation sequence: ',observation))

# initializing the transition probability matrix (TPM)

x= c(8.126161e-01, 2.324226e-11, 0.18738395,
     2.917443e-18, 9.398943e-01, 0.06010571,
     9.606052e-02, 4.410835e-02, 0.85983113)
dim(x) = c(3,3)
Transprob <- t(x)
```

```
print(list('Transition Probability Matrix: ',Transprob))

# initializing the observation probabilities (OP)
B1 <- c(1)
B2 <- c(1)
B3 <- c(1)
Weights <- matrix(rbind(B1,B2,B3),3)            # mixture weights


x1 <- c(0.5193372,0.4754052,0.5723557,0.5133089,0.3451783)
x2 <- c(0.23680909,0.28388960,0.34649317,0.21863538,0.08212454)
x3 <- c(0.4617714,0.4497982,0.5144821,0.2606850,0.1378977)
Means <- list(x1,x2,x3)
dim(Means) <- c(1,3)
Means <- t(Means)                               # means


y1 <- c( 0.0275085318,  0.022048924,  0.007546027,   0.003517284,  -0.0009371224,
         0.0220489240,  0.022699353,  0.010512684,   0.003226340,  -0.0033244139,
         0.0075460268,  0.010512684,  0.014698090,  -0.004336467,   0.0036101192,
         0.0035172844,  0.003226340, -0.004336467,   0.028190172,   0.0141630163,
        -0.0009371224, -0.003324414,  0.003610119,   0.014163016,   0.0544553587)

y2 <- c( 0.0119700491,  0.0146749639, 0.0070177553, -0.0013867273, -0.0006730231,
         0.0146749639,  0.0212469950, 0.0078343473, -0.0045388849, -0.0009609869,
         0.0070177553,  0.0078343473, 0.0111916891,  0.0001440021,  0.0003975450,
        -0.0013867273, -0.0045388849, 0.0001440021,  0.0104023052,  0.0007251109,
        -0.0006730231, -0.0009609869, 0.0003975450,  0.0007251109,  0.0009209680)

y3 <- c( 0.0167159252,  1.423341e-02, 9.763214e-04,  2.868413e-03, -9.903016e-04,
         0.0142334142,  1.819647e-02, 5.651296e-05,  2.100225e-03, -1.071399e-03,
         0.0009763214,  5.651296e-05, 1.030234e-02,  9.241889e-04,  3.797069e-04,
         0.0028684135,  2.100225e-03, 9.241889e-04,  1.180866e-02, -1.959229e-05,
        -0.0009903016, -1.071399e-03, 3.797069e-04, -1.959229e-05,  3.295581e-03)


dim(y1) <- c(5,5)
dim(y2) <- c(5,5)
dim(y3) <- c(5,5)
Covs <- list(y1,y2,y3)
dim(Covs) <- c(1,3)
Covs <- t(Covs)                                 # covariances

B1 <- c(2)
B2 <- c(2)
B3 <- c(2)
Dofs <- matrix(rbind(B1,B2,B3),3)               # degrees of freedom


Paramdt <- list(A=Weights,B=Means,C=Covs,D=Dofs) # build parameter set

print(list('Distribution Probability Weights: ',Weights))
print(list('Student Distribution Means: ',Means))
print(list('Student Distribution Covariance Matrices: ',Covs))
print(list('Student Distribution Degrees of Freedom: ',Dofs))

# initializing the starting probability (SP)

Inprob <- c(0.3,0.3,0.4)

print(list('Initial Observation Probability: ',Inprob))

# initialize auxiliary variables
Sequence <- vector("list", 0) # initialize list of Viterbi-sequences
LogProb <- vector("list", 0)  # initialize list of probabilities
```

```
Difference <- 1.0e+10 # set initial Difference to a implausible high value
threshold <- 1.0e-10  # threshold for overall Difference of TPM, OPM and SP ...
                      # ... in two subsequent estimation-maximization-steps
count <- 0            # count variable


################################################################################
# Iteration:

t1 = Sys.time()     # set starting time

print("iteration")

while (threshold <= Difference) { # loop breaks if difference of parameters ...
                                  # in subsequent steps smaller than threshold

  count <- count+1
  print(count)

  Mix <- Mixprob(observation,Paramdt)

  # Forward-algorithm:
  z <- Forward(Inprob,Paramdt,Transprob,observation)
  Alpha <- z$B
  LogProb <- append(LogProb,list(z$A))

  # Viterbi-alorithm:
  Sequence <- append(Sequence,list(Viterbi(Inprob,Paramdt,
                                           Transprob,observation)))

  # Backward-algorithm:
  Beta <- Backward(Inprob,Paramdt,Transprob,observation,z$C)

  # EM-algorithm:
  New_Param <- EM(Alpha,Beta,observation,Transprob,Paramdt,
                  dim(observation)[2],Mix)

  Inprob    <- New_Param$A
  Transprob <- New_Param$B
  Paramdt   <- New_Param$C

  # set absolute difference between the last two steps
  if (count>1) {Difference <- (abs(LogProb[[count]]-LogProb[[count-1]]))}

}

t2 = Sys.time()     # set ending time

################################################################################
# Display results:

print(list('count of iterations:',count))
print(list('optimal initial probability: ',Inprob))
print(list('optimal state transition probability: ',Transprob))
print(list('optimal parameters: ',Paramdt))
print(list('optimal state sequence: ',Sequence[length(Sequence)]))
print(list('best possible log probability:',round(LogProb[[length(LogProb)]],2)))
print(round(t2-t1,2))
```

TDISTRIBUTION.R

```
# density evaluation of student-t distribution

TDistribution <- function(x,my,sigma,dof,p){

  gamma((dof+p)/2)*(det(sigma))^(-1/2)*(pi*dof)^(-p/2)*
    gamma(dof/2)^(-1)*(1+((x-my) %*% pseudoinverse(sigma) %*%
    (x-my))/dof)^(-(dof+p)/2)

}
```

MIXPROB.R

```
# calculate mixture probabilities of observations

Mixprob <- function (x,Paramdt){

  e <- dim(Paramdt$A)[1]            # auxiliary variables
  f <- dim(Paramdt$A)[2]
  g <- dim(x)

  probmat <- array(0,c(e,f,g[1]))   # initialize probability array

  for (k in 1:g[1]){

    for (j in 1:e){

      for (i in 1:f){

      probmat[j,i,k] <- Paramdt$A[j,i]*
      TDistribution(x[k,],Paramdt$B[[j,i]],Paramdt$C[[j,i]],Paramdt$D[j,i],g[2])

      }

    }

  }

  return(list(A=probmat,B=apply(probmat,c(3,1), sum)))
}
```

FORWARD.R

```
# Forward-algorithm

Forward <- function (Inprob,Paramdt,Transprob,observation){

  d <- dim(observation)

  ##########################################################################
  # Initialization

  Nextprob <- Mixprob(observation,Paramdt)$B  # probabilites of observations
  Alpha <- mat.or.vec(d[1],length(Inprob))
  Alpha[1,] <- Inprob * Nextprob[1,]          # initialization of alpha

  fact <- mat.or.vec(d[1],1)                  # scaling factor
  fact[1] <- 1

  ##########################################################################
```

```
  # Induction

  for (i in 1:(d[1]-1)){

    Alpha[i+1,] <- (Alpha[i,] %*% Transprob)*Nextprob[i+1,]
    fact[i+1] <- 1/sum(Alpha[i+1,])
    Alpha[i+1,] <- Alpha[i+1,]*fact[i+1]      # scale alpha

  }

  ###########################################################################
  # Termination & return values

  LogProb <- -sum(log(fact))                  # logarithmic probability

  return(list(A=LogProb,B=Alpha,C=fact))
}
```

BACKWARD.R

```
# Backward-algorithm

Backward <- function (Inprob,Paramdt,Transprob,observation,fact){

  ###########################################################################
  # Initialization

  Nextprob <- Mixprob(observation,Paramdt)$B

  Beta <- mat.or.vec(dim(observation)[1],dim(Transprob)[1])
  Beta[dim(observation)[1],] <- fact[dim(observation)[1]]

  ###########################################################################
  # Iteration

  for (i in (dim(observation)[1]-1):1){

    Beta[i,] <- Transprob %*% (Nextprob[i+1,]*Beta[i+1,])
    Beta[i,] <- Beta[i,] * fact[i]

  }

  ###########################################################################
  # Return values

  return(Beta)
}
```

EM.R

```
# Expectation-Maximization-algorithm

EM <- function(alpha,beta,observation,Transprob,Paramdt,p,Mix){

  # Variable shortening
  Weights <- Paramdt$A
  Means <- Paramdt$B
  Covs <- Paramdt$C
  Dofs <- Paramdt$D

  l <- dim(observation)[1]
```

```
d <- dim(Weights)

###############################################################################
# Initialization

u_t  <- array(0,c(d[1],d[2],l))
xi_t <- array(0,c(d[1],d[2],l))
r_t  <- array(0,c(d[1],d[2],l))
gamma_ht <- array(0,c(d[1],d[1],l-1))

###############################################################################
# Estimation step


gamma_t <- ((alpha*beta)/rowSums(alpha*beta))

for (i in 1:dim(Mix$A)[1]){

  r_t[i,,] <-  t(Mix$A[i,,]) * (gamma_t / Mix$B)[,i]

}

for (i in 1:l){

  Mahalanobisdistance <- Weights*0  # initialize Mahalanobian distance matrix

  for (j in 1:prod(d)){

    m <- ((j-1) %% d[1]) + 1             # indices for all matrix elements
    n <- ((j-1) %/% d[1]) + 1            # indices for all matrix elements

    Mahalanobisdistance[m,n] <- sqrt( (observation[i,] - Means[[m,n]])
                                      %*% pseudoinverse(Covs[[m,n]])
                                      %*% (observation[i,]-Means[[m,n]]) )
  }

  u_t[,,i]  <- (Dofs + p)/(Dofs + Mahalanobisdistance)
  dev <- Mix$A[,,i]

  if (is.array(Mix$A[,,i]) == T){
    dev <- rowSums(Mix$A[,,i])
  }

  xi_t[,,i] <- Mix$A[,,i] / dev

  if (i < l){
    z <- alpha[i,]*t(t(Transprob) * (Mix$B[i+1,]*beta[i+1,]))
    gamma_ht[,,i] <- z / sum(z)
  }

}

###############################################################################
# Maximization step

Inprob_new <- gamma_t[1,]

# new transition probability matrix
A_new <- apply(gamma_ht,c(1,2),sum)/colSums(gamma_t[2:l-1,])

# new weights for mixture distributions
Weights_new <- apply(r_t,c(1,2),sum)/colSums(gamma_t)

# new means
Means_new <- as.list(Weights*0)
dim(Means_new) <- dim(Weights)
```

```
div <- apply(u_t * r_t,c(1,2),sum)
up <- r_t * u_t

for (k in 1:l){
  for(i in 1:d[1]){
    for (j in 1:d[2]){

      vec <- observation[k,] * up[i,j,k]
      Means_new[[i,j]] <- Means_new[[i,j]] + vec/div[i,j]

    }
  }
}

# new covariance matrices
Covs_new <- as.list(Weights*0)
dim(Covs_new) <- dim(Weights)
div <- apply(r_t,c(1,2),sum)

for (k in 1:l){
  for(i in 1:d[1]){
    for (j in 1:d[2]){

      vec <- observation[k,] - Means_new[[i,j]]
      mat <- (vec %*% t(vec)) * up[i,j,k]
      Covs_new[[i,j]] <- Covs_new[[i,j]] + mat/div[i,j]

    }
  }
}

# covariance regularization (shrinkage regularization)

func <- function(S,l,p){

  top <-  (1-2/p)*sum(diag(S%*%S)) + (sum(diag(S)))^2
  bottom <- (l+1+2/p)*(sum(diag(S%*%S)) + (sum(diag(S)))^2/p)
  return(top/bottom)

}

for (i in 1:(d[1]*d[2])){

  x <- (i-1) %% d[1] + 1         # indices for all matrix elements
  y <- i %/% (d[1]+1)+ 1         # indices for all matrix elements
  f <-  (sum(diag(Covs_new[[x,y]]))/l)*diag(p)
  rho <- min(1, func(S = Covs_new[[x,y]], l = l, p = p))
  Covs_new[[x,y]] <-  (1-rho)*Covs_new[[x,y]] + rho*f

}


# new degrees of freedom for Student-t distributions

Dofs_new <- Dofs*0

fn <- function(n,nk,r,u,p){

  out <-  abs(1 - digamma(n/2) + log(n/2) + digamma((nk+1)/2)
              - log((nk+1)/2) + (1/sum(r))*sum(r*(log(u)-u)))
  return(out)

}

for (i in 1:(d[1]*d[2])){
```

```
    x <- (i-1) %% d[1] + 1         # indices for all matrix elements
    y <- i %/% (d[1]+1)+ 1         # indices for all matrix elements
    Dofs_new[x,y] <- optimize(fn, c(1, 100), tol = 0.0001, nk = Dofs[x,y],
                              r = r_t[x,y,], u = u_t[x,y,], p = p )$minimum

  }

  Dofs_new <- ceiling(Dofs_new)    # careful estimation (higher values)


  ###############################################################################
  # Return Values

  Param_new <- list(A=Weights_new,B=Means_new,C=Covs_new,D=Dofs_new)

  return(list(A=Inprob_new,B=A_new,C=Param_new))
}
```

VITERBI.R

```
# Viterbi-algorithm

Viterbi <- function(Inprob,Paramdt,Transprob,observation){

  ###############################################################################
  # Initialization

  Nextprob <- Mixprob(observation,Paramdt)$B # probabilites of observations

  delta <- t(as.matrix(log(Inprob) + log(Nextprob[1,])))
  psi <- Inprob*0
  logTransprob <- log(Transprob)

  ###############################################################################
  # Iteration

  for (i in 2:dim(observation)[1]){

    delta <- rbind (delta, apply(delta[i-1,] + logTransprob, 2, max)
                    + log(Nextprob[i,]))
    psi <- rbind(psi,apply(delta[i-1,] + logTransprob, 2, which.max))

  }

  ###############################################################################
  # Termination

  logpstar <- max(delta[dim(delta)[1],])
  qstarT <- which.max(delta[dim(delta)[1],])

  ###############################################################################
  # Path backtracking

  qstar <- rep(0,dim(psi)[1])

  qstar[length(qstar)] <- qstarT

  for (i in (length(qstar)-1):1){qstar[i] <- psi[i+1,qstar[i+1]]}

  ###############################################################################
  # Return values

  return (qstar)
}
```

BIBLIOGRAPHY

[1] M. Moore-Ede, *The twenty-four-hour society: understanding human limits in a world that never stops*. Addison-Welsey, 1993.

[2] "Annual Safety Review 2010," tech. rep., EASA - European Aviation Safety Agency, 2011.

[3] "2011 Aviation Safety Performance," tech. rep., IATA - International Air Transport Association, 2012.

[4] "Annual Statistical Report 2010," tech. rep., European Road Safety Observatory, 2010.

[5] "Road Safety in Austria; Annual Report 2010," tech. rep., Federal Ministry for Transport, Innovation and Technology, 2011.

[6] T. Little, W. Hann, W. Bramble, N. Marshall, and M. Garber, "Factual Report – Aviation, NTSB ID: SEA08IA080," tech. rep., National Transportation Safety Board, 2008.

[7] J. A. Horne and L. A. Reyner, "Sleep related vehicle accidents," *British Medical Journal*, vol. 310, pp. 565–567, 1995.

[8] "Fatigue, Alcohol, Other Drugs, and Medical Factors in Fatal-to-the-Driver Heavy Truck Crashes. Safety Study NTSB/SS-90/01 and NTSB/SS-90/02," tech. rep., National Transportation Safety Board, 1990.

[9] J. H. Goode, "Are pilots at risk of accidents due to fatigue?," *Journal of Safety Research*, vol. 34, no. 3, pp. 309–313, 2003.

[10] J. A. Caldwell, "Fatigue in aviation," *Travel Medicine and Infectious Disease*, vol. 3, no. 2, pp. 85–96, 2005.

[11] D. M. Gaba and S. K. Howard, "Fatigue among Clinicians and the Safety of Patients," *New England Journal of Medicine*, vol. 347, no. 16, pp. 1249–1255, 2002.

[12] A. E. Rogers, *Patient Safety and Quality: An Evidence-Based Handbook for Nurses*, ch. 40: The Effects of Fatigue and Sleepiness on Nurse Performance and Patient Safety. Rockville (MD): Agency for Healthcare Research and Quality (US), 2008.

[13] G. S. Larue, A. Rakotonirainy, and A. N. Pettitt, "A model to predict hypovigilance during a monotonous task," in *2009 Australasian Road Safety Research, Policing and Education Conference : Smarter, Safer Directions*, 2009.

[14] J. Stutts, J. Wilkins, and B. Vaughn, *Why Do People Have Drowsy Driving Crashes? Input from Drivers who Just Did*. Diane Publishing Company, 1999.

[15] J. J. Pilcher and A. I. Huffcutt, "Effects of sleep deprivation on performance: a meta-analysis," *Sleep Research & Sleep Medicine*, vol. 19, no. 4, pp. 318–326, 1996.

[16] A. B. Dollins, I. V. Zhdanova, R. J. Wurtman, H. J. Lynch, M. H. Deng, and R. J. Wurtman, "Effect of inducing nocturnal serum melatonin concentrations in daytime on sleep, mood, body temperature, and performance," *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, vol. 91, no. 5, pp. 1824–1828, 1994.

[17] P. van den Hurk, "Validation of a Hypovigilance-Mangagement-System for Industrial Workers," Master's thesis, Eindhoven University of Technology, 2007.

[18] L. K. Barger, B. E. Cade, N. T. Ayas, J. W. Cronin, B. Rosner, F. E. Speizer, and C. A. Czeisler, "Extended Work Shifts and the Risk of Motor Vehicle Crashes among Interns," *New England Journal of Medicine*, vol. 352, no. 2, pp. 125–134, 2005.

[19] P. Thiffault and J. Bergeron, "Monotony of road environment and driver fatigue: a simulator study," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 381–391, 2003.

[20] P. Tucker, L. Smith, I. Macdonald, and S. Folkard, "Shift length as a determinant of retrospective on-shift alertness," *Scandinavian Journal of Work, Environment & Health*, vol. 24, pp. 49–54, 1998.

[21] D. F. Neri, R. L. Oyung, L. M. Colletti, M. M. Mallis, P. Y. Tam, D. F. Dinges, and D. F. Dinges, "Controlled breaks as a fatigue countermeasure on the flight deck," *Aviation, Space and Environmental Medicine*, vol. 73, no. 7, pp. 654–664, 2002.

[22] P. Tucker, "The impact of rest breaks upon accident risk, fatigue and performance: A review," *Work Stress*, vol. 17, no. 2, pp. 123–137, 2003.

[23] M. Hayashi, Y. Chikazawa, and T. Hori, "Short nap versus short rest: recuperative effects during VDT work," *Ergonomics*, vol. 47, no. 14, pp. 1549–1560, 2004.

[24] T. Åkerstedt, A. Knutsson, P. Westerholm, T. Theorell, L. Alfredsson, and G. Kecklund, "Mental fatigue, work and sleep," *Journal of Psychosomatic Research*, vol. 57, no. 5, pp. 427–433, 2004.

[25] J. A. Horne and L. A. Reyner, "Counteracting driver sleepiness: Effects of napping, caffeine, and placebo," *Psychophysiology*, vol. 33, no. 3, pp. 306–309, 1996.

[26] M. M. Lorist and M. Tops, "Caffeine, fatigue, and cognition," *Brain and Cognition*, vol. 53, no. 1, pp. 82–94, 2003.

[27] L. Galvani, "De viribus electricitatis in motu musculari. Commentarius.," *De Bononiesi Scientarium et Ertium Instituto atque Academia*, vol. Commentarii 7, pp. 363–418, 1791.

[28] A. Volta, "On the electricity excited by the mere contact of conducting substances of different kinds.," *Philosophical Transactions of the Royal Society, London*, vol. 90, pp. 403–431, 1800.

[29] J. C. Maxwell, "A Dynamical Theory of the Electromagnetic Field," *Philosophical Transactions of the Royal Society, London*, vol. 155, pp. 459–512, 1865.

[30] H. Snellen, *Willem Einthoven (1860-1927): father of electrocardiography : life and work, ancestors and contemporaries*. Kluwer Academic Publishers, 1995.

[31] J. Malmivuo and R. Plonsey, *Bioelectromagnetism : principles and applications of bioelectric and biomagnetic fields*. New York : Oxford University Press, 1995.

[32] S. Venkataramanan, P. Prabhat, S. Choudhury, H. Nemade, and J. Sahambi, "Biomedical instrumentation based on electrooculogram (EOG) signal processing and application to a hospital alarm system," in *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on*, pp. 535–540, 2005.

[33] J. Bronzino, *The biomedical engineering handbook*. The electrical engineering handbook series, CRC Press, 2000.

[34] A. Kales and A. Rechtschaffen, *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. U. S. National Institute of Neurological Diseases and Blindness, Neurological Information Network, Bethesda, Md., 1968.

[35] V. Häkkinen, K. Hirvonen, J. Hasan, M. Kataja, A. Värri, P. Loula, and H. Eskola, "The effect of small differences in electrode position on EOG signals: application to vigilance studies," *Electroencephalography and Clinical Neurophysiology*, vol. 86, no. 4, pp. 294–300, 1993.

[36] A. B. Usakli, S. Gurkan, F. Aloise, G. Vecchiato, and F. Babiloni, "On the use of electrooculogram for efficient human computer interfaces," *Computational Intelligence and Neuroscience*, vol. 2010, 2010.

[37] N. Nöjd and J. Hyttinen, "Modeling of EOG and Electrode Position Optimization for Human-Computer Interface," in *3rd*

International ICST Conference on Body Area Networks, Association for Computing Machinery, 2010.

[38] C. Braun, M. Gründl, C. Marberger, and C. Scherber, "Beautycheck - Ursachen und Folgen von Attraktivität. Projektabschlussbericht," 2001.

[39] M. Merino, O. Rivera, I. Gomez, A. Molina, and E. Dorronzoro, "A Method of EOG Signal Processing to Detect the Direction of Eye Movements," pp. 100–105, 2010.

[40] S. R. Choudhury, S. Venkataramanan, H. B. Nemade, and J. S. Sahambi, "Design and Development of a Novel EOG Biopotential Amplifier," *International Journal of Bioelectromagnetism*, vol. 7, no. 1, pp. 271–274, 2005.

[41] M. Marmor, M. Brigell, D. McCulloch, C. Westall, and M. Bach, "ISCEV standard for clinical electro-oculography (2010 update)," *Documenta Ophthalmologica*, vol. 122, pp. 1–7, 2011.

[42] A. Bulling, J. Ward, H. Gellersen, and G. Tröster, "Eye Movement Analysis for Activity Recognition Using Electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2011.

[43] T. Yagi, "Eye-gaze interfaces using electro-oculography (EOG)," in *Proceedings of the 2010 workshop on Eye gaze in intelligent human machine interaction*, pp. 28–32, 2010.

[44] K. Shinomiya, N. Itsuki, M. Kubo, and H. Shiota, "Analyses of the characteristics of potential and cross-talk at each electrode in electro-oculogram," *The Journal of Medical Investigation*, vol. 55, no. 1,2, pp. 120–126, 2008.

[45] D. Dinsdale, J. Chadwick, "Communication aid for disabled patients makes use of facial EMG signals and neural nets," *Mechatronic Aids for the Disabled, IEE Colloquium on*, vol. 6, p. 1, 1995.

[46] W. Li and K. Sakamoto, "The influence of location of electrode on muscle fiber conduction velocity and EMG power spectrum during voluntary isometric contraction measured with surface array electrodes," *Applied Human Science*, vol. 15, pp. 25–32, 1996.

[47] A. R. Bentivoglio, S. B. Bressman, E. Cassetta, D. Carretta, P. Tonali, and A. Albanese, "Analysis of blink rate patterns in normal subjects," *Movement Disorders*, vol. 12, pp. 1028–1034, 1997.

[48] M. S. Reddy, A. Sammaiah, B. Narsimha, and K. S. Rao, "Analysis of EOG Signals Using Empirical Mode Decomposition for Eye Blink Detection," in *Multimedia and Signal Processing (CMSP), 2011 International Conference on*, vol. 2, pp. 293–297, 2011.

[49] M. Bruyneel, C. Sanida, G. Art, W. Libert, L. Cuvelier, M. Paesmans, R. Sergysels, and V. Ninane, "Sleep efficiency during sleep studies: results of a prospective study comparing home-based and in-hospital polysomnography," *Journal of Sleep Research*, vol. 20, pp. 201–206, 2011.

[50] N. B. Melek, S. Blanco, and H. Garcia, "Electro-oculography of smooth pursuit and optokinetic nystagmus eye movements in type I Duane's retraction syndrome," *Binocul Vis Strabismus Q*, vol. 21, pp. 37–44, 2006.

[51] I. Ingster-Moati, E. Bui Quoc, M. Pless, R. Djomby, C. Orssaud, J. P. Guichard, and F. Woimant, "Ocular motility and Wilson's disease: a study on 34 patients," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 78, pp. 1199–1201, 2007.

[52] M. Kirbis and I. Kramberger, "Multi Channel EOG Signal Recognition for an Embedded Eye Movement Tracking Device," in *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*, pp. 1–4, 2009.

[53] A. Bulling, D. Roggen, and G. Tröster, "Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments," *Journal of Ambient Intelligence and Smart Environment*, vol. 1, no. 2, pp. 157–171, 2009.

[54] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "EMG and EOG artifacts in brain computer interface systems: A survey," *Clinical Neurophysiology*, vol. 118, pp. 480–494, 2007.

[55] B. Jammes, H. Sharabty, and D. Esteve, "Automatic EOG analysis: A first step toward automatic drowsiness scoring during wake-sleep transitions," *Somnologie - Schlafforschung und Schlafmedizin*, vol. 12, pp. 227–232, 2008.

[56] J. A. Stern, D. Boyer, and D. Schroeder, "Blink rate: a possible measure of fatigue," *Human Factors*, vol. 36, pp. 285–297, 1994.

[57] Y. Wang, S. S. Toor, R. Gautam, and D. B. Henson, "Blink frequency and duration during perimetry and their relationship to test-retest threshold variability," *Investigative Ophthalmology & Visual Science*, vol. 52, pp. 4546–4550, 2011.

[58] T. Åkerstedt, M. Ingre, G. Kecklund, A. Anund, D. Sandberg, M. Wahde, P. Philip, and P. Kronberg, "Reaction of sleepiness indicators to partial sleep deprivation, time of day and time on task in a driving simulator – the DROWSI project," *Journal of Sleep Research*, vol. 19, no. 2, pp. 298–309, 2010.

[59] L. De Gennaro, M. Ferrara, F. Ferlazzo, and M. Bertini, "Slow eye movements and EEG power spectra during wake-sleep transition," *Clinical Neurophysiology*, vol. 111, pp. 2107–2115, 2000.

[60] J. X. Ma, L. C. Shi, and B. L. Lu, "Vigilance estimation by using electrooculographic features," *Conference proceedings: IEEE Engineering in Medicine and Biology Society*, vol. 2010, pp. 6591–6594, 2010.

[61] S. Hanke, J. Zeitlhofer, G. Wiest, W. Mayr, and D. C. Moser, "Automated Vigilance Classification based on EOG signals: Preliminary Results," in *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, vol. 25 of *IFMBE Proceedings*, pp. 428–431, 2009.

[62] J. Minkwitz, M. U. Trenner, C. Sander, S. Olbrich, A. J. Sheldrick, P. Schonknecht, U. Hegerl, and H. Himmerich, "Prestimulus vigilance predicts response speed in an easy visual discrimination task," *Behavioral and Brain Functions*, vol. 7, p. 31, 2011.

[63] T. Ouyang and H.-T. Lu, "Vigilance Analysis Based on Continuous Wavelet Transform of EEG Signals," in *Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on*, pp. 1–4, 2010.

[64] D. Coufal, "EEG Signals Classification by S-shaped Radial Implicative Fuzzy Systems," in *Proceedings of the World Congress on Engineering and Computer Science 2011*, vol. 1, International Association of Engineers.

[65] J. S. Barlow, "EMG artifact minimization during clinical EEG recordings by special analog filtering," *Electroencephalography and Clinical Neurophysiology*, vol. 58, no. 2, pp. 161–174, 1984.

[66] W. Zhou and J. Gotman, "Removal of EMG and ECG artifacts from EEG based on wavelet transform and ICA," *Engineering in Medicine and Biology Society, 26th Annual International Conference of the IEEE*, vol. 1, pp. 392–395, 2004.

[67] J. Edmonds, H.L., L. Couture, and M. Paloheimo, "Clinical applications of combined EEG/EMG monitoring," in *Engineering in Medicine and Biology Society, Proceedings of the Annual International Conference of the IEEE*, vol. 4, pp. 1757–1758, 1988.

[68] M. Akin, M. Kurt, N. Sezgin, and M. Bayram, "Estimating vigilance level by using EEG and EMG signals," *Neural Computing & Applications*, vol. 17, pp. 227–236, 2008.

[69] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[70] A. Duchowski, *Eye tracking methodology: theory and practice*. Springer, 2007.

[71] T. Duong, H. Bui, D. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 838–845, 2005.

[72] Q. Ji and X. Yang, "Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[73] T. D'Orazio, M. Leo, P. Spagnolo, and C. Guaragnella, "A neural system for eye detection in a driver vigilance application," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pp. 320–325, 2004.

[74] M. Sigari, "Driver Hypo-vigilance Detection Based on Eyelid Behavior," in *Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on*, pp. 426–429, 2009.

[75] B. Wilhelm, H. Wilhelm, H. Lüdtke, M. Adler, and P. Streicher, "Pupillography for objective vigilance assessment. methodological problems and possible solutions," *Der Ophthalmologe, Zeitschrift der Deutschen Ophthalmologischen Gesellschaft*, vol. 93, no. 4, pp. 446–450, 1996.

[76] D. B. Henson and T. Emuh, "Monitoring vigilance during perimetry by using pupillography," *Investigative Ophthalmology & Visual Science*, vol. 51, no. 7, pp. 3540–3543, 2010.

[77] L. Deng, X. Xiong, J. Zhou, P. Gan, and S. Deng, "Fatigue Detection Based on Infrared Video Pupillography," in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, pp. 1–4, 2010.

[78] F. W. Hartel, E. Uhlenhuth, M. W. Fischman, and S. McCracken, "Reinforcement, skin conductance, and performance in a vigilance-reaction time task," *Psychiatry Research*, vol. 4, no. 2, pp. 239–251, 1981.

[79] W. Boucsein, A. Haarmann, and F. Schaefer, "Combining Skin Conductance and Heart Rate Variability for Adaptive Automation During Simulated IFR Flight," in *Engineering Psychology and Cognitive Ergonomics*, vol. 4562 of *Lecture Notes in Computer Science*, pp. 639–647, 2007.

[80] F. Friedrichs, M. Miksch, and B. Yang, "Estimation of lane data-based features by odometric vehicle data for driver state monitoring," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 611–616, 2010.

[81] Ford Europe, "Ford Technology Newsbrief 08-2010," 2010.

[82] R. Ibarra-Orozco, M. Gonzalez-Mendoza, N. Hernandez-Gress, F. Diederichs, and J. Kortelainen, "Towards a Ready-to-Use Drivers' Vigilance Monitoring System," in *Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation*, pp. 802–807, 2008.

[83] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, 1989.

[84] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[85] A. Wilson and A. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, 1999.

[86] Q. He and C. Debrunner, "Individual Recognition from Periodic Activity Using Hidden Markov Models.," in *Workshop on Human Motion*, pp. 47–52, 2000.

[87] R. Hughey and A. Krogh, "Hidden Markov models for sequence analysis: extension and analysis of the basic method," *Computer applications in the biosciences : CABIOS*, vol. 12, no. 2, pp. 95–107, 1996.

[88] D. Mount, "Using hidden Markov models to align multiple sequences.," *Cold Spring Harbor Protocols*, vol. 2009, no. 7, p. pdb top41, 2009.

[89] W. Zucchini and I. MacDonald, *Hidden Markov Models for Time Series: An Introduction Using R*. Monographs on Statistics And Applied Probability, Chapman and Hall, 2009.

[90] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau, "A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling," *Bioinformatics*, vol. 17, no. 12, pp. 1113–1122, 2001.

[91] W. Ching, E. Fung, and M. Ng, "Higher-order Markov chain models for categorical data sequences," *Naval Research Logistics (NRL)*, vol. 51, no. 4, pp. 557–574, 2004.

[92] A. L. Buchsbaum and R. Giancarlo, "Algorithmic aspects in speech recognition: An introduction.," *ACM Journal of Experimental Algorithmics*, p. 1.

[93] A. Yonezawa, T. Watanabe, M. Yokokawa, M. Sato, and K. Hirao, "Advanced Institute for Computational Science (AICS): Japanese national high-performance computing research institute and its 10-petaflops supercomputer "K"," in *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–8, 2011.

[94] P. Baldi and Y. Chauvin, "Smooth On-Line Learning Algorithms for Hidden Markov Models," 1994.

[95] J. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," tech. rep., International computer science institut, Berkley, CA., 1998.

[96] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[97] R. Y. Kahsay, G. Gao, and L. Liao, "An improved hidden Markov model for transmembrane protein detection and topology prediction and its applications to complete genomes," *Bioinformatics*, vol. 21, no. 9, pp. 1853–1858, 2005.

[98] M.-Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using a hidden Markov model type stochastic network," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 481 –496, 1994.

[99] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," in *Machine Learning*, vol. 32, pp. 41–62, 1998.

[100] A. Krogh, I. S. Mian, and D. Haussler, "A hidden Markov model that finds genes in E. coli DNA," *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4768–4778, 1994.

[101] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 729–734, 1982.

[102] P. Maybeck, *Stochastic models, estimation, and control*. Mathematics in science and engineering, Academic Press, 1982.

[103] J. D. Schutter, J. D. Geeter, T. Lefebvre, and H. Bruyninckx, "Kalman Filters: A Tutorial," *Journal A Benelux Quarterly Journal on Automatic Control*, vol. 40, no. 4, pp. 52–59, 1999.

[104] L. M. Arslan and J. H. Hansen, "Selective training for hidden Markov models with applications to speech classification," *IEEE*

Transactions on Speech and Audio Processing, vol. 7, no. 1, pp. 56–54.

[105] J.-T. Chien and C.-P. Liao, "Maximum Confidence Hidden Markov Modeling for Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 606–616, 2008.

[106] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust Sequential Data Modeling Using an Outlier Tolerant Hidden Markov Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1657–1669, 2009.

[107] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[108] Z. Zhang and T. U. of Iowa, *Linear Model Selection for Exactly and Nearly Replicated Data Based on Conceptual Predictive Statistics*. The University of Iowa, 2007.

[109] P. Pucar and M. Millnert, "Three techniques for state order estimation of hidden Markov models," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 3, pp. 1812–1815, 1995.

[110] R. J. Mackay, "Estimating the order of a hidden markov model," *Canadian Journal Of Statistics*, vol. 30, no. 4, pp. 573–589, 2002.

[111] T. Åkerstedt and M. Gillberg, "Subjective and Objective Sleepiness in the Active Individual," *International Journal of Neuroscience*, vol. 52, no. 1-2, pp. 29–37, 1990.

[112] K. Kaida, M. Takahashi, T. Åkerstedt, A. Nakata, Y. Otsuka, T. Haratani, and K. Fukasawa, "Validation of the Karolinska sleepiness scale against performance and EEG variables.," *Clinical Neurophysiology*, vol. 117, no. 7, pp. 1574–1581, 2006.

[113] M. Hofmann, R. Schleicher, N. Galley, and M. Golz, "eogui - matlab toolbox," 2011.

[114] O. Taramasco and S. Bauer, "R-package 'rhmm'." `http://r-forge.r-project.org/projects/rhmm`, `http://www.r-project.org`, 2012.

[115] R. W. Freund and R. H. W. Hoppe, *Stoer/Bulirsch: Numerische Mathematik 1*. Springer, 10th, revised edition ed., 2007.

[116] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero, "Shrinkage algorithms for MMSE covariance estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5016–5029, 2010.

[117] L. S. de Jong, "Towards a formal definition of numerical stability," *Numerische Mathematik*, vol. 28, pp. 211–219, 1977.

[118] D. Sandberg, "The performance of driver sleepiness indicators as a function of interval length," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1735–1740, 2011.

[119] O. Cappé, "Online EM Algorithm for Hidden Markov Models," 2009.

[120] C. Postelnicu, F. Barbuceanu, T. Topoleanu, and D. Talaba, "EOG-Based Interface for Manipulation tasks," *Applied Mechanics and Materials*, vol. 162, 2012.

[121] C.-C. Postelnicu, F. Girbacia, and D. Talaba, "EOG-based visual navigation interface development," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10857 – 10866, 2012.

DECLARATION

I hereby declare that this master thesis has been my independent work and has not been aided with any prohibited means.

I declare, to the best of my knowledge and belief, that all passages taken from published and unpublished sources or documents have been reproduced whether as original, slightly changed or in thought, have been mentioned as such at the corresponding places of the thesis, by citation, where the extent of the original quotes is indicated.

The paper has not been submitted for evaluation to another examination authority or has been published in this form or another.

Vienna, May 2012

Christoph Schneider