



ulm university universität
uulm

Fahrerintentionserkennung und Kursprädiktion mit erweiterten Maschinellen Lernverfahren

DISSERTATION

zur Erlangung des Doktorgrades Dr. rer. nat.
der Fakultät für Ingenieurwissenschaften und Informatik
der Universität Ulm

Joachim Börger
aus Aachen

Fakultät für Ingenieurwissenschaften und Informatik
Institut für Neuroinformatik
Institutsleiter: Prof. Dr. Günther Palm

2013

Amtierender Dekan: Prof. Dr.-Ing. Klaus Dietmayer
Gutachter: Prof. Dr. Günther Palm
Gutachter: Prof. Dr. Dr.-Ing. Wolfgang Minker
Tag der mündlichen Prüfung: 11.12.2012

Kurzfassung

Vorausschauende Fahrerassistenzsysteme (Abb. 1) benötigen eine präzise Kenntnis über den zukünftigen Verlauf der Verkehrssituation; dieser kann in Echtzeitsystemen nur anhand vorheriger Beobachtungen geschätzt werden. Im Anwendungsbeispiel wird der Fokus auf die Schätzung der Fahrerintention und des zukünftigen Fahrzeugkurses gelegt, welche in der Form diskreter Manöver oder einer kontinuierlichen Größe vorhergesagt werden. Die Abbildung eines solch komplexen Anwendungsfalls auf die formale Ebene der Klassifikation wird durch eine saubere Problembeschreibung sowie die Definition der zu optimierenden Kriterien ermöglicht.

Im methodischen Kern dieser Arbeit wird ein Modell zur Zeitreihenanalyse erarbeitet, welches für die Fahrerassistenz Prädiktionen zukünftiger diskreter Ereignisse als auch kontinuierlicher Signalverläufe ermöglicht. Diese komplexen Schätzungen werden durch verschiedene Verfahren Maschinellen Lernens durchgeführt, die anhand von Trainingsdaten die relevanten Indikatoren und deren Abbildung auf die Schätzgröße suchen. Dafür werden zwei bekannte Algorithmen neu miteinander kombiniert: Der Zeitreihencharakter der Daten wird durch *Hidden Markov Modelle* strukturiert, während die hohe Dimensionalität und Komplexität der Eingangssignale durch den *Random Forest Algorithmus* berechnet wird.



Abbildung 1: Vorausschauende Fahrerassistenzsysteme in Aktion: In dieser unerwarteten Stausituation können durch die Umfeldsensorik verschiedene Funktionen ausgeübt werden, zum Beispiel eine Alarmierung und Bremsunterstützung des Fahrers (*Predictive Emergency Braking Systems*) oder automatische Geschwindigkeitskontrolle (*Adaptive Cruise Control*).

Inhaltsverzeichnis

Kurzfassung	iii
1 Einleitung	1
1.1 Motivation	2
1.2 Erweiterungsbedarf bestehender Untersuchungen	3
1.3 Zielsetzung	4
1.4 Gliederung	5
1.5 Notationen	5
2 Stand der Technik	11
2.1 Unfallursachen und Nutzenanalyse	11
2.2 Fahrerassistenzsysteme	12
2.3 Fahrermodelle	14
2.4 Fahrer als Reglersysteme	15
2.5 Fahrstile	17
2.6 Fahrdynamik und Kursprädiktion	18
2.7 Fahrerintentionserkennung	20
2.8 Reaktionszeit	22
2.9 Kritikalitätsempfinden	23
2.10 Maschinelles Lernen	24
2.11 Zusammenfassung	25
3 Grundlagen der Fahrerassistenz für die Kollisionsvermeidung	27
3.1 Beispielszenen	27
3.2 Fahrermodell	30
3.3 Warnstrategie	30
3.4 Warndilemma	31
3.5 Zusammenfassung	32
4 Modellierung und Evaluierung der Fahrerintention	35
4.1 Probabilistisches Framework	35
4.2 Highlevelmanöver	37
4.3 Deeskalationsmanöver	38
4.4 Kursprädiktion	40
4.5 Evaluierung	42
4.5.1 Generische Auswertungen	42
4.5.2 Zeitliche Verzögerung	44
4.5.3 Erweiterte bedingte Sensitivität	44

4.5.4	Auswertung der Kursprädiktion	45
4.5.5	Nutzen des Warnsystems zur Unfallvermeidung	46
4.5.6	Diskussion der Auswertungsmöglichkeiten	47
4.6	Zusammenfassung	48
5	Untersuchte Signaldaten und Vorverarbeitung	51
5.1	Sensorik und Signale	52
5.2	Verwendung der Multiobjektdateien	52
5.3	Szeneneinteilung und Kritikalität	54
5.4	Scheinziele	56
5.5	Transformation der Kursparameter	58
5.5.1	Zielkoordinaten	58
5.5.2	Normierung der Kurskrümmung anhand der Komfortgrenzen	59
5.5.3	Differenzkodierung der Parameter	59
5.6	Merkmalsextraktion	60
5.6.1	Generische Merkmale	60
5.6.2	Kursversatz	61
5.6.3	Time to collision (TTC)	61
5.6.4	Vermeidungsbeschleunigung	62
5.6.5	Reaktionszeitpunkte	62
5.6.6	Time To React (TTR)	63
5.6.7	Vergleich	64
5.7	Zusammenfassung	65
6	Grundlagen der Datenverarbeitung	67
6.1	Hidden Markov Modelle	67
6.1.1	Algorithmen	69
6.1.2	Modellierung eines Zeitschritts	71
6.1.3	Ensemble-HMMs zur Klassifikation	73
6.1.4	Diskriminatives Training mit Mutual Information	74
6.1.5	Bedeutung der Zustände	75
6.1.6	Zusammenfassung	76
6.2	Random Forest	76
6.2.1	Occam's Razor	76
6.2.2	Konstruktion eines Entscheidungsbaums	77
6.2.3	Randomisierung	79
6.2.4	Prädiktion	79
6.2.5	Zusammenfassung	79
6.3	Moving Window	80
6.4	Bayes-Fehler	81
6.5	Bestimmung der Abtastfrequenz	83
6.6	Zusammenfassung	84
7	Methodik	85
7.1	Growing Window Sequenzen	86
7.2	Klassifikations-HMMs	87
7.2.1	Konstruktion	87

7.2.2	Klassifikation und Beweis der Äquivalenz	88
7.2.3	Training mit dem Baum Welch Algorithmus	90
7.2.4	Vorteile	90
7.2.5	Verwandte Arbeiten	91
7.3	Beobachtungen mit Random Forest	92
7.3.1	Ein Random Forest Modell für alle Zustände	93
7.3.2	Ein Random Forest Modell je Zustand	93
7.4	Struktur des HMMs	95
7.4.1	Vermeidung isolierter Zustände	95
7.4.2	Variation der Zustandsverweildauern	96
7.4.3	Modellierung expliziter Ablaufphasen	98
7.4.4	Explizite Pfade im HMM je Klasse	98
7.5	Adaptive Abtastpunkte für Moving Window	99
7.6	Random Forest und Weighted Classes	101
7.7	Kursprädiktion durch Random Forest	102
7.7.1	Diskretisierung	102
7.7.2	Prädiktion der Kursparameter	103
7.8	Zusammenfassung	103
8	Resultate	105
8.1	Datenbeschreibung	105
8.2	Verfahren und Parametrierungen	106
8.3	Bayes-Fehler	108
8.4	Random Forest zur Fahrerintentionserkennung	109
8.5	Größe der HMMs und Datenfrequenz	113
8.6	Growing Window	115
8.7	Klassifikations-HMMs	116
8.8	Beobachtungen mit Random Forest	118
8.9	Struktur des HMMs	121
8.9.1	Vermeidung isolierter Zustände	122
8.9.2	Variation der Zustandsverweildauern	123
8.9.3	Untersuchung der trainierten Zustände	124
8.10	Kursprädiktion mit Random Forest	126
8.11	Gesamtvergleich der Methoden	129
8.11.1	Fahrerintentionserkennung	130
8.11.2	Korrektheit der Warnfunktion	132
8.11.3	Nutzen des Warnsystems zur Unfallvermeidung	133
8.11.4	Ergebnisse im Vergleich	134
8.12	Zusammenfassung	134
9	Zusammenfassung und Fazit	137
9.1	Zusammenfassung	137
9.2	Fazit	139
9.3	Ausblick	139

1 Einleitung

Vorausschauende Fahrerassistenzsysteme basieren auf einer Prädiktion der Verkehrssituation. Wird ein Fahrfehler detektiert, kann ein frühzeitiger Systemeingriff bis zu 75% der Auffahrunfälle vermeiden und damit Leben retten (Kapitel 2.1). Unberechtigte Eingriffe z.B. in knappen oder sportlichen Fahrsituationen müssen jedoch dringend vermieden werden: Sie können nicht nur zu Akzeptanzproblemen, sondern durch Manipulation des Fahrgeschehens sogar zu einem erhöhten Risiko führen [FAB08].



Abbildung 1.1: Eskalierende Auffahrszene, aus der leicht eine Kollision entstehen könnte. Um nicht mit dem verzögernden Vorderfahrzeug zu kollidieren, muss der Fahrer diesem auszuweichen oder ebenfalls sein Tempo verringern. Falls er nicht reagiert, würde er bei unveränderter Fahrweise eine Kollision verursachen.

Je nach Fahrsituation ist die Interpretation der Situation mehr oder weniger eindeutig, so dass der Handlungsbedarf zur Fahrerunterstützung teilweise falsch eingeschätzt wird (Abb. 1.1 und 1.2). Schon durch den Bremsweg und die Fahrerreaktionszeit muss mit wachsenden Geschwindigkeiten auch der Prädiktionszeitraum auf bis zu $\Delta t = 5s$ ansteigen. Zeitlich höhere Prädiktionen bringen auch zwangsläufig eine höhere Irrtumswahrscheinlichkeit mit sich, was in Form des *Warndilemmas* ausführlich diskutiert wird (Kapitel 3.4).

Zur Erkennung von zukünftigen Gefahrensituationen kann die Situation physikalisch extrapoliert werden: Die Fahrzeuggeschwindigkeit, Beschleunigung und Kurskrümmung sowie mittels Umfeldsensorik detektierte Fremdobjekte erlauben eine rudimentäre Situationsprädiktion. Jedoch haben sicherheitskritische Fahrerassistenzfunktion stets mit der Herausforderung zu kämpfen, dass der Fahrer sich in den letzten Sekunden noch unentscheiden kann; diese Kurzzeitdynamik, etwa ein in einer Sekunde erfolgreicher Lenkeingriff, ist zum relevanten Zeitpunkt nicht explizit in den sensorischen Daten ersichtlich und kann nur anhand typischer Anzeichen geschätzt werden.

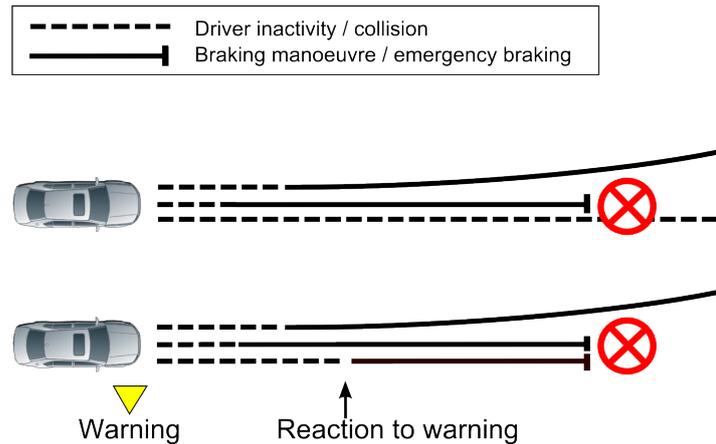


Abbildung 1.2: Verschiedene Möglichkeiten des Situationsausgangs. Alle drei Möglichkeiten sind oben dargestellt; unten dargestellt ist die Annahme, dass eine Warnung den Fahrer zu einer Vollverzögerung veranlassen würde, während sie in den anderen Fällen überflüssig ist.

Maschinelles Lernen ist eine Methode der künstlichen Intelligenz, welche genau solche Herausforderungen adressiert: Anhand zahlreicher Indikatorsignale wird nach Mustern gesucht, die eine Situationsprädiktion und die Schätzung zukünftiger Fahrerhandlungen verbessern. In bisherigen Untersuchungen wurde jedoch meist nur eine eingeschränkte Form der Fahrerintentionserkennung behandelt: Fahrmanöver wie Spurwechsel oder Einparken werden erkannt, sind jedoch für den Anwendungsfall der Kollisionsdetektion nicht ausreichend feingranular. Zudem erzielen komplexe Algorithmen zwar häufig gute Klassifikationsraten und zuverlässige Schätzer; was genau diese Modelle gelernt haben, ist jedoch häufig nur schwer interpretierbar und erweitert damit nicht den Erfahrungshorizont.

In dieser Arbeit wurde das Ziel verfolgt, basierend auf bisherigen Erkenntnissen der Fahrerintentionserkennung ein universelles Modell zu erhalten, welches für verschiedene Anwendungsfälle eingesetzt werden kann. Auf methodischer Seite wurde hierzu die Analyse von Zeitreihendaten vertieft und auf die Komplexität des Problems hin optimiert. Durch formale Schritte wie die saubere Definition der Fahrerintention, Optimierungskriterien sowie eine Rückführung der Erkenntnisse Maschinellen Lernens zu Expertenwissen wurde eine Brücke zwischen Anwendung und Wissenschaft gebaut.

1.1 Motivation

Kollisionswarnsysteme haben bereits Einzug in die Serienproduktion gehalten; die Erfahrungen zeigen jedoch, dass bei sportlicher Fahrweise und spätem Steuern noch relativ häufig unberechtigte Unterstützungen stattfinden. Diese können die Akzeptanz des Fahrers beeinträchtigen oder sogar ein Risiko für den Folgeverkehr darstellen und begründen, dass eine fundierte Berücksichtigung von Unsicherheit sowie unterschiedlicher Fahrweisen nötig ist.

Seriensysteme kompensieren diese Unsicherheit, indem Unterstützungen nur in ausgewählten Szenarien erfolgt, in welchen aufgrund von Expertenwissen ein hinreichendes Maß an Zuversicht angenommen wird; die Fahrerintention wird dabei meist durch heuristische Regeln geschätzt. Mit diesem Prinzip kann zwar ein bestimmter Szenekatalog abgehandelt werden, es entbehrt jedoch der Universalität und damit der allgemeineren Verwendbarkeit (Abb. 1.3).

Weil neben Kollisionswarnsystemen auch zahlreiche andere Fahrerassistenzsysteme ermöglicht werden sollen, muss ein allgemeinerer Ansatz gefunden werden, um die Situationsprädiktion generisch durchzuführen. Dieser kann zukünftig die Grundlage für verschiedenartige Assistenzfunktionen bilden und unter Verwendung eines probabilistischen Konfidenzmaßes anhand einer individuellen Strategie die Unterstützung berechnen.

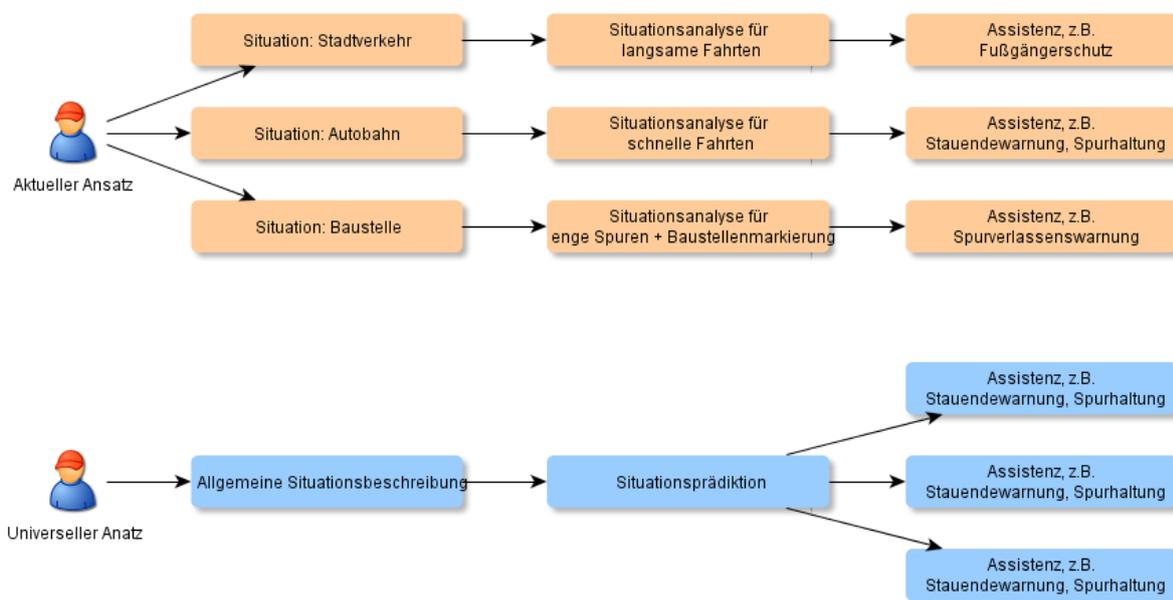


Abbildung 1.3: Vorteile eines universellen Ansatzes zur Situationsprädiktion. Oben: Zumeist werden Fahrerassistenzsysteme anwendungsspezifisch entwickelt und nur in sehr speziellen Szenarien angewandt. Neue Szenarien oder Funktionen sind damit stets mit hohem Aufwand verbunden. Unten: Die Situationsanalyse und -prädiktion geschieht universell, verschiedenartige Systeme können spezifisch darauf reagieren. Dadurch wird gemeinsames Wissen ausgenutzt und für mehr Funktionalitäten zur Verfügung gestellt.

1.2 Erweiterungsbedarf bestehender Untersuchungen

Im aktuellen Stand der Technik (Kapitel 2) wird meist Maschinelles Lernen zur Erkennung vordefinierter Fahrmanöver verwendet (Spurwechsel, Kurve...), die im Folgenden als *Highlevel-Klassen* Ψ^H bezeichnet werden. Ein wichtiges Ergebnis dieser Arbeiten ist die grundsätzliche Anwendbarkeit sowie Erfahrungen mit Maschinellern Lernen. Viele Studien basieren jedoch allein auf Simulatordaten, welche das Ergebnis in der Realität verfälschen; teilweise wird zusätzliche Sensorik (z.B. Feststellung der Blickzuwendung, Muskulatur oder Puls) verwendet, die in Serienfahrzeugen nicht zur Verfügung steht (Kapitel 2.7).

Darüberhinaus liegt zwar die Erkennensrate teilweise sehr hoch (z.B. für Spurwechsel bis zu 100%); der fehlende Bezug zur Kollisionsdetektion schränkt die Ergebnisse jedoch ein und stellt die universelle Bedeutung in Frage. In anderen Arbeiten sind die Fehlerkennungsraten zu hoch: Auch wenn diese häufig nur im Prozentbereich liegen, entstehen bei der hohen Fahrleistung der heutigen Gesellschaft in der Praxis schnell Probleme.

In der Industrie werden minimale und beherrschbare Fehlerraten gefordert - beim verbreiteten Six-Sigma-Prinzip sind dies entsprechend einer Standardnormalverteilung Ausfallraten unterhalb von 3.4 *DPMO* (*Defects Per Million Opportunities*) bzw. 0.00034%. Das Systemgeschehen muss also nachvollziehbar und absicherbar sein, ebenso werden Aussagen über die jeweilige Schätzsicherheit benötigt.

1.3 Zielsetzung

Die zukünftige Handlung des Fahrers, repräsentiert durch die *Fahrerintention*, ist das zentrale Element der angestrebten Situationsprädiktion. Sie wird zunächst durch diskrete Manöverklassen modelliert und mittels Maschinellen Lernens anhand von Indikatorsignalen geschätzt. Dieser bekannte Ansatz weist jedoch einige grundlegende Schwächen der diskreten Modellierung auf, zum Beispiel die Mehrdeutigkeit und nicht genau definierbare Anfangs- und Endzeitpunkte der Manöver.

Als methodische Neuerung wird deshalb die Fahrerintention zusätzlich durch den zukünftigen Fahrzeugkurs ermittelt, welcher viele weitere Anwendungsmöglichkeiten zulässt. Ein großer Unterschied zur Fahrmanövererkennung ist, dass die Schätzgröße nun kontinuierlich wird. Sie wurde bisher fast ausschließlich für gleichmäßige (stationäre) Fahrten untersucht. In dieser Arbeit werden für dynamische Fahrsituationen (Kurveneingänge, Ausweichen...) nun ebenfalls erweiterte Indikatorsignale hinzugezogen. Dafür wird ein parametrierbares Fahrmodell entworfen, welches gleichzeitig physikalische Plausibilität garantiert.

Für die Analyse solcher Indikatorsignale stellt der *Random Forest* Algorithmus eines der erfolgreichsten Verfahren dar; er kann jedoch per se keine *zeitlichen Informationen* analysieren. Genau für diese Zwecke sind *Hidden Markov Models* ein klassisches Prinzip, die ihrerseits Schwächen auf den komplexen Beobachtungen zeigen. Beide Verfahren werden miteinander kombiniert, wovon sowohl ein verbesserter Schätzer als auch gesteigerte Robustheit bezüglich der Sensordaten erwartet wird.

Die Erweiterungen wurden weitestgehend generisch durchgeführt und sind auch für andere Gebiete der Mustererkennung verwendbar. Durch eine hohe Transparenz in den verwendeten Methoden wird eine genaue Auswertbarkeit und Interpretierbarkeit der Ergebnisse angestrebt, um auch Erkenntnisse über die Gestalt der Daten selbst zu erhalten.

1.4 Gliederung

Abschnitt 1.5 fasst zunächst zentrale verwendete Begriffe und Symbole zusammen. Kapitel 2 stellt wichtige Grundlagen und den Stand der Technik bezüglich Fahrerassistenz und Maschinellen Lernen dar.

Für das Anwendungsgebiet der Kollisionsvermeidung werden in Kapitel 3 wichtige physikalische und psychologische Grundlagen hergeleitet, woraus schließlich das durchgängig verwendete Fahrermodell hervorgeht. Das Warndilemma begründet die Notwendigkeit der Fahrerintentionsschätzung, welche in Kapitel 4 zunächst formal modelliert wird. Ebenso wichtig ist die Definition der Auswertemetriken (Kapitel 4.5), um Schätzer beurteilen zu können. Die Vorverarbeitungsschritte speziell für Fahrerassistenzdaten (Kapitel 5) beschließen den anwendungsspezifischen Teil.

Kapitel 6 stellt die verwendeten Klassifikatoren in ihrer Grundfunktionalität vor. Im methodischen Teil dieser Arbeit werden darauf aufbauend Klassifikations-HMMs zur praktischen und flexiblen Analyse unendlich langer Datensequenzen erarbeitet und mit Random Forest kombiniert (Kapitel 7).

Die Methoden der diskreten Fahrerintentionsschätzung und Kursprädiktion werden in Kapitel 8 validiert und verglichen. Hierbei werden sowohl generische als auch anwendungsspezifische Metriken verwendet, um schließlich das Potenzial zur Unfallvermeidung zu beurteilen. Kapitel 9 fasst die erzielten methodischen Fortschritte und Resultate zusammen und interpretiert deren Bedeutung für verwandte Probleme der Zeitreihenanalyse. Die Arbeit endet mit einem Ausblick über weitere Einsatzmöglichkeiten dieser Verfahren sowie noch offenem Verbesserungspotenzial.

1.5 Notationen

Zeitreihendarstellung

Die in dieser Arbeit verwendeten Signale entstammen verschiedenen Steuergeräten zur Erfassung der Umfeld-, Fahrzeug- und Fahrerdaten. Die Steuergeräte arbeiten asynchron und der verwendete Sensor für Umfelddaten zusätzlich mit einer variablen Taktdauer, was u.a. aus der sich verändernden Komplexität der Umfelddaten resultiert.

Für die durchgeführten Untersuchungen wurden die Daten teilweise durch lineare Interpolation auf kontinuierliche Zeitpunkte erweitert, was die formale Darstellung und Umsetzung der Algorithmen stark vereinfacht. Somit entspricht jedes Signal f einer kontinuierlichen Funktion

$$f : \mathbf{R}^{\geq 0} \rightarrow \mathbf{D}, \quad D \in \{\mathbf{R}, \mathbf{N}, \{\text{true}, \text{false}\}\} \quad (1.1)$$

die jedem Zeitpunkt t ein kontinuierliches, diskretes oder binäres Signal zuordnet. Ebenso ist eine Klassifikation Ψ oder ein Schätzer definiert, mit $D = \mathbf{N}$. Der Zeitpunkt t_0 selbst bezeichnet den Beginn der Daten; liegen bei Offline-Verwendung mehrere einzelne Messungen vor, werden sie zu einer Sequenz aneinandergehängt und die Szenenbeginne markiert (Kapitel 5.3).

Begriffe der Fahrerassistenz

Egofahrzeug	Das Fahrzeug, welches mit dem untersuchten Fahrerassistenzsystem und Umfeldsensorik ausgestattet ist
Egofahrer	Fahrer des Egofahrzeugs
Objekt	Ein anderes, per Umfeldsensorik wahrgenommenes Objekt der Verkehrsszene: Zum Beispiel andere Fahrzeuge, Verkehrsschilder, Fußgänger oder auch Scheinziele wie Gullideckel
Intention	Von lateinisch <i>intentio</i> : Absicht, Vorsatz, Vorhaben. Im Kontext der Fahrerassistenz beschreibt die Fahrerintention den vom Fahrer geplanten zukünftigen Vorgang, modelliert als Fahrmanöver oder zukünftigen Fahrzeugkurs.
Fahrmanöver	Ein eindeutig benennbares Manöver, z.B. Anhalten, Ausweichen
Kursprädiktion	Kontinuierliche Schätzung des zukünftig gefahrenen Kurses in Form physikalischer Parameter oder einer Trajektorie
Trajektorie	Menge von geometrischen Positionen, welche einen Fahrzeugkurs beschreibt
Fahrschlauch	Konkrete Realisierung einer Kursprädiktion durch zirkuläre Extrapolation der Fahrzeugposition
Prädiktionshorizont	Zeitraum Δt , über den eine Situation oder ein Fahrzeugkurs in die Zukunft prädiziert wird
Komfortgrenzen	Maximal vom Fahrer tolerierte bzw. erreichbare Fahrzeugbeschleunigungen in Längs- oder Querrichtung
Reaktionszeit	Zeit zwischen einer Warnung an den Fahrer und dessen Reaktion auf diese Warnung
Warnzeitpunkt	Der Zeitpunkt, zu dem ein Fahrer gewarnt werden muss, damit er anhand seiner Reaktionszeit auf diese Warnung und seinem Fahrvermögen eine Kollision verhindern kann.

Abkürzungen

TTC	<i>Time to collision</i> : Die geschätzte Zeit bis zur Kollision, welche anhand des Abstandes, der Relativgeschwindigkeit und u.U. der relativen Beschleunigung zwischen dem Egofahrzeug und dem betrachteten Objekt berechnet wird
TTR, TTBrake	Verbleibende Zeit, bis eine allgemeine Reaktion (<i>Time to react</i>) oder bestimmtes Manöver (<i>Time to brake</i>) erforderlich wird, um die Situation zu deeskalieren.
PEBS	Predictive Emergency Braking Systems: Vorausschauende Fahr-sicherheitssysteme, die derzeit die Funktionen Warnung, Teilbremsung zur Reaktionszeitverlängerung, Bremsunterstützung oder Automatische Notbremsung beinhalten
ACC	Adaptive Cruise Control, ein Abstands- und Geschwindigkeits-regler, der meist auf Umfeldsensorik durch ein Radarsystem ba-siert
HMM	Hidden Markov Model
RF	Random Forest

Allgemeine Symbole

t	Ein einzelner Zeitpunkt, je nach Kontext kontinuierlich $t \in \mathbf{R}$ oder diskret $t \in \mathbf{N}$.
o_t	Beobachtungsvektor zum diskreten Zeitpunkt t
$f_t, f(t)$	Merkmalsvektor zum diskreten oder kontinuierlichen Zeitpunkt t
$S = (a, b)$	Beginn und Ende einer Szene, $a, b \in \mathbf{R}$
$S = o_1 \dots o_T$	Sequenz von Beobachtungen
c	Konstante oder Parameter eines Modells
q	Zustand eines HMMs
$ x $	Länge des Vektors x
$ M $	Kardinalität der Menge M
$p(X = x), p(x)$	Wahrscheinlichkeitsdichte der Zufallsvariable X für den Wert x . Da die Werte meist mit dem Kleinbuchstaben der Zufallsvariable beschrieben werden, wird die zweite Schreibweise abkürzend verwendet.

Spezielle Symbole

ω	Frequenz der aufgenommenen Daten oder verwendete Abtastrate für HMMs oder Moving Window.
a_ψ	Vermeidungsbeschleunigung, so dass mit diesem Parameter eine Kollision durch das Manöver ψ verhindert werden kann.
$t_{\psi,C}, t_{\psi,E}$	Letztmöglicher Zeitpunkt, zu dem durch das Fahrmanöver ψ eine Kollision noch vermieden werden kann, unter Verwendung der Komfortparameter (C) oder Notfallparameter (E).
$\Delta t_{\psi,C}$	Differenz zwischen beschriebenem und aktuellem Zeitpunkt. z.B. $\Delta t_{\text{Brake},C}$ = verbleibende Zeit zu einer Komfortbremsung (TTBrake).
t_{Warn}	Zeitpunkt, zu dem eine Warnung erfolgt
d_x, d_y	Abstand eines Objektes in Längs- und Querrichtung Geschwindigkeit in Längs- und Querrichtung Beschleunigung in Längs- und Querrichtung Kurskrümmung, entspricht dem Reziprok des Kurvenradius r bzw. 0 bei Geradeausfahrt. κ ist der Koeffizient des zweiten Polynoms bei der Taylor-Entwicklung der Kreisgleichung (stationäre Kurvenfahrt).
v_x, v_y	
a_x, a_y	
κ	

Klassenbezeichnungen

$\Psi, \hat{\Psi}$	Klassifikation bzw. deren Schätzung, welche jedem Zeitpunkt t eine diskrete Information, z.B. ein Fahrmanöver oder Label zuweist.
Ψ^{Warn}	Warnfunktion Kollisionsindikatorfunktion, sie ist erfüllt wenn ohne Fahrereingriff eine Kollision droht und wird anhand der Objektposition zum Fahrerschlauch berechnet.
Ψ^{Coll}	
Ψ^{Critical}	Indikatorfunktion für subjektive Kritikalität
Ψ^C	Der zukünftige Fahrzeugkurs, dargestellt durch eine Trajektorie

Highlevel-Manöver

Ψ^H	Zufallsvariable für das derzeitige Highlevel-Manöver, mit den 4 folgenden Ausprägungen.
ψ_{Neutral}	Neutrale Klasse, wenn kein Manöver stattfindet
ψ_{LCL}	Ausweichen Links
ψ_{LCR}	Ausweichen Rechts
$\psi_{\text{StopBehindCar}}$	Bremsen

Deeskalationsmanöver

Ψ^D	Zufallsvariable für das derzeitige Deeskalationsmanöver, mit den 5 folgenden Ausprägungen.
ψ_{Neutral}	Neutrale Klasse, wenn die Situation unkritisch ist.
ψ_{Left}	Ausweichen Links
ψ_{Right}	Ausweichen Rechts
ψ_{Brake}	Bremsen, bis keine weitere Annäherung stattfindet
ψ_{Coll}	Kollision

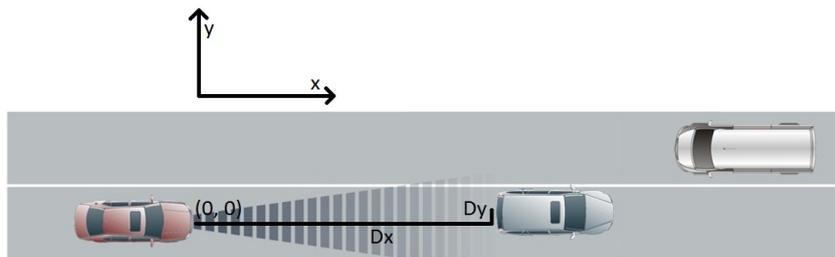


Abbildung 1.4: Verwendetes Koordinatensystem: Der Ursprung liegt im Einbauort des Radarsensors (mittige Fahrzeugfront); soweit nicht anders angegeben, bewegt sich das Koordinatensystem mit dem Egofahrzeug, so dass sämtliche Umfelddaten relativ zu interpretieren sind. In dieser Grafik ist zudem der laterale Öffnungswinkel des LRR3-Sensors von bis zu 30° schematisch dargestellt

Koordinatensystem

Es werden durchgängig rechtshändige Koordinatensysteme verwendet (Abb. 1.4). Der Koordinatenursprung liegt, soweit nicht anders angegeben, in der frontal mittigen Position des Egofahrzeugs, wo üblicherweise auch ein Radarsensor verbaut ist. Die x-Achse zeigt entlang der Fahrzeugachse (longitudinal), die y-Achse zeigt nach links (lateral)

Abbildungen

Szenen werden in dieser Arbeit auf zwei Arten dargestellt: Die einfache Art ist ein Videobild, welches ein schnelles Verständnis und subjektiven Eindruck vermittelt. Dieses kann jedoch nur die Daten eines einzelnen Zeitpunktes darstellen. Deshalb wird zusätzlich zu meist die Vogelperspektive verwendet, für welche zuvor der gefahrene Kurs odometrisch berechnet wurde. Sie erlaubt eine Visualisierung des gefahrenen Kurses und damit des zeitlichen Ablaufs in einer Szene (Abb. 1.5).

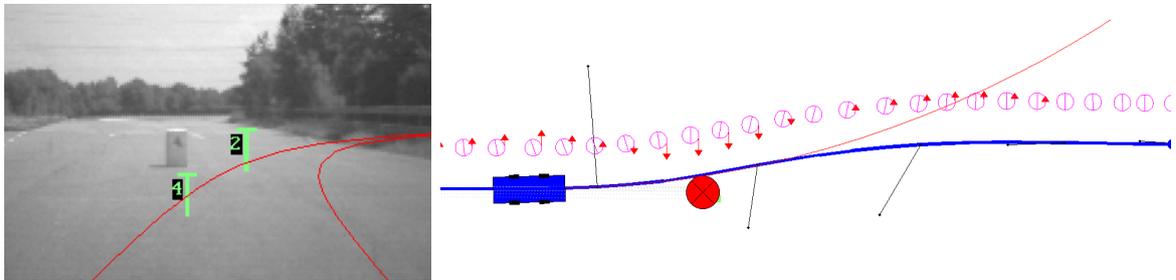


Abbildung 1.5: Darstellung von Fahrscenen. Links: In dieser Szene fährt der Fahrer in einer Rechtskurve an einem Schaumstoffwürfel mit montiertem Reflektor vorbei. Zu sehen ist das Videobild, die Kamera ist unter dem Rückspiegel verbaut. Rote Linien stellen den kreisförmig prädizierten Fahrzeugkurs dar, grüne Linien die Positionen und Standardabweichungen der Objekte, grüne Ziffern deren Nummer.

Rechts: Eine andere Szene, in der einem Objekt mit schnellen Links-Rechts-Lenkungen ausgewichen wird. Wiederum ist die Kursprädiktion zu sehen (rot), die kombinierten Längs- und Querschleunigungsvektoren (schwarz), die Lenkradwinkelstellung (violett) und die Lenkradbewegung (rote Pfeile).

2 Stand der Technik

Fahrerassistenz verbindet die Disziplinen Fahrzeugdynamik, Unfallforschung, Psychologie und künstliche Intelligenz miteinander. Dieses Kapitel gibt eine Einführung in die zunächst verschiedenen und voneinander unabhängigen Gebiete. Dabei werden die für den Kontext der vorliegenden Arbeit relevanten Zusammenhänge herausgearbeitet und wichtige verwendete Grundlagen vorgestellt.

Einführend werden häufige Unfallursachen sowie das geschätzte Potenzial zur Unfallvermeidung dargestellt, um schließlich zu einem Überblick über Fahrerassistenzsysteme zu kommen. Einher geht die Spezialisierung auf vorwärtsgerichtete Kollisionsvermeidungssysteme mit den verschiedenen Ausbaustufen, wobei der Fokus auf Warnsystemen liegt. Hierfür ist ein gutes Verständnis des Fahrers, seines Fahrstils und Reaktionsvermögens nötig, was durch Fahrermodelle, Untersuchungen zum Kritikalitätsempfinden sowie der Schätzung der Reaktionszeit detailliert wird.

Um schließlich situativ die beste Unterstützung zu ermöglichen, sind Kursprädiktion und Fahrerintentionserkennung nötig, wodurch Objekte im Gefahrenbereich identifiziert und mit der Fahrerhandlung abgeglichen werden können. Die methodische Verbindung zwischen diesen beiden anwendungsspezifischen Schätzungen zur Analyse und Prädiktion sequentieller Daten sind ein wichtiges Ergebnis dieser Arbeit; demzufolge endet dieses Kapitel mit einer kurzen Einführung in aktuelle Verfahren des Maschinellen Lernens, welche für die Fahrerintentionsschätzung verwendet und weiterentwickelt wurden.

2.1 Unfallursachen und Nutzenanalyse

„The price paid for mobility in Europe is still far too high.“

Mit diesen Worten erklärte 2001 die Europäische Kommission ihr ehrgeiziges Ziel, die Anzahl der Verkehrstoten bis 2010 zu halbieren [Eur01, S. 65]. Im Jahre 2000 kamen europaweit noch über 40000 Menschen durch Verkehrsunfälle ums Leben. Bis 2010 konnte mit einer Verringerung um 36% ein großer Erfolg erzielt werden, und bis 2020 wird darauf basierend erneut eine Halbierung der Anzahl Verkehrstoten angestrebt [Eur10].

Einen wichtigen analytischen Ansatzpunkt für dieses Ziel bietet die German In-Depth Accident Study (GIDAS) [GZL⁺09]. Sie umfasst zwei Teams, die in den Großräumen

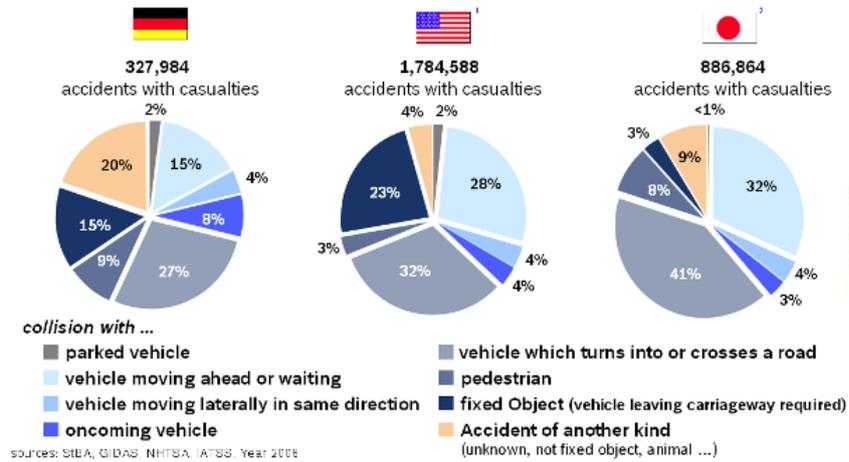


Abbildung 2.1: Unfallursachen in Deutschland, USA und Japan getrennt nach Situation. Der sehr hohe Anteil von Auffahrunfällen zeigt deutliches Potenzial, das Verkehrsgeschehen durch vorwärtsgerichtete Kollisionsvermeidungssysteme sicherer zu machen [GZL⁺09].

Hannover und Dresden nach einem Unfall den Hergang detailliert rekonstruieren. Jährlich werden dabei Daten zu etwa 2000 Unfällen erhoben, die zur Ursachenanalyse sowie Simulationen von Vermeidungsmaßnahmen verwendet werden.

Auffahrunfälle machen einen hohen Anteil von 15% in Deutschland aus, in den USA und Japan sind es gemittelt sogar 30% (Abb. 2.1). In einer genaueren Aufschlüsselung wurde ermittelt, dass drei von vier dieser Szenen Frontalkollisionen mit vorausfahrendem oder stehendem Verkehr wie Stauenden waren [HWB09]. Diese Situationen decken also einen Großteil der Unfälle ab und motivieren somit stark die Untersuchung vorwärtsgerichteter Kollisionsvermeidungssysteme.

Mit der GIDAS Datenbank untersuchten Georgi et al. den Nutzen verschiedener solcher Systeme [GZL⁺09]: Im Mittel konnten durch eine Warnfunktion (PCW) 38%, durch zusätzliche Bremsunterstützung (EBA) 57% und durch die letzte Ausbaustufe autonomes Bremsen (AEB) sogar 75% aller Unfälle vermieden werden (Abb. 2.2).

2.2 Fahrerassistenzsysteme

Motiviert durch das hohe Potenzial zur Unfallvermeidung ergeben sich vielfältige Möglichkeiten, die starke Fehlerquelle des menschlichen Fahrers zu unterstützen. Gröndl gibt einen Überblick über bekannte Fahrerassistenzsysteme [Grü05]:

- Adaptive Cruise Control (ACC): Umfeldbasierte Reglersysteme, welche Geschwindigkeit und Abstand zum Vorderfahrzeug an den Verkehrsfluss anpassen.
- Kollisionsvermeidungssysteme, durch Warnfunktionen oder autonome (Teil-) Bremsengriffe

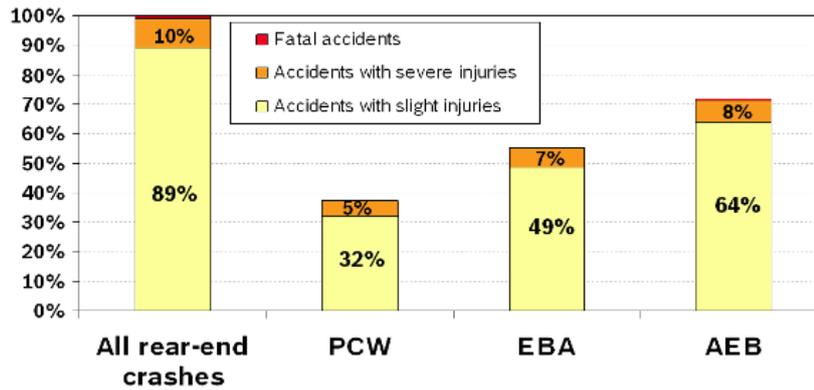


Abbildung 2.2: Anteil vermeidbarer Unfälle durch verschiedene Ausbaustufen der Frontalkollisionswarnung; bei dieser Simulation wurde eine realistische Parametrierung für die Fahrerreaktionen verwendet [GZL⁺09].

- Einparkassistentz
- Navigationssysteme
- Spurhalteassistentz und Spurverlassenswarnung
- Fußgängerschutz
- Adaptives Kurvenlicht
- Verkehrszeichenerkennung

Der Fokus dieser Arbeit liegt dabei auf vorwärtsgerichteten Kollisionsvermeidungssystemen, mit denen in einem Großteil der Unfallsituationen Menschenleben gerettet werden können. Hierfür gibt es die folgenden Unterstützungsmöglichkeiten:

- Reaktionsunterstützung: Akustische, haptische oder optische Warnung
- Ausweich- und Bremsunterstützung der Fahrerhandlung
- Autonome Notbremsung oder Ausweichen

Die Funktionen sind nach Eingriffsstärke aufgezählt. Durch Reaktionsunterstützung wird der Fahrer lediglich auf die Gefahr aufmerksam gemacht, wodurch eine frühere Reaktion erzielt werden soll. Dabei ist die Ausprägung der Warnung ein wichtiger Faktor. Lee et al. verglichen graduelle Warnungen, deren Intensität mit dem Eskalationsgrad der Situation korrespondiert, mit Warnungen konstanter Intensität [LHH04]. Die graduellen Warnungen erzielten nicht nur bestmögliche Fahrerreaktionen, sondern gleichzeitig mehr Vertrauen durch erhöhtes Systemverständnis des Fahrers.

Die Ausweich- oder Bremsunterstützung geht einen Schritt weiter: Zwar bleibt die Reaktion und damit auch die Verantwortung vollständig dem Fahrer überlassen, er wird aber bei seiner selbstinitiierten Handlung unterstützt.

In der letzten Ausbaustufe steuert das Fahrzeug selbstständig - z.B. indem es bis in den Stillstand bremst. Diese Systeme sind zwar die effektivsten Möglichkeiten zur Unfallvermeidung, sie stellen bei fehlerhaftem Eingriff jedoch auch ein großes Risiko dar. Eine autonome Notbremsung kann z.B. selbst einen Unfall hervorrufen, wenn ein Folgefahrzeug dicht auffährt und nicht schnell genug reagiert [HWB09]. Aus Gründen der Produkthaftung werden autonom eingreifende Systeme deshalb nur vorsichtig eingesetzt.

2.3 Fahrermodelle

Fahrerassistenzsysteme benötigen Beschreibungen des Fahrverhaltens, um z.B. eine Warnfunktion ideal zu konzipieren. Abb. 2.3 gibt einen Überblick über bestehende Modellierungsansätze. Im Bereich Fahrdynamik wird eine Kenntnis über das typische Steuerverhalten wie die Ausführungs geschwindigkeit oder tolerierte Beschleunigungen benötigt (*control theoretic*). Psychologische Modelle der kognitiven Belastung, Aufmerksamkeit und Wahrnehmung hingegen werden für Informationssysteme oder Unfallforschung eingesetzt (*human factors*). Assistenzsysteme wiederum benötigen eine frühe Erkennung zukünftiger Gefahrensituationen, wozu datengetriebene Verfahren für die Interpretation von Signalen eingesetzt werden (*stochastic*).

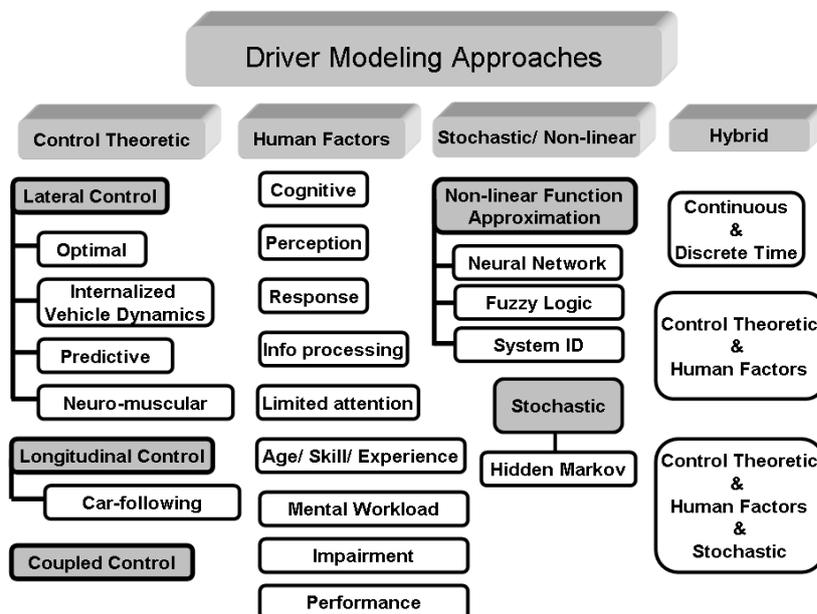


Abbildung 2.3: Ansätze zur Fahrermodellierung nach Boyraz et al. für verschiedene Anwendungsgebiete [BSH09]. Für eine Kollisionswarnfunktion müssen Faktoren aus allen Bereichen berücksichtigt werden.

Diese Modelle sind für unterschiedliche Einsatzgebiete spezialisiert, weisen aber wichtige Gemeinsamkeiten auf. Im Folgenden wird anhand exemplarischer Arbeiten von der kon-

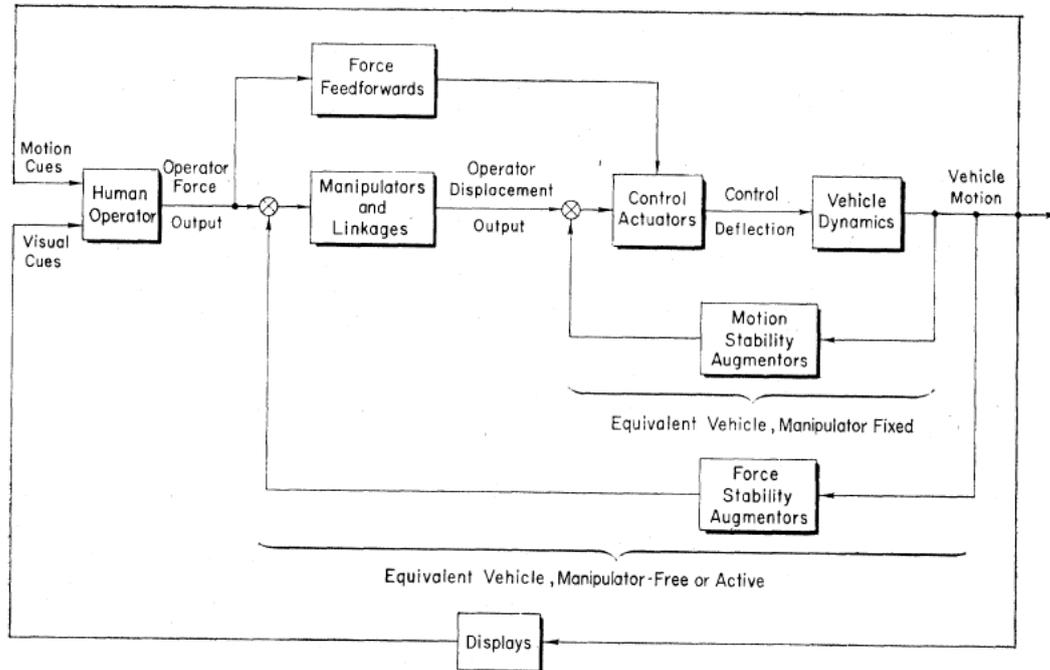


Abbildung 2.4: Modellierung der Interaktion Fahrer - Fahrzeug als Reglersystem. Im inneren Regelkreis kann das Modul *Motion Stability Augmentor* die Stabilisierung des Fahrzeugs übernehmen; das Modul *Force Stability Augmentors* dient der Umsetzung größerer Steuerimpulse; Visuelle Signale, evtl. verbessert durch das Modul *Displays*, runden zusammen mit haptischen Signalen den Reglerkreis ab [MW69].

trolltheoretischen Modellierung der Signalverläufe zur Fahrerintentionserkennung überleitet, wodurch das Zusammenspiel der verschiedenen Modelle im hybriden Bereich deutlich wird.

2.4 Fahrer als Reglersysteme

Schon 1969 modellierten McRuer und Weir [MW69] das menschliche Steuerverhalten für Flugzeuge oder Fahrzeuge in Form eines Reglerkreises. Dieses System enthält mehrere geschlossene Regelschleifen für die Fahrzeugstabilisierung, Steuerung der Bewegung und menschliche Wahrnehmung (Abb. 2.4). Der Fahrer kann durch *Stability Augmentor*-Module auf Stabilisierungs- und Führungsebene sowie durch verbesserte Darstellungen unterstützt werden. Zum Verständnis des Steuerverhaltens werden vier zentrale Einflussgrößen herausgearbeitet:

- Systemgrößen bezeichnen die Eingangssignale des Fahrzeugsystems sowie alle Größen außerhalb des menschlichen Fahrers wie Fahrzeugeigenschaften, aktuelle Geschwindigkeit, Objektabstände, Straßengeometrie.
- Umwelteigenschaften sind indirekt mit der Fahraufgabe verbundene Umstände wie Beleuchtung oder Temperatur.

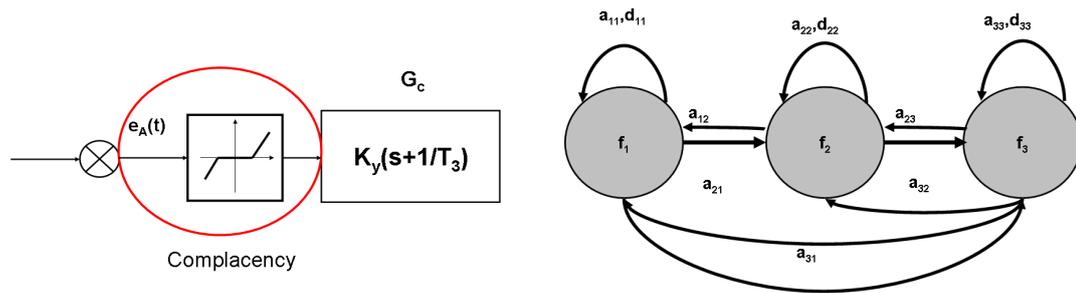


Abbildung 2.5: Modellierung des menschlichen Steuerverhaltens durch zwei Komponenten. Links: Der Lateralregler drückt Diskontinuitäten in der Querführung aus, die von Fahrern verzögert wahrgenommen werden. Rechts: Die Längsführung wird durch ein einfaches Hidden Markov Model dargestellt [BSH09].

- Eigenschaften des menschlichen Fahrers wie Training, Motivation, Müdigkeit, Unaufmerksamkeit
- Prozedurale Variablen beschreiben die aktuelle Fahraufgabe und die Art der Instruktionen; sie werden vor allem verwendet, um die Ausgangssituation in Studien zu modellieren.

Die Autoren untersuchten schließlich den Zusammenhang verschiedenartiger visueller Reize auf das Steuerverhalten. Dabei fanden sie einen proportionalen Zusammenhang zwischen dem Lenkradwinkel und der wahrgenommenen lateralen Fahrzeugbewegung. Dieser Zusammenhang lässt zum Beispiel auf Sinus-förmige Lenkradwinkelverläufe schließen, woraus die Abtastrate der Daten geschätzt werden kann (Kapitel 6.5). Des weiteren motiviert die Studie eine systematischen Analyse der beschriebenen Fahrereinflussfaktoren, um das Steuerverhalten besser präzisieren zu können.

Auf den Arbeiten von McRuer und Weir basierend wurden von Boyraz et al. das Steuerverhalten der Fahrer zur Längs- und Querführung des Fahrzeugs durch ein konkretes Modell formalisiert [BSH09]: Für die laterale Fahrzeugführung (Lenken) wurde der Reglerkreis um ein Hystereseglied erweitert, welches eine Verzögerung zwischen der Fahrzeugbewegung und dessen Wahrnehmung durch den Fahrer darstellt (Abb. 2.5). Diese Verzögerung wird durch ein Plateau um den mittleren Ruhezustand (complacency) herum modelliert; erst nach Verlassen dieses Plateaus erfolgen Korrekturbewegungen. Die Längssteuerung wird durch ein Hidden Markov Model mit den diskreten Zuständen {Geschwindigkeit halten, Bremsen, Beschleunigen} modelliert.

Stanley et al [SKSM05] trainierten für einen Vorversuch ein neuronales Netzwerk zur Simulation eines Rennfahrers. Der Versuch fand vollständig im Simulator statt, und als sensorische Eingaben wurden die Distanzen zum Fahrbahnrand verwendet. Das mittels *NeuroEvolution of Augmenting Technologies (NEAT)* erstellte Fahrermodell erzielte mittlere Rundenzeiten - verglichen mit manuell programmierten Fahrermodellen (Abb. 2.6). Dieses Ergebnis ist ein Indiz dafür, dass das menschliche Fahrverhalten prinzipiell durch einen Computer nachgebildet werden kann; für hohe Transparenz und eine Interpretation des gelernten Modells werden aber einfachere Modelle bevorzugt.

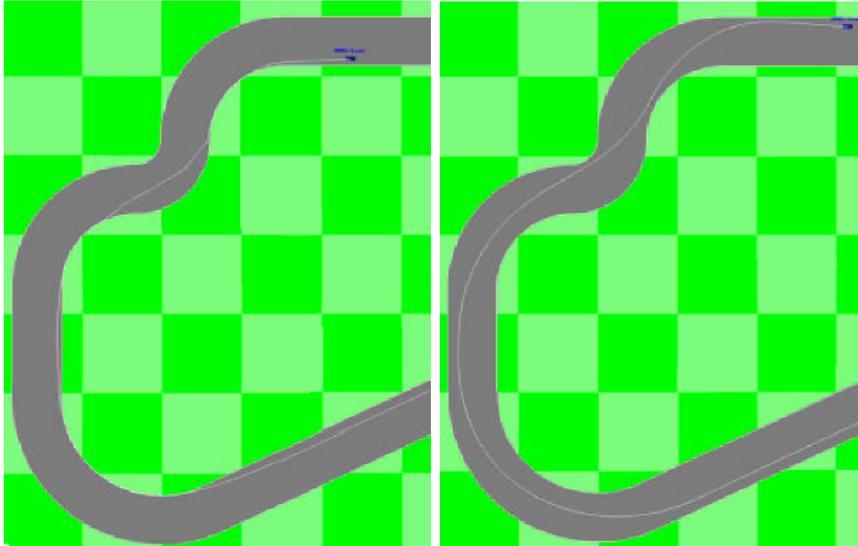


Abbildung 2.6: Simulation der Fahrt durch einen Parcours, der Wagen wird durch ein neuronales Netzwerk gesteuert und erhält als Eingabe die Luftliniendistanz zum Fahrbahnrand in verschiedene Richtungen. Links: Zu Beginn des Trainings wird vor allem die gefahrene Distanz minimiert, nicht jedoch die Rundenzeit. Rechts: Nach etwa 400 Generationen fährt das neuronale Netzwerk entlang der Ideallinie [SKSM05].

2.5 Fahrstile

Aus den zuvor vorgestellten Arbeiten [BSH09] wurde abgeleitet, dass gute Fahrer eine schnelle Regelschleife der Querführung erreichen, indem sie zur Wahrnehmung der Fahrzeugbewegung primär haptische Reize verwenden; die schnellere Wahrnehmung resultiert in zeitnahen Korrekturbewegungen, was schließlich eine gleichmäßige und komfortable Fahrt ermöglicht. Schlechtere Fahrer verwenden eher visuelle Rückkopplung, wodurch sie Fahrzeugbewegungen verzögert wahrnehmen und entsprechend später und unstetig lenken. Dieses Ergebnis verdeutlicht die individuellen Fahrweisen und zeigt, dass eine gute Modellierung verschiedene Fahrstile abdecken muss.

Entsprechende Untersuchungen zeigten, dass der Fahrstil, wie er vom Fahrer selbst oder einem Beobachter empfunden wird, mit den Beschleunigungswerten beim Bremsen oder Lenken zusammenhängt [DFF07, SF08]. Für diese Beschleunigungswerte und für die Reaktionszeit ermittelten Georgi et al. Standardparameter für verschiedene Fahrertypen [GZL⁺09]:

- Best driver: Reaktionszeit nach akustischer Warnung: 0.7s; nach haptischer Warnung: 0.4s, Bremskraft: 100%
- Realistic driver: Reaktionszeit nach akustischer Warnung: 1s; nach haptischer Warnung: 0.7s, Bremskraft: 80%
- Lethargic driver: Reaktionszeit nach akustischer Warnung: 2s; nach haptischer Warnung: 1.5s, Bremskraft: 60%

Der Großteil der Population wird durch die Parameter *Realistic driver* abgedeckt. In Kapitel 3.2 wird mit diesem Modell die verwendete Warnfunktion und das unumgängliche Warndilemma beschrieben.

2.6 Fahrdynamik und Kursprädiktion

Viele frühe Fahrerassistenzsysteme wie ACC wurden als Komfortfunktion entwickelt; ihr Einsatz ist vor allem in relativ gleichmäßigen Fahrsituationen wie Autobahnfahrten geplant. Der zukünftige Fahrzeugkurs wird geschätzt und zur Spurzuordnung der anderen Fahrzeuge eingesetzt, damit eine stabile Geschwindigkeitsregelung bezüglich des Vorderfahrzeugs erfolgen kann. Diese Kursprädiktion kommt auch in anderen Funktionen wie adaptivem Kurvenlicht oder Navigation zum Einsatz.

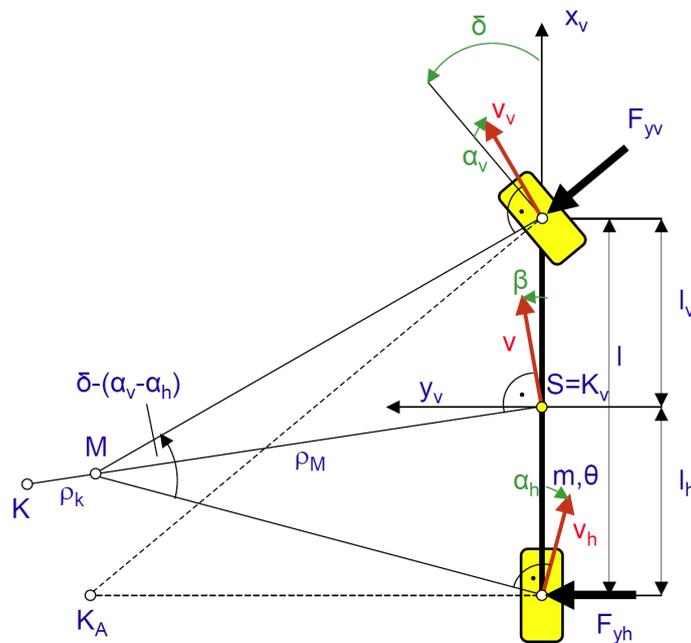


Abbildung 2.7: Lineares Einspurmodell zur Berechnung der Fahrzeugbewegung. Bei schneller Fahrt wird der Drehpunkt K_A durch die Fliehkraft auf den Drehpunkt K verschoben: Das Fahrzeug untersteuert [DGW07].

Fahrdynamische Extremsituationen wie eine Umverteilung der Radlasten oder Wanken kann für solche Fahrerassistenzfunktionen meist vernachlässigt werden [DGW07]. Das lineare Einspurmodell bietet dann eine geeignete, einfache Modellierung der Fahrzeugbewegung (Abb. 2.7): Die linken und rechten Reifen werden je Achse in der Fahrzeugmitte zusammengefasst, und die gesamte Fahrzeugmasse im Schwerpunkt angenommen. Bei langsamer Kreisfahrt liegt der Mittelpunkt K_A des Kreises genau in der Verlängerung der Hinterachse. Bei schnellerer Fahrt bewirkt die Fliehkraft, dass die Drehachse um den Schwimmwinkel β verschoben wird - es ergibt sich der neue Drehpunkt K . Diese Verschiebung wird je nach Richtung als *Untersteuern* oder *Übersteuern* bezeichnet [Mit90].

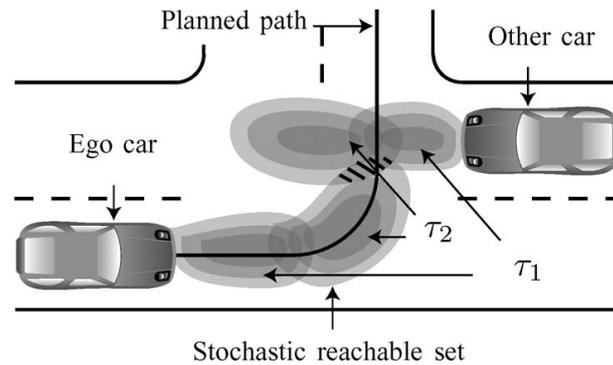


Abbildung 2.8: Modellierung der aktuellen und zukünftigen Aufenthaltswahrscheinlichkeiten beteiligter Fahrzeuge durch *Stochastic reachable sets*. Diese erlauben die Berücksichtigung von Unsicherheiten im Verhalten und das Testen einer geplanten Trajektorie auf Kollisionsfreiheit [ASB09].

Mit dem linearen Einspurmodell kann der Fahrzeugkurs präzisiert werden (Kapitel 4.4); hierfür werden zusätzlich die Geschwindigkeit, Beschleunigung, Gierrate und Schwimmwinkel benötigt. Weil diese Signale meist verrauscht sind, wird die Prädiktion häufig durch die Verwendung des Kalman Filters [JJG02] sowie des Extended bzw. Unscented Kalman Filters [BWB09, Cav07] verbessert. Hu et al. [HXX⁺04] verwenden Kohonen Self-organizing Neural Networks, um Fahrzeugkurse an einer Kreuzung unter Verwendung von Videodaten einer außenstehenden Kamera zu präzisieren und anschließend auf eine bestehende Kollision zu prüfen.

Spezialisiert auf autonomes Fahren wurden von Althoff et al. [ASB09] *Stochastic reachable sets* vorgestellt: Für gewählte diskrete Zeitpunkte wird die Aufenthaltswahrscheinlichkeit der beteiligten Fahrzeuge modelliert (Abb. 2.8). Diese werden für zukünftige Zeitpunkte als Markov-Kette präzisiert; die Übergangswahrscheinlichkeiten wurden von Experten modelliert und orientieren sich an den typischen Manöververläufen, erlaubten Geschwindigkeitsbereichen sowie dem Folgen der Straßengeometrie.

Für Fahrerassistenzsysteme sind viele Erkenntnisse des autonomen Fahrens jedoch nur sehr eingeschränkt verwendbar: Hier liegt per definitionem das Prinzip zugrunde, dass das Egofahrzeug seinen Fahrkurs beliebig planen könnte, um Kollisionswahrscheinlichkeiten gegen 0 anzunähern; bei Fahrerassistenzsystemen muss umgekehrt die Planung des Egofahrers erkannt werden und auf Kollisionsfreiheit überprüft werden.

Polychronopoulos et al. [PTAA07] konnten durch Modellierung der Egofahrzeugbewegung anhand Kalman-Filtern, der kollektiven Bewegung anderer Verkehrsteilnehmer sowie Informationen über die Straßengeometrie eine zuverlässige Prädiktion des zukünftigen Egoekurses gestalten. Durch geschickte Fusionierung der Informationen wurden für einen Prädiktionszeitraum von 3-4s mittlere laterale Fehler unterhalb 0.5m erreicht; lediglich während schneller Manöver stieg der Fehler auf bis 1m an. Während diese und ähnliche Arbeiten die Prädiktion des Fahrzeugkurses grundsätzlich motivieren, wird auch deutlich, dass besonders in dynamischen Fahrsituationen der Prädiktionsfehler ansteigt. Für Kollisionswarnsysteme müssen solche Situationen also besonders untersucht werden und anhand der Fahrerabsicht verbessert werden.

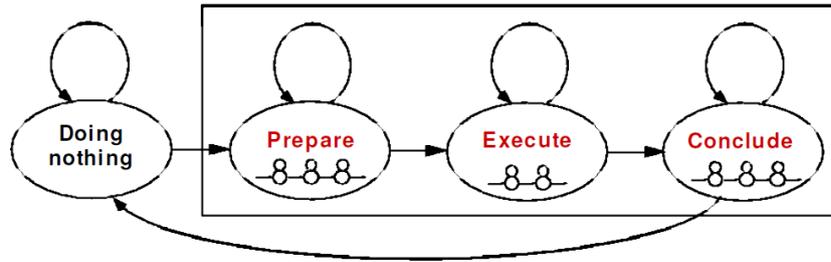


Abbildung 2.9: Modellierung von Fahrmanövern durch Segmentierung in die drei intuitiven Phasen; dieser Ansatz zeigt, dass der Prozess des Steuerns eines Fahrzeuges durch einen Markovprozess modellierbar ist; folglich werden Hidden Markov Modelle zur Erkennung der Fahrmanöver auf Simulatordaten untersucht [PL99].

2.7 Fahrerintentionserkennung

Die Fahrerintention beschreibt den vom Fahrer geplanten zukünftige Vorgang, in Form diskreter Fahrmanöver oder des zukünftigen Fahrzeugkurses. Die Erkennung der Fahrmanöver wie $\Psi \in \{\text{Abbiegen, Spur wechseln, Kreisverkehr, ...}\}$ wurde bereits 1999 von Pentland und Liu [PL99] beschrieben. Dabei werden Fahrmanöver zunächst in Phasen unterschiedlicher Länge segmentiert. Diese Phasen entsprechen dem Zustand eines Markovprozesses, sie werden entsprechend durch ein *Hidden Markov Modell* abgebildet und erkannt (Abb. 2.9).

Weil ausschließlich instruierte Fahrmanöver in einer Simulatorumgebung analysiert wurden, ist die direkte Übertragbarkeit der Ergebnisse auf reale Szenarien nicht möglich. Zudem wurden Fahrmanöver nicht prädiziert, sondern konnten erst mit zeitlicher Verzögerung von ca. 1.5s nach Manöverbeginn zu 93% erkannt werden. Dennoch liefert die Arbeit wichtige Hinweise auf Charakteristiken von Fahrdaten wie die Markov-Eigenschaft und variierende zeitliche Skalierung der Manöversequenzen.

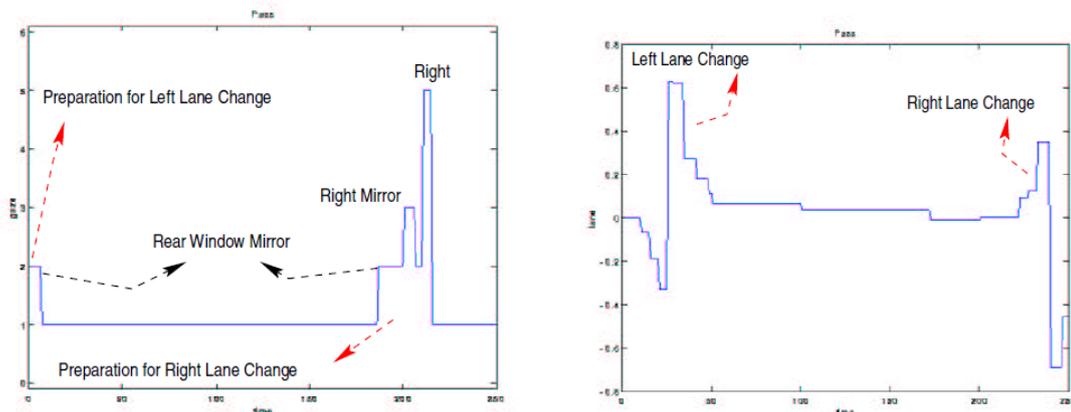


Abbildung 2.10: Zusammenhang zwischen Blickbewegungen (links) und Beginn des Spurwechselmanövers (rechts). Durch die Hinzunahme des aufwändigen Blickbewegungssensors konnten Spurwechsel sogar vor der Fahrzeugbewegung detektiert werden [OP00].

In einer Folgearbeit von Oliver und Pentland [OP00] wurden Blickbewegungen als Indikatoren der Fahrerabsicht auf realen Fahrdaten angewandt. Abbildung 2.10 zeigt den

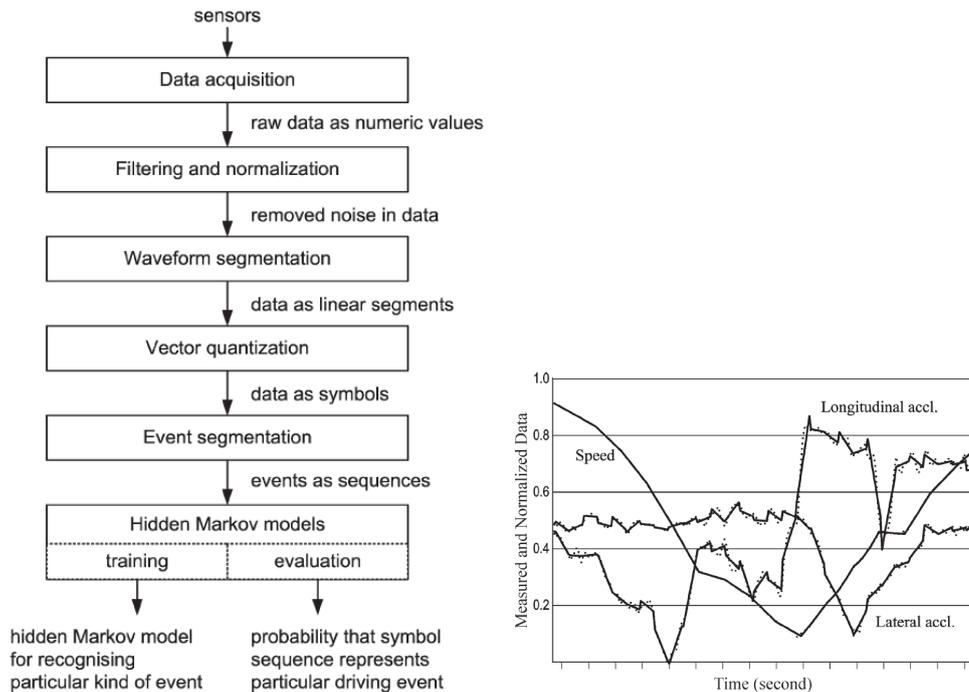


Abbildung 2.11: Links: Schema und Informationsfluss von den Sensordaten eines Fahrzeugs bis hin zur Klassifikation des Fahrmanövers durch Hidden Markov Modelle. Rechts: Kompression hoch abgetasteter Signale (Punkte) durch die Waveform-Segmentation (durchgezogene Linien). Hierbei werden die Signaldaten fensterweise auf die statistischen Merkmale Mittelwert und lokale Steigung reduziert [Mit05].

systematischen Zusammenhang vorhergehenden Blickbewegungen (z.B. in den Rückspiegel) mit einem folgenden Spurwechselmanöver. Mit dieser zusätzlichen Sensorik konnten Spurwechsel 0.67s vor einer Fahrzeugbewegung festgestellt werden. Hohe Schwankungen von 0 - 100% Erkennungsrate je nach Manöver und Sensorik sowie die aufwändige Blickbewegungsmessung führen jedoch dazu, dass diese Methode besonders für Kollisionsvermeidung in Serienfahrzeugen ungeeignet sind.

Ähnlich untersuchte Mitrovic [Mit05] die Erkennung von Fahrmanövern auf realen Messdaten bei natürlicher Fahrweise ohne Instruktoranweisungen. Zwar publiziert der Autor eine vielversprechende Erkennungsrate von 98%, jedoch wurden auch hier abgeschlossene Fahrmanöver klassifiziert und keine Resultate für die Prädiktion genannt. Als wichtiges Resultat wird jedoch ein Schema zur Datenverarbeitung vorgestellt, welches auch in dieser Arbeit angewandt wurde (Abb. 2.11). Für die Datenvorverarbeitung wird ein Butterworth-Filter mit einer Frequenz von 2Hz verwendet, wodurch Rückschlüsse auf die Abtastrate gemacht wurden (Kapitel 6.5).

Torkkola et al [TVL06] ziehen Parallelen zwischen Fahrsequenzen und der Verarbeitung akustischer Schwingungen zur Spracherkennung. Hierbei bilden *Phoneme* die kleinste sinnvolle Einheit; sie werden zur Sprachverarbeitung herbeigezogen, um die Daten schon früh zu komprimieren, ohne Informationen zu verlieren. In Analogie wurden kurze Fahrsequenzen als *Drivemes* definiert und ähnlich verwendet. Diese abstrakte Repräsentation entstand aus überwachtem Lernen mit HMMs und dem anschließend unüberwachten Clustern und Zusammenziehen von Zuständen.

Die Verwendung von Hidden Markov Modellen zur Erkennung der Fahrerintention ist also ein bewährtes Verfahren, welches in vielen Arbeiten eingesetzt wird. Dies bestätigt sich auch in aktuellen Publikationen, in denen meist das Basisschema mit Modifikationen verfolgt wird [BED08, MDPB09].

Neben HMMs gibt es auch andere Ansätze: Ein expertenkodiertes Fuzzy-Logic System wird in [BSF08] zur Erkennung von Überholvorgängen verwendet. Hang et al. [HKM⁺94] verwenden z.B. ein expertenkodiertes Bayessches Netzwerk, um mit Hilfe einer fest montierten Kamera den Verkehrsfluss und Spurwechselmanöver an einer Autobahn zu analysieren. In anderen Arbeiten kommen Support Vector Machines [MS05] oder Random Forest [TVL04] zur Fahrmanövererkennung zum Einsatz. Weil Random Forest besonders durch seine Stabilität in hochdimensionalen Merkmalsräumen hervorsteicht, stellte er auch im Rahmen dieser Arbeit die erste Wahl dar.

2.8 Reaktionszeit

Die Kenntnis des Fahrerszustandes und seiner Aufmerksamkeit lässt Rückschlüsse auf die Reaktionsfähigkeit zu. In diesem Kontext sind die Reaktionszeit, maximale Brems- und Lenkkraft wichtige Größen, die ein optimales Timing einer Kollisionswarnfunktion ermöglichen (Kapitel 3.3).

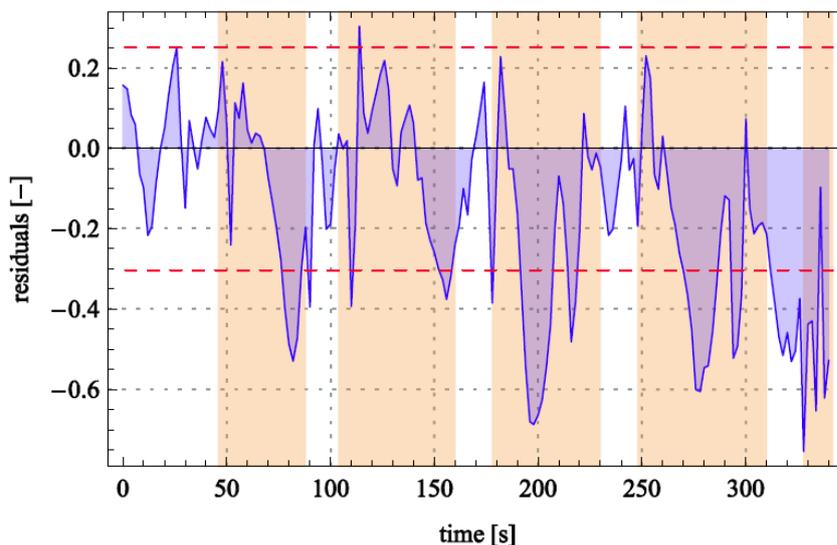


Abbildung 2.12: Differenz zwischen realer und prädizierter Fahrerreaktion in Situationen, in welchen der Fahrer aufmerksam oder abgelenkt ist: In abgelenkten Zeiträumen (farblich hinterlegt) wächst die Differenz deutlich an, weil der Fahrer nicht wie erwartet handelt [EFT⁺10].

Ersal et al. publizierten eine interessante Methode, um den Ablenkungsgrad des Fahrers zu schätzen [EFT⁺10]. Zunächst wurde für nicht abgelenkten Fall mit Hilfe eines *Radial Basis Function Neural Networks* ein Modell trainiert, welches die Fahrerreaktion prädiziert. Zur Schätzung des Ablenkungsgrades wird die Prämisse verfolgt, dass sich die Fahrerreaktion in abgelenkten Situationen von der Prädiktion unterscheidet (Abb. 2.12). Anhand dieser

Differenz schätzte also eine *Support Vector Machine* den Ablenkungsgrad. In einer Studie mit $N = 16$ Probanden konnten mit diesem Prinzip in 50 - 100% der Zeitpunkte der Ablenkungsgrad korrekt geschätzt werden. In einem verwandten Ansatz versuchen Choi et al. [CKK⁺07], neben dem Ablenkungsgrad auch den Fahrer selbst und dessen Handlung zu identifizieren, wofür jeweils Hidden Markov Modelle zum Einsatz kommen.

Aufgrund hoher Klassifikationsschwankungen sowie der Einschränkung auf bestimmte Situationen wie Folgefahrten sind solche Ansätze jedoch nur schwer in die Praxis übertragbar. Die unterschiedlichen Ausprägungen des Ablenkungsgrades von Fahrern sowie das grundlegende Prinzip der Fahrerzustandsschätzung sind dennoch wichtige Erkenntnisse für Assistenzsysteme.

2.9 Kritikalitätsempfinden

Van der Horst und Hogema [vH95] erforschten 1995 die Wichtigkeit des Merkmals *Time to collision (TTC)*, welches die verbleibende Zeit bis zur Kollision schätzt (Kapitel 5.6.3). Probanden wurden instruiert, während der Geradeausfahrt auf ein stehendes Hindernis im subjektiv letzten Moment zu bremsen; dieser Zeitpunkt hing systematisch vom TTC ab. Im weiteren Verlauf wurden Kollisionswarnsysteme mit verschiedenen Strategien getestet, wobei eine Warnung abhängig vom TTC wie erwartet zur größten Akzeptanz führte. Die Autoren empfehlen eine Warnung bei Erreichen der Bedingung Zeitpunkt $TTC = 4s$.

Kiefer et al. [KLF05] publizierten 2005 ein erweitertes Modell für die empfundene Kritikalität; hierzu wurde das Merkmal TTC durch die Berücksichtigung der Beschleunigung zu der *Enhanced time to collision (ETTC)* erweitert. In einer ähnlichen Studie wurden Probanden instruiert, während der Geradeausfahrt auf ein langsames Vorderfahrzeug Bremsungen und Ausweichvorgänge im Komfort- und Notbereich durchzuführen. Dabei konnte ein systematischer Zusammenhang zwischen dem Ausweichzeitpunkt, der Fahrzeuggeschwindigkeit und der inversen ETTC beobachtet werden. Mit logistischer Regression wurde ein Modell erstellt, welches die empfundene Kritikalität schätzt und das Fahrerverhalten als Komfort- oder Notreaktion klassifiziert.

Li und Milgram fanden eine starke Korrelation zwischen der menschlichen Wahrnehmung der Kritikalität und der Ausdehnungsgeschwindigkeit eines sich annähernden Objektes auf der Netzhaut, bzw. mit dem optischen Fluss [LM04]; dieses Ergebnis stimmt mit der Bedeutung der inversen TTC überein, die mathematisch ähnlich berechnet wird.

In vergleichbaren Szenarios suchten Schmidt et al. nach einer quantitativen Möglichkeit, eine differenzierte Schätzung der subjektiven Kritikalität zu ermitteln [SKB09]. Durch Knopfdruck konnte jeweils eine Eskalation in die nächste höhere subjektive Gefahrenstufe signalisiert werden, gebremst wurde im empfundenen letzten Moment. Anhand der Ergebnisse dieser Studie kann parallel zu den Erkenntnissen von Kiefer et al. eine grobe Kritikalitätsschätzung geschehen (Abb. 2.13).

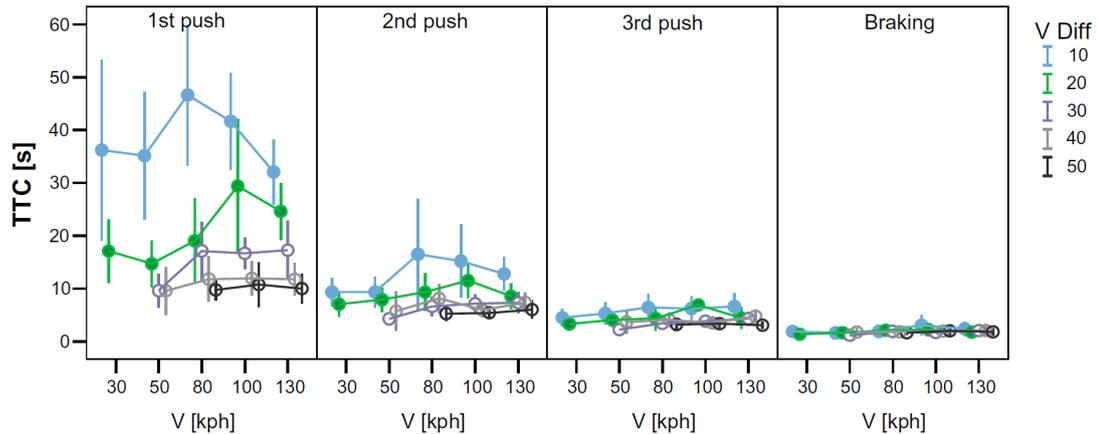


Abbildung 2.13: Subjektive Kritikalität, in einer Auffahrsituation ermittelt. Während das Erreichen der ersten Eskalationsstufe primär von der Distanz des Objektes abhängt (Distanz = $TTC \cdot \text{Differenzgeschwindigkeit}$), werden höhere Eskalationsstufen fast ausschließlich unter Berücksichtigung des TTCs bewertet [SKB09].

Diese Erkenntnisse zur subjektiven Kritikalität sind wichtig, um die Akzeptanz einer Warnfunktion zu untersuchen. Für deren Timing ist jedoch nur der tatsächliche Bremsweg und die Reaktionszeit des Fahrers von Belang. Diese objektive Kritikalität wird im Kontext des Warndilemmas (Kapitel 3.4) dargestellt.

Das New Car Assessment Programme (NCAP) liefert eine unabhängige Bewertung der Fahrzeugsicherheit. Dabei werden für verschiedene Kriterien Sicherheitspunkte vergeben, etwa den Insassenschutz, Fußgängerschutz durch Auslegung der Motorhaube, Funktionsweise der elektronischen Stabilitätskontrolle (ESP) oder neuerdings auch ein Kollisionswarnsystem. Eine für das letzte Kriterium zu erfüllendes Testszenario ist eine Frontalfahrt bei 70 km/h auf ein stehendes Hindernis, wobei 2.1 Sekunden vor der Kollision eine deutliche Warnung an den Fahrer ertönen muss [Har08].

2.10 Maschinelles Lernen

Maschinelles Lernen ist ein breites Forschungsgebiet der Informatik: Das Ziel ist es, Algorithmen universell so zu gestalten, dass sie nicht auf ein einzelnes Anwendungsgebiet beschränkt sind, sondern in verschiedenen Anwendungen mit ähnlicher Problemstruktur eingesetzt werden können.

Die bekanntesten Ausprägungen des maschinellen Lernens sind u.A. überwachtes, semi-überwachtes, schwach überwachtes und aktives Lernen: Ein zunächst generischer Algorithmus wird mit problemspezifischen Daten und manuellen Labels ausgeführt; sein Resultat ist ein Modell, welches mehr oder weniger gut das gewünschte Verhalten gelernt hat. Zu den häufigsten verwendeten Modellen gehören *Neuronale Netzwerke*, *Support Vector Machines (SVM)*, *Bayessche Netzwerke* und *Decision Trees*; zur Verbesserung des Trainings und der anschließenden Klassifikation kommen Ensemble-Verfahren wie *Simple-Bayes*, *Adaboost* oder *Random Forest* zum Einsatz [RN02, HTF01, ASP10].

Ein Fokus dieser Arbeit liegt auf der Analyse sequentieller Daten mit zeitlicher Bedeutung und Veränderung, wie sie in fast allen Echtzeit- und Assistenzsystemen vorliegen. Hierbei können die benötigten Informationen erst aus der Abfolge von Signalen gewonnen werden können, was spezielle Anforderungen an die Algorithmen stellt. Dietterich [Die02] gibt einen empfehlenswerten Überblick über diese Problemstruktur.

Hidden Markov Modelle (HMMs) sind aus der Sprachverarbeitung bekannte Modelle, mit denen solche sequentielle Daten klassischerweise analysiert werden [Rab89, MPPM07]. Das Verfahren zeichnet sich durch gut verständliche Modelle mit meistens nur wenigen Parametern aus, deren Zustände einen direkten Zusammenhang zur Bedeutung der Daten hat. Diskriminatives Training ermöglicht darüber hinaus auch die Klassifikation geeigneter Sequenzen [Kap98]. Als primäres Mittel der Wahl für diese Untersuchungen werden die Grundlagen zu HMMs in Kapitel 6.1 zusammengefasst und schließlich methodisch erweitert.

HMMs besitzen diskrete Zustände, wodurch flexible Übergänge möglich sind. *Lineare dynamische Systeme (LDS)* sind diesbezüglich restriktiver; sie ermöglichen nur lineare, dafür stetige und glatte Zustandsübergänge. Eine Kombination beider Verfahren sind *Switching linear dynamic systems (SLDS)*: Hierbei wird über einen diskreten Zustand $s(t) = i$ ausgewählt, welche Transitionsmatrix w_i die Fortpflanzung des kontinuierlichen Zustands beschreibt [ORBD08]. Somit können Vorgänge modelliert werden, die stückweise linear verlaufen. Für Anhaltvorgänge konnten Kumagai und Akamatsu mit SLDS bereits bessere Modellierungseigenschaften als mit HMMs nachweisen [KA04].

Coupled HMMs können voneinander losgelöste, jedoch zeitlich aneinander gekoppelte Prozesse modellieren. Brand et al. berichten gute Ergebnisse für die Erkennung von Gesten, die mit zwei Händen durchgeführt werden [BOP97]. *Hierarchical HMMs* modellieren verschachtelte Sequenzen, bei denen eine Handlung z.B. aus mehreren Unterhandlungen besteht [NPVB05]. Mit solchen komplexeren Strukturen kann das Gesamtsysteme zwar viele Zustände bei geringer Parameterzahl annehmen und den zu analysierenden Prozess eventuell besser abbilden, das Training wird jedoch entsprechend erschwert.

Für die Bestimmung der Relevanz einzelner Merkmale ist *Mutual Information* ein bewährtes Verfahren [Bat94]. Dieses Maß drückt aus, wieviel Gemeinsamkeiten zwischen einem Signal und der zu schätzenden Klasse existieren. Mutual Information ist damit für ein Merkmalsranking und -selektion geeignet und wurde auch im Kontext dieser Arbeit erfolgreich auf Fahrerassistenzdaten angewandt [HBZ10].

2.11 Zusammenfassung

Die Analyse von Verkehrsunfällen sowie das erklärte Ziel der EU-Kommision, die Zahl der Verkehrstoten von 2010 bis 2020 zu halbieren, zeigt deutlichen Handlungsbedarf im Bereich der Fahrerassistenz. Um zusätzlich zu Sicherheitsfunktionen weitere Unterstützungs-

möglichkeiten zu ermöglichen, wurden Modelle herangezogen, die das Fahrerverhalten auf motorischer und kognitiver Ebene abbilden.

Verfahren des Maschinellen Lernens ermöglichen es schließlich, anhand von Indikatorsignalen Vorhersagen über die zukünftige Fahrerhandlung zu treffen. Diese Arbeit setzt bestehende Untersuchungen dabei in zwei wesentlichen Punkten fort: Auf methodischer Ebene wurden Algorithmen des Maschinellen Lernens miteinander kombiniert, um die komplexen Datenstrukturen besser abzubilden. Auf anwendungsspezifischer Ebene wurde die Fahrerintentionserkennung auch auf die Kursprädiktion übertragen, um eine allgemeinere Modellierung und Prädiktion des Fahrerverhaltens zu erreichen.

3 Grundlagen der Fahrerassistenz für die Kollisionsvermeidung

Dieses Kapitel stellt das verwendete Kollisionsvermeidungssystem vor und diskutiert dabei dessen Funktion und Anforderungen. Es ist die Verbindung zwischen dem technischen Anwendungsbereich der Fahrerassistenz, der psychologischen Einschätzung der Kritikalität und der methodischen Klassifikation der Fahrerintention.

Anhand von Beispielszenen wird zunächst die potenzielle Kollisionsgefahr und deren Unvorhersehbarkeit demonstriert. Daraus ergibt sich der Bedarf für die Schätzung der *Fahrerintention* ermittelt. Um parallel die Kritikalität der Szene zu erfassen, wird ein *Fahrermodell* anhand seiner Reaktionszeit und Manövrierfähigkeit parametrisiert. In Kombination mit der *Vermeidungsbeschleunigung* sind dies zentrale Größen, um zu bewerten, ob und zu welchem Zeitpunkt eine drohende Kollision verhinderbar ist. Die abgeleitete Warnstrategie führt schließlich zum unvermeidbaren *Warndilemma*, welches die Wichtigkeit der Fahrerintentionsschätzung abschließend hervorhebt.

3.1 Beispielszenen

Die Abbildungen 3.1, 3.2 und 3.3 zeigen drei repräsentative Fahrerszenen mit einem stehenden Objekt, aus Sicherheitsgründen auf einer Teststrecke nachgestellt. In den kritischen Szenen wurde ein Reflektor (Tripelspiegel) verwendet, montiert auf einem Schaumstoffwürfel, wie es für Fahrzeugtests üblich ist. Der Reflektor weist zudem eine stärkere Rückstrahlcharakteristik als ein reales Fahrzeug auf und ist damit für Tests, die die Sensorik nicht betreffen, ideal.

In den Szenen findet jeweils eine Annäherung statt und der Fahrer bremst spät bzw. weicht im letzten Moment aus. In der dritten Szene kollidiert er mit dem Hindernis: Eine Warnung etwa ab $f_{TTC} = 2s$ hätte den Fahrer hier dazu gebracht, rechtzeitig zu bremsen (Kapitel 3.3).

Trotz unterschiedlicher Ausgänge verlaufen die Szenen lange Zeit identisch: Die dargestellten Signale, welche primär entscheidend sind für die Fahrzeugbewegung, unterscheiden sich in den ersten 2 Sekunden kaum. Es ist lediglich eine minimale Bewegung des Lenkradwinkels ($f_{WheelAngleDt}$) sowie des Objektversatzes ($f_{ObjectDy}$) bemerkbar, die aber nicht

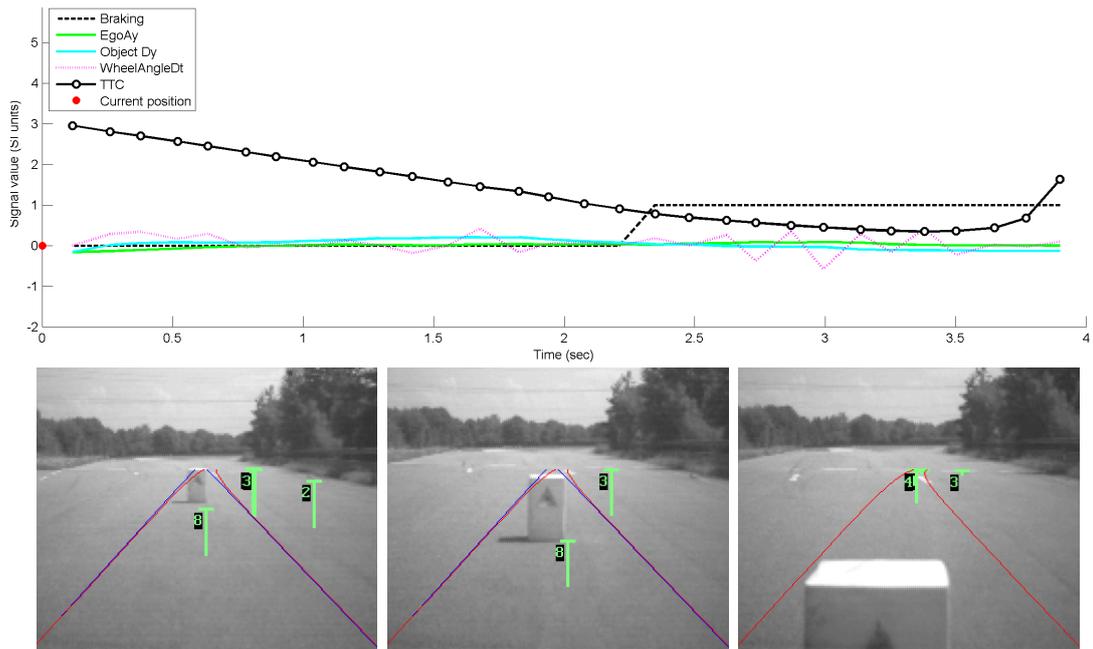


Abbildung 3.1: Bremsvorgang: Der Fahrer bremst zum Zeitpunkt 2.3s, bei dem die geschätzte Kollisionszeit $f_{TTC} = 0.8s$ beträgt. Da die Fahrzeuggeschwindigkeit abnimmt, steigt f_{TTC} ab diesem Moment wieder an.

unterscheidbar vom üblichen Rauschpegel dieser Signale ist und somit keine Rückschlüsse zulässt.

Bezüglich stationärer Objekte decken die gezeigten Szenen alle Möglichkeiten des Situationsausgangs ab, lediglich ein Ausweichvorgang nach rechts fehlt. Für bewegte Objekte, etwa andere Fahrzeuge oder Fußgänger, steigt jedoch die Szenendynamik. Damit existiert eine Vielzahl weiterer kritischer Szenen, zum Beispiel [KHB09]:

1. Ein langsames Fahrzeug schert auf der Autobahn plötzlich von der Nachbarspur auf die Spur des Egofahrzeugs aus
2. Auf einer Kreuzung fährt ein Fahrzeug ohne Vorfahrt von rechts in den Gefahrenbereich vor dem Egofahrzeug.
3. Das Vorderfahrzeug bremst unerwartet, zum Beispiel bei spontanem Abbiegen oder an einem Stauende.
4. Der Egofahrer versucht, trotz Gegenverkehr zu überholen.
5. Multiobjekt-Szenario: Das Fahrzeug vor dem Vorderfahrzeug bremst, so dass das Vorderfahrzeug auch bremsen muss, das Egofahrzeug also ebenfalls. Durch die Betrachtung beider Vorderfahrzeuge kann hier ein frühes Szenenverständnis aufgebaut werden und eventuelle Fahrfehler noch früher detektiert werden [HW09].

Die genannten Situationen führen meistens nicht zu einer Kollision, weil der Fahrer durch Ausweichen oder Bremsen deeskaliert. Auch das Fremdfahrzeug kann die Deeskalation

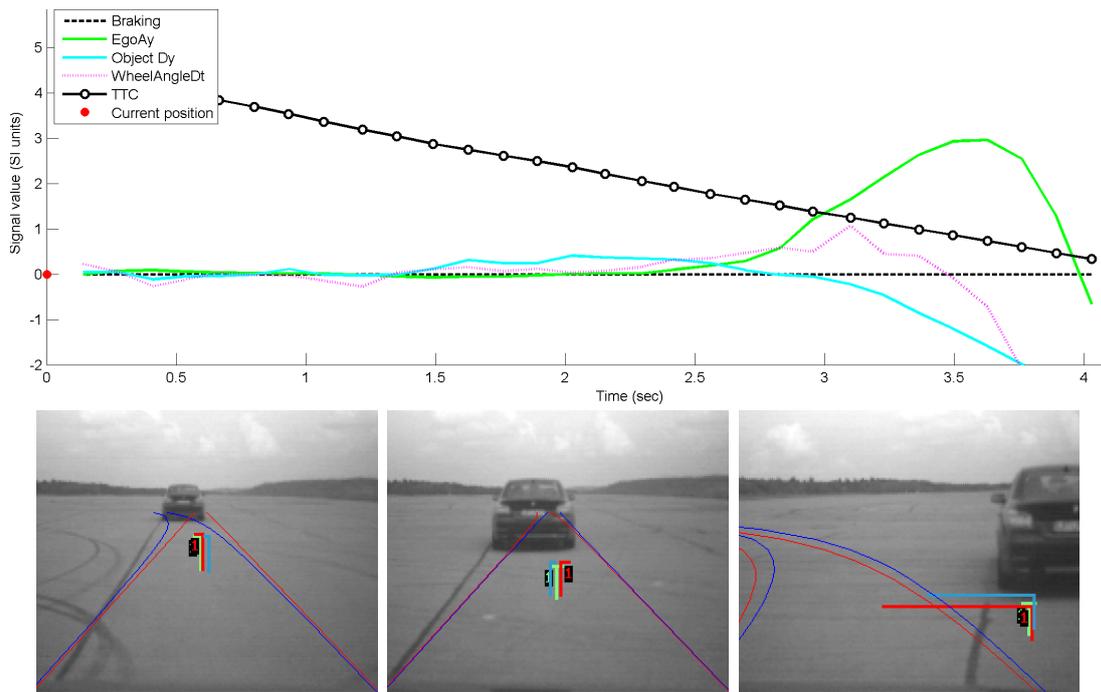


Abbildung 3.2: Ausweichen: Mit geringem Versatz nach rechts nähert sich das Egofahrzeug dem Hindernis, um schließlich links auszuweichen. Im Signalverlauf wird die Ausweichbewegung durch die Querbeschleunigung f_{EgoAy} und die Lenkbewegung $f_{WheelAngleDt}$ deutlich.

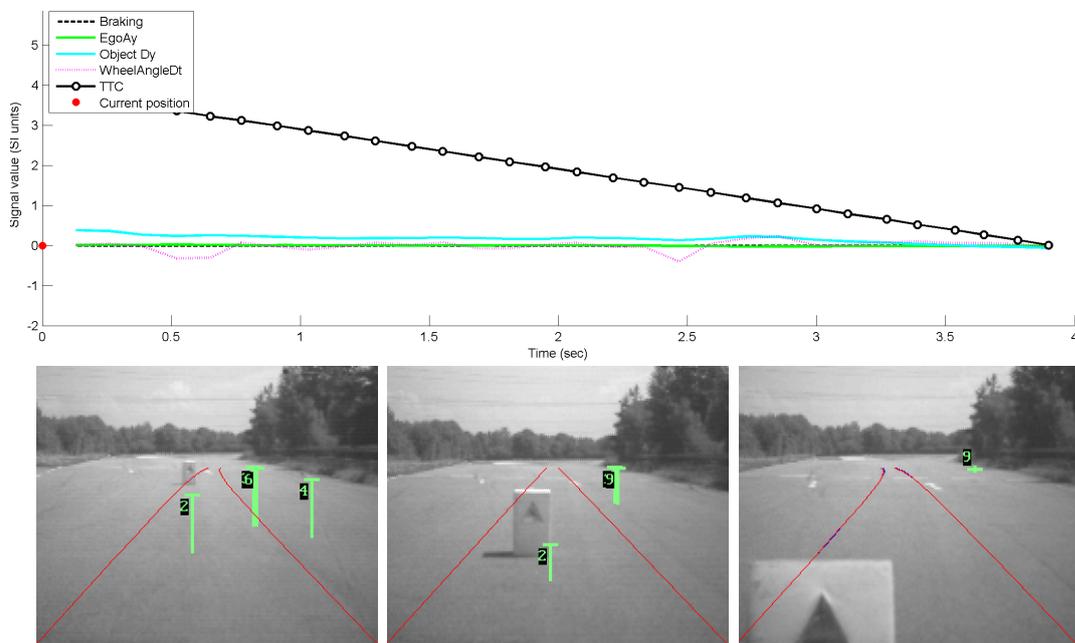


Abbildung 3.3: Kollision: Mit konstanter Geschwindigkeit wird hier eine Kollision mit dem künstlichen Hindernis erzeugt. Man erkennt im Signalverlauf, dass f_{TTC} kontinuierlich fällt und zum Kollisionszeitpunkt per definitionem den Wert 0 annimmt.

verursachen, indem es den Gefahrenbereich des Egofahrzeugs verlässt. Für eine unterstützende Funktion muss also die zukünftige Handlung des Egofahrers und der anderen relevanten Verkehrsteilnehmer geschätzt werden. In Kapitel 4 werden alle Möglichkeiten des Szenenausgangs formal modelliert und damit eine wichtige Grundlage für die Fahrerintentionsschätzung erarbeitet.

3.2 Fahrermodell

Das Fahrermodell beschreibt die Reaktionszeit sowie das Können des Fahrers anhand von Parametern, die für die Konzeption von Warnsystemen benötigt werden. Es definiert die maximale Fahrzeugdynamik, innerhalb welcher der Fahrer sich komfortabel fühlt. Dadurch kann der letztmögliche Zeitpunkt geschätzt werden, zu dem der Fahrer in einer kontrollierten Situation noch reagieren kann; umgekehrt bedeutet ein Übertreten dieser Werte auch das Verlassen des Komfortbereichs und führt damit zur Definition der Warnfunktion.

Zusätzlich beinhaltet das Modell die maximale Fahrzeugdynamik in Extremsituationen, also die maximale Beschleunigung während Brems- und Ausweichvorgängen. Sie werden benötigt, um den physikalisch letztmöglichen Moment der Reaktion zu bestimmen. Diese Parameter werden unabhängig vom Fahrer als maximal angenommen, da meist eine technische Brems- und Lenkunterstützung realisiert werden kann (Kapitel 2.2).

c_{React}	Reaktionszeit auf Warnung	$0.7s$
$a_{x,C}$	Max. Bremsverzögerung, komfortable Fahrt	$6m/s^2$
$a_{x,E}$	Max. Bremsverzögerung, Notfallsituation	$8m/s^2$
$a_{y,C}$	Max. Querb beschleunigung, komfortable Fahrt	$4m/s^2$
$a_{y,E}$	Max. Querb beschleunigung, Notfallsituation	$6m/s^2$

Tabelle 3.1: Verwendete Parameter des Fahrermodells

Die Reaktionszeit c_{React} ist im Allgemeinen schwer zu definieren; hier bezeichnet sie die Zeitspanne, bis der Fahrer auf eine Warnung reagiert. Diese Parameter hängen natürlich nicht nur individuell vom Fahrer ab, sondern auch von seinem aktuellen Zustand und Umwelteinflüssen (Kapitel 2.4). Zudem nehmen die tolerierten Beschleunigungen mit steigender Geschwindigkeit ab [SH82]. Zusammenfassend enthält Tabelle 3.1 die verwendeten Parameter, nach den Ergebnissen von [GZL⁺09], die auf Kollisionswarnsysteme fokussiert waren.

3.3 Warnstrategie

Um ein Kollisionswarnsystem zu konzipieren, muss verdeutlicht werden, dass für viele Systeme aus Vorsichtsmaßnahmen keine direkte Manipulation des Fahrgeschehens - etwa

durch autonomes Bremsen - gewünscht ist. Der Fahrer kann lediglich durch eine Warnung beeinflusst werden, eine frühere Reaktion auszulösen.

Hierfür wird die allgemein anerkannte und empirisch belegte Annahme verwendet, dass die meisten Fahrer in einer Gefahrensituation primär durch Bremsen reagieren [WN99]. Falls also eine drohende Kollision festgestellt wird, wird mit $t_{\text{Brake, E}}$ der letztmögliche Reaktionszeitpunkt für ein Notbremsmanöver bestimmt (Kapitel 5.6.5). Mit dem entsprechend früheren Warnzeitpunkt

$$t_{\text{Warn}} = t_{\text{Brake, E}} - c_{\text{React}} \quad (3.1)$$

wird der Fahrer so gewarnt, dass er laut Modellannahmen zum berechneten Reaktionszeitpunkt aktiv wird und die Bremsung vollführt (ähnlich [MWH07]).

3.4 Warndilemma

Die Warn- und Reaktionszeitpunkte der zuvor ermittelten Warnstrategie werden in Abb. 3.4 exemplarisch verdeutlicht. Der Warnzeitpunkt t_{Warn} liegt wegen der Reaktionszeit *vor* dem Zeitpunkt $t_{\text{Brake, C}}$ eines komfortablen Bremsvorgangs. Das bedeutet, dass im Kollisionsfall der Fahrer zu einem Zeitpunkt gewarnt werden muss, zu dem er im Normalfall noch gar nicht reagieren müsste.

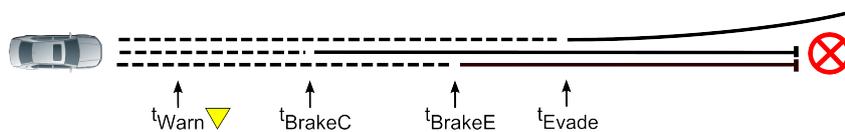


Abbildung 3.4: Verdeutlichung der Warn- und Reaktionszeitpunkte. $t_{\text{Brake, C}}$, $t_{\text{Brake, E}}$ und t_{Evade} sind die Reaktionszeitpunkte für komfortables Bremsen bzw. Notbremsung und Notausweichen; der Warnzeitpunkt t_{Warn} liegt vor diesen Zeitpunkten, woraus das Warndilemma entsteht.

Da das Fahrerassistenzsystem schon zum Warnzeitpunkt eine Entscheidung fällen muss, liegt ein *Warndilemma* vor: Wenn der Fahrer die Situation erkannt hat und lösen kann, ist eine Warnung unberechtigt. Man spricht dann von *Übernutzen*, was schnell zu Akzeptanzproblemen führen kann [Wil06, BTK05]. Warnt das System hingegen nicht, so stellt es im Fall einer wahren Kollision keine Hilfe dar und erzeugt *Unternutzen*. Zum Warnzeitpunkt ist also eine zuverlässige Erkennung der Fahrerintention (Kapitel 4) ausschlaggebend.

Der Tradeoff zwischen möglichem Über- und Unternutzen kann wie folgt abgeschätzt werden: In Abb. 3.5 wurde die Fahrzeuggeschwindigkeit systematisch variiert und die relativen Reaktions- und Warnzeitpunkte bezüglich eines stationären Objektes dargestellt (Kapitel 5.6, ähnlich [MWH07]). Man erkennt den Zusammenhang zwischen der Fahrzeuggeschwindigkeit und dem Warndilemma, z.B.:

- Bei 5 m/s kann ein aufmerksamer Fahrer zum Zeitpunkt $f_{\text{TTC}} = 0.4\text{ s}$ noch komfortabel bremsen. Ein unaufmerksamer Fahrer hätte bereits zum Zeitpunkt $f_{\text{TTC}} = 1.1\text{ s}$ gewarnt werden müssen.

- Bei 45m/s kann ein aufmerksamer Fahrer zum Zeitpunkt $f_{\text{TTC}} = 3.8\text{s}$ noch komfortabel bremsen. Eine *spätere* Warnung zum Zeitpunkt $f_{\text{TTC}} = 3.6\text{s}$ würde noch ausreichen, damit der unaufmerksame Fahrer alarmiert wird und mit einer Notbremsung reagiert.
- Bei Geschwindigkeiten über etwa 12m/s liegt der Bremszeitpunkt stets vor dem Ausweichzeitpunkt. Eine Deeskalation durch Ausweichen kann also meistens noch relativ spät stattfinden.

Da im niedrigen Geschwindigkeitsbereich die Warnzeitpunkte also deutlich vor den jeweiligen Reaktionszeitpunkten liegen, ist besonders hier eine Erkennung der Fahrerintention schwierig und eine Fehlwarnung damit wahrscheinlich. Im hohen Geschwindigkeitsbereich liegt der Warnzeitpunkt kurz nach dem Zeitpunkt, zu dem normalerweise eine Bremsung begonnen würde, und ist damit automatisch plausibel - es sei denn, der Fahrer weicht dem Hindernis später noch aus.

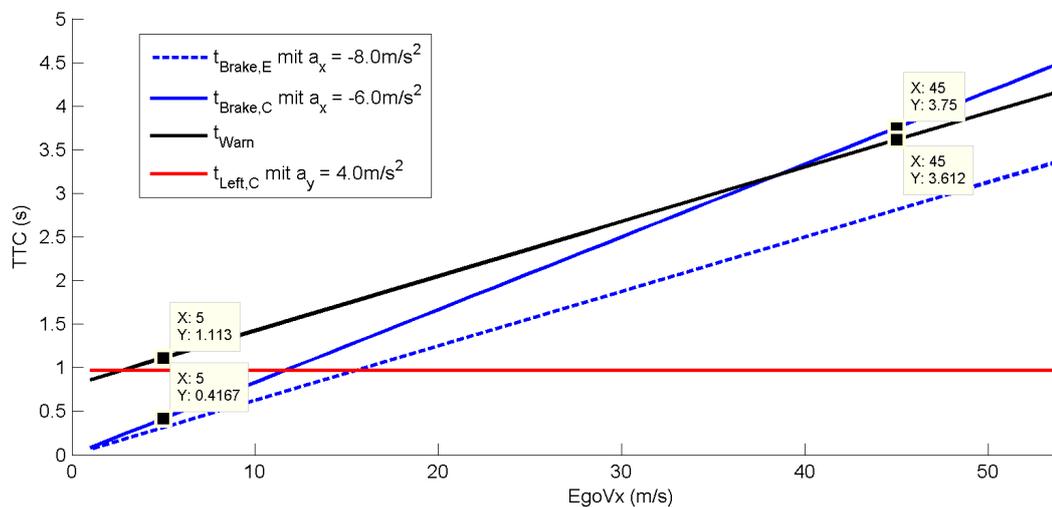


Abbildung 3.5: Warndilemma. Auf der X-Achse ist die anfängliche Ego-Geschwindigkeit in einer Annäherungssequenz aufgetragen. Die Zeit verläuft also innerhalb der fixierten Spalte von oben nach unten. Bei den markierten Beispielen liegt der Warnzeitpunkt einmal vor, einmal hinter dem Reaktionszeitpunkt.

3.5 Zusammenfassung

Anhand eines parametrierbaren Fahrermodells wurde ein Kollisionsvermeidungssystem hergeleitet, welches durch ein Warnsignal den Fahrer zu einer rechtzeitigen Notbremsung hilft. Es demonstriert die Herausforderungen: Szenen können trotz unterschiedlicher Ausgänge bis zum letzten Zeitpunkt fast identisch verlaufen, so dass die zukünftige Fahrerhandlung und damit die Fahrerintention zuverlässig geschätzt werden muss.

Eine solche Schätzung kann niemals perfekt sein, weshalb ein optimaler Tradeoff zwischen berechtigten und unberechtigten Warnungen gefunden werden muss. Das Warndilemma

erlaubt es, bei gegebener Geschwindigkeit den A-Priori-Anteil an Fehlwarnungen abzuschätzen. Zusammen mit dem Kritikalitätsmaß (Kapitel 5.3) kann ein Fahrerassistenzsystem damit für maximalen Nutzen oder hohe Akzeptanz parametrisiert werden.

4 Modellierung und Evaluierung der Fahrerintention

Im vorherigen Kapitel wurde die Wichtigkeit einer frühen Fahrerintentionserkennung motiviert; sie stellt einen essentiellen Teil von Kollisionswarnsystemen dar. Diese Früherkennung ist erforderlich, um dem Warndilemma begegnen zu können und unberechtigte Eingriffe zu vermeiden. Um die Bedeutung der Fahrerintention zu präzisieren und eine Modellierung zu erarbeiten, wird zunächst das verwendete probabilistische Framework vorgestellt. Es zeigt den Informationsfluss von den Eingangssignalen über die Fahrerintentionsschätzung hin zur letztlich gesuchten Kollisionsdetektion.

Davon ausgehend wird die Fahrerintention auf eine Größe Ψ abgebildet, welche auch die Verwendung Maschinellen Lernens ermöglicht (Kapitel 2.10). Zunächst stellt Ψ diskrete Highlevel- und Deeskalationsmanöver dar; in einer anschließenden Abstraktion wird der zukünftige Fahrzeugkurs geschätzt, wodurch die Schätzung generischer und auch für weitere Fahrerassistenzsysteme verwendbar wird.

Neben der Entwicklung von Funktionen erhält deren Bewertung häufig einen fast ebenso hohen Stellenwert: Fast immer werden transparente und objektive Kriterien benötigt, um die Qualität der Funktion zu analysieren und deren Verbesserung voranzutreiben. Geeignete Kriterien für die Bewertung eines Fahrerintentionsschätzers zu finden ist kompliziert, allein schon weil die Definition der Fahrerintention nicht eindeutig ist. Dieses Kapitel stellt dazu am Ende die erarbeiteten Ansätze vor und diskutiert die jeweilige Vor- und Nachteile.

4.1 Probabilistisches Framework

Der Zusammenhang zwischen der Fahrerintentionserkennung, der Kursprädiktion und der gesuchten Kollisionsdetektion kann elegant unter Verwendung der Wahrscheinlichkeitstheorie formalisiert werden. Die Abhängigkeiten werden in Bayesschen Netzwerken dargestellt, woraus sich drei verschiedene Modellierungen des Gesamtsystems ergeben (Abb. 4.1):

- a) Als Highlevel-Manöver $\Psi^H \in \{\text{Spurwechsel, Anhalten, ...}\}$ modelliert, beschreibt die Fahrerintention nur das Verhalten des Egofahrzeugs; Informationen über Frem-

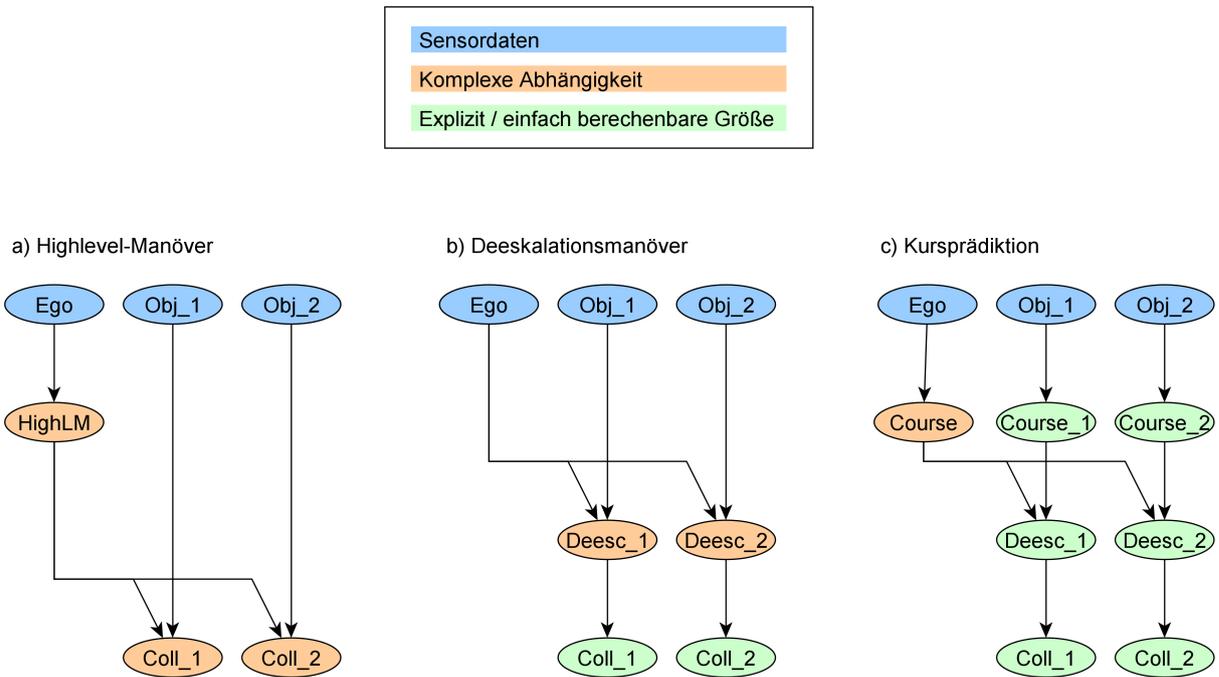


Abbildung 4.1: Modellierungen der Fahrerintention auf verschiedene Weisen, jeweils durch ein Bayessesches Netzwerk.

objekte können jedoch weitere Indikatoren liefern. Das Kollisionsereignis Ψ hängt vom Fahrmanöver und Objekt ab.

- b) Deeskalationsmanöver $\Psi^D \in \{\text{Ausweichen, Bremsen, Kollision, ...}\}$ beschreiben die zukünftige Relation zwischen dem Egofahrzeug und jedem einzelnen Objekt. Die Kollision ist ein Spezialfall der Deeskalationsmanöver, so dass alle möglichen Relationen modellierbar sind.
- c) Die Kursprädiktion C_E, C_O wird für das Egofahrzeug und die Objekte zunächst unabhängig durchgeführt. Anhand zweier Fahrzeugkurse kann das Deeskalationsmanöver Ψ^D je Objekt geometrisch abgeleitet werden.

Zur Veranschaulichung wurden in dieser Modellierung nicht alle Abhängigkeiten eingezeichnet: Zum Beispiel kann ein vorausfahrendes Objekt Indikatoren für den zukünftigen Egokurs liefern. Diese Indikatoren werden bei der Anwendung Maschinellen Lernens in Form zusätzlicher Indikatorsignale berücksichtigt (Kapitel 5.1). Die Kursprädiktion der Objekte wird stets deterministisch durch zirkuläre Prädiktion (Kapitel 4.4) durchgeführt, um den Fokus auf dem Egofahrzeug zu belassen.

Im Folgenden wird nur noch von einem einzelnen Objekt ausgegangen (Kapitel 5.2). Die Zufallsvariablen werden abgekürzt zu S_E, S_O, C_E, C_O (Sensordaten und Kursprädiktion für das Egofahrzeug und Objekt). Unter Ausnutzung der stochastischen Unabhängigkeit

wird die Wahrscheinlichkeit des Kollisionsereignisses Ψ^C durch Marginalisierung wie folgt berechnet:

$$p(\Psi^C|S_E, S_O) = \int_{\Psi^H} p(\Psi^C|\Psi^H) p(\Psi^H|S_E, S_O) \quad (4.1)$$

$$p(\Psi^C|S_E, S_O) = \int_{\Psi^D} p(\Psi^C|\Psi^D) p(\Psi^D|S_E, S_O) \quad (4.2)$$

Die Kursprädiktion wird von der Darstellung als Trajektorie zunächst in Deeskalationsmanöver Ψ^D umgerechnet (Kapitel 4.3):

$$p(\Psi^D|S_E, S_O) = \int_{C_E} \int_{C_O} p(\Psi^D|C_E, C_O) p(C_E|S_E, S_O) p(C_O|S_O) \quad (4.3)$$

Diese Verteilung wird in (4.2) eingesetzt, um wiederum die Kollisionswahrscheinlichkeit zu erhalten. Die drei Modellierungen der Fahrerintention werden im Folgenden detailliert vorgestellt.

4.2 Highlevelmanöver

Um Vergleichsmöglichkeiten mit Resultaten anderer Forschungsgruppen (z.B. [BED08, PL99]) zu schaffen und gleichzeitig auf leicht verständlichen Klassen zu arbeiten, wurden im ersten Schritt Highlevel-Manöver untersucht. Sie werden als Zufallsvariable Ψ^H dargestellt, welche die folgenden Werte annehmen kann:

ψ_{Neutral}	Neutrale Klasse, wenn kein Manöver stattfindet
ψ_{LCL}	Spurwechsel Links
ψ_{LCR}	Spurwechsel Rechts
$\psi_{\text{StopBehindCar}}$	Anhalten

Diese Manöver wurden als relevant für die Kollisionserkennung identifiziert - sie drücken die Ausweich- und die Anhalteabsicht aus. Zum Labelling wurde offline anhand des Videobildes jedem Zeitpunkt der Daten genau eine Klasse zugewiesen (Abb. 4.2). Beginn und Ende der jeweiligen Manöver wurden vergeben, sobald der Vorgang im Videobild sichtbar war bzw. abgeschlossen war. Anhaltevorgänge galten dann als beendet, sobald keine weitere Annäherung an das Vorderfahrzeug mehr zu sehen war.

Die Ergebnisse der Prädiktion auf diesen Klassen zeigen, dass mit den untersuchten Sensoren die Fahrmanöverschätzung grundsätzlich möglich ist und ließen wertvolle Erkenntnisse über zeitliche Abtastung und Wahl der Merkmale zu (Kapitel 8.4). Ein hinreichend genauer Vergleich mit Ergebnissen anderer Forschungsgruppen ist jedoch unmöglich, da sich sowohl die sensorische Konfiguration als auch die Wahl der Manöverklassen stets unterscheiden.

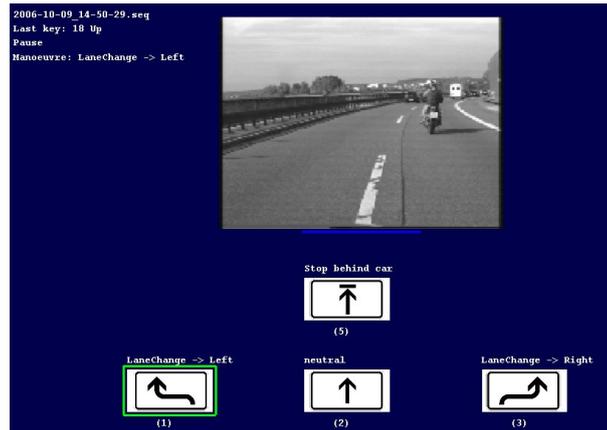


Abbildung 4.2: Annotationstool für Fahrdaten. Anhand des Videobildes wird jedem Zeitpunkt manuell eine Highlevel-Klassen zugewiesen - in diesem Fall Spurwechsel nach links

4.3 Deeskalationsmanöver

Die *Fahrerintention* bezüglich Deeskalationsmanövern wird als die Aktion definiert, welche zur Beendigung einer kritischen Annäherung führt. Sie beschreibt damit zukünftige Relation zwischen dem Egofahrzeug und Objekt beschreibt. Modelliert durch eine Zufallsvariable Ψ^D kann dies eine der folgenden Aktionen sein:

$\psi_{Neutral}$	Neutrale Klasse bei unkritischer Situation
ψ_{Left}	Ausweichen Links
ψ_{Right}	Ausweichen Rechts
ψ_{Brake}	Bremsen, bis keine weitere Annäherung stattfindet
ψ_{Coll}	Kollision mit dem Objekt

Die Klasse ist stets im dem zeitlichen Bereich definiert, in dem die Situation kritisch gilt - also während einer kritischen Szene $S = (t', t'')$ (Kapitel 5.3). Man beachte, dass eine Kollision hier auch als Deeskalationsmanöver bezeichnet wird: Trotz scheinbaren Widerspruchs kann somit jeder mögliche Szenenausgang modelliert werden.

Die Klassifikation ist eindeutig bestimmbar: Bezüglich jedes Objektes kann anhand zukünftiger Daten festgestellt werden, welche Relation vorliegt, und die Fahrerintention entsprechend rückwirkend gelabelt werden (Abb. 4.4). Falls gleichzeitig gebremst und ausgewichen wird, wird die Klasse angenommen, die zuerst abgeschlossen ist - diese Kombinationen tauchen jedoch nur selten auf (Abb. 4.3).

Mit diesem Algorithmus kann auch von der Kursprädiktion auf die zukünftige Relation und damit das Deeskalationsmanöver geschlossen werden; somit ist es also möglich, $p(\Psi^D | C_E, C_O)$ zu bestimmen und mit anderen Schätzern zu vergleichen.

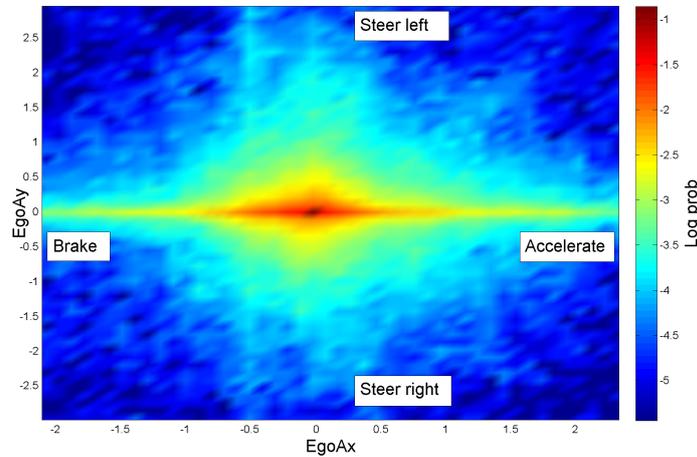


Abbildung 4.3: Häufigkeitsverteilung der Fahrzeugbeschleunigungen in Längs- und Querrichtung (X- und Y-Achse), ermittelt aus 7.8h gemischten repräsentativen Fahrdaten. Die Farbe spiegelt die logarithmische Wahrscheinlichkeit (Basis 10) wieder. Ab einer Beschleunigung von $1m/s^2$ zum Bremsen oder Ausweichen nimmt die jeweils andere Beschleunigung mit hoher Wahrscheinlichkeit den Wert 0 an.

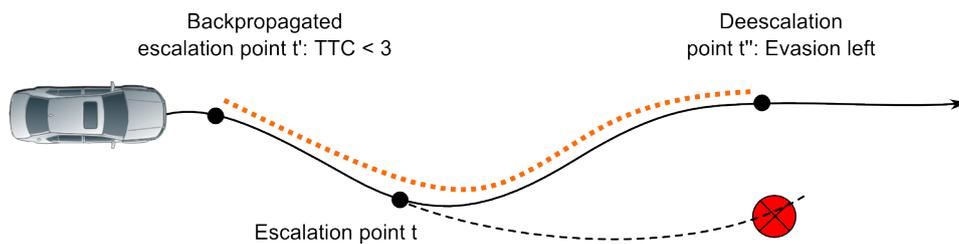


Abbildung 4.4: Definition einer eskalierenden Szene und Klassifikation des Deeskalationsmanövers: Zum Zeitpunkt t' wird die Szene akut kritisch, da die subjektiv kritische Schwelle von $TTC = 3s$ unterschritten wird und sich das Objekt zukünftig mindestens einmal innerhalb des Fahrschlauchs befindet (zum Zeitpunkt t). Zum Zeitpunkt t'' wird der abgeschlossene Ausweichvorgang nach links festgestellt; das Label ψ_{Left} wird somit im gesamten Intervall $[t', t'']$ gesetzt.

4.4 Kursprädiktion

Bei der Kursprädiktion wird davon ausgegangen, dass alles relevante Wissen über die Fahrerintention in der räumlichen Information der zukünftigen Bewegung enthalten ist. Diskrete Klassen werden damit nicht mehr benötigt, können aber für darauf aufbauende Algorithmen anhand des Kurses berechnet werden (Kapitel 4.3). Der Kurs des Ego-fahrzeugs und des potenziellen Kollisionsobjektes muss dazu über einen hinreichenden Prädiktionszeitraum geschätzt werden, um den Ausgang einer eskalierenden Situation zu bestimmen (Abb. 4.5) [BHP10].

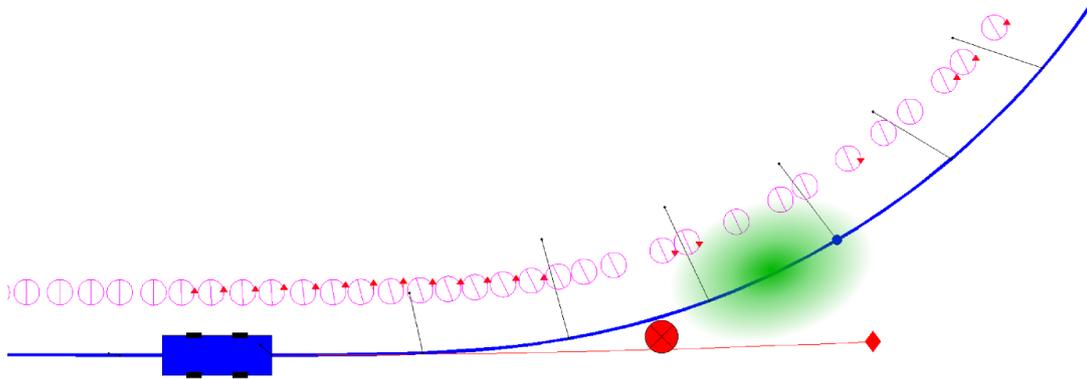


Abbildung 4.5: Kursprädiktion: Der zukünftige Fahrzeugkurs (blau) kann entweder deterministisch z.B. durch zirkuläre Extrapolation geschätzt werden (rote Linie) oder in Form einer Wahrscheinlichkeitsverteilung (grün). Letztere entstammt hier einer Schätzung mit dem Random Forest Algorithmus und wurde nur für einen einzelnen Prädiktionszeitpunkt $\Delta t = 2s$ visualisiert. Offensichtlich schätzt der Random Forest Algorithmus in dieser Szene korrekterweise eine hohe Ausweichwahrscheinlichkeit vor dem Hindernis (roter Kreis).

Der Fahrzeugkurs kann beliebig komplex modelliert werden: Für die Positionen (x, y) über den Zeitraum $t = 3s$ bei einer Abtastrate von 10 Hz ergeben sich z.B. $2 * 3 * 10 = 60$ Werte; die Dimensionalität steigt weiterhin an, wenn auch die Ausrichtung und Geschwindigkeit jedes Zustandes mitgeschätzt werden. Da der Fahrverlauf selbst in hochdynamischen Fahrsituationen stetig verläuft, sind diese Werte in hohem Grad voneinander abhängig; durch eine unabhängige Prädiktion könnte aber ein physikalisch unmöglicher Kurs entstehen. Bei der Verwendung Maschinellen Lernens für die Prädiktion muss berücksichtigt werden, dass die meisten Methoden pro vorherzusagender Größe ein trainiertes Modell benötigen. Im Beispiel müssten also 60 Modelle trainiert werden, was neben der erwarteten Unstetigkeit auch einen hohen Zeit- und Ressourcenaufwand bedeuten würde.

Der untersuchte Ansatz löst dieses Problem, indem ein parametrisches Kursmodell verwendet wird. Es approximiert den Fahrzeugkurs anhand weniger Parameter, so dass nur noch diese erlernt und geschätzt werden müssen (ähnlich [FM98]). Dadurch sinkt zwar im Allgemeinen die Flexibilität der Kursprädiktion, aber auch die Komplexität des verwendeten Modells.

Zirkuläres Kursmodell

Eine einfache Kursprädiktion entsteht aus der Anwendung des Einspurmodells (Kapitel 2.6) mit konstanter Kurskrümmung κ , Beschleunigung a und initialer Geschwindigkeit v . Für $\kappa > 0$ wird die Fahrzeugposition (x, y) zum zukünftigen Zeitpunkt t wie folgt berechnet:

$$g^{\text{Circ}}(\kappa, v, a, t) = (x, y) \quad (4.4)$$

$$l = \frac{1}{2}at^2 + vt \quad (4.5)$$

$$\psi = l\kappa \quad (4.6)$$

$$(x, y) = \kappa^{-1}(\sin \psi, 1 - \cos \psi) \quad (4.7)$$

Hierbei bezeichnet l die Länge der gefahrenen Strecke und ψ die Orientierungsänderung. Durch den kreisförmigen Kursverlauf sind die meisten Geradeaus- und Kurvenfahrten anhand weniger Parameter gut modellierbar; in dynamischen Situationen wie Spurwechseln oder spätem Ausweichen können sich jedoch Einschränkungen ergeben, und der approximierte Kurs kann sogar zu einer fehlerhaften Kollisionsdetektion führen (Abb. 4.6).

Dieses Phänomen hat zwar für die Kollisionserkennung keine zu großen Einschränkungen erbracht, wirkt sich aber geringfügig auf die Erkennungsleistung der Ausweichvorgänge aus (Kapitel 8.11.1). Im Rahmen dieser Arbeit wurde deshalb auch die Approximation der Kurse durch klothoidenförmige Trajektorien untersucht; weil dabei jedoch ein weiterer Freiheitsgrad entstand, konnte mit der verwendeten Methode der Prädiktion keine signifikante Verbesserung erzielt werden [BFHP10].

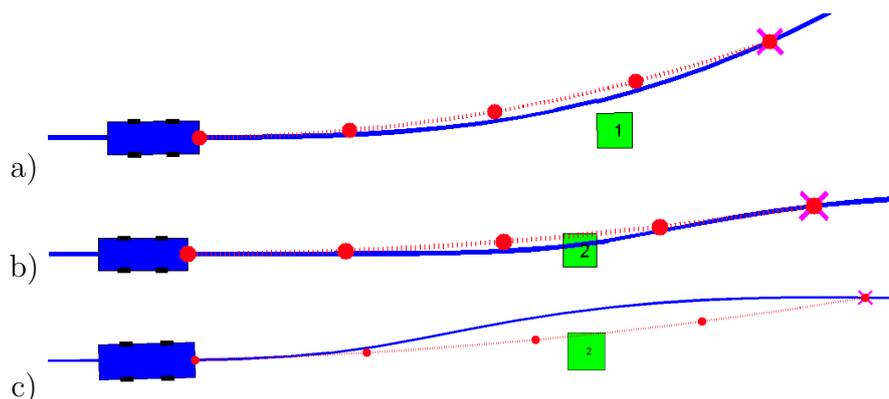


Abbildung 4.6: Approximation (rot) des zukünftigen Fahrzeugkurses (blau) durch das zirkuläre Kursmodell. In den Szenen a) und b) ist die Kursapproximation hinreichend genau, so dass der Szenenausgang bezüglich des Hindernisses (grün) gleich bleibt: Ausweichen links bzw. Kollision. In c) wird die Schwäche des Kreismodells sichtbar: Das Ausweichmanöver kann nicht durch ein einzelnes Kreissegment beschrieben werden, so dass der approximierte Kurs fälschlicherweise eine Kollision vermuten lässt.

Ermittlung der Modellparameter

Für das zirkuläre Kursmodell werden die Parameter (κ, v, a) verwendet. Weil die Fahrzeuggeschwindigkeit v stetig verläuft, wird der zum Prädiktionszeitpunkt gemessene Wert $v(t)$ verwendet. Die Beschleunigung a und Kurskrümmung κ werden von dieser Stetigkeitsannahme befreit, da ihr Wert sich bei dynamischer Fahrweise innerhalb kurzer Zeit ändern kann. Diese freien Parameter werden ermittelt, indem der Abstand des approximierten zum tatsächlichen Kurs durch eine Kostenfunktion ausgedrückt und minimiert wird.

Für die Approximation mit dem Kreismodell wurde hierfür der Luftlinienfehler e^{Beeline} am Ende des Prädiktionshorizontes verwendet (Kapitel 4.5.4), woraus sich eine geschlossene Lösung für (a, κ) ergibt. Bei komplexeren Kursmodellen mit mehr Freiheitsgraden wird eine strengere Kostenfunktion benötigt; dazu wurde ein Optimierungsalgorithmus zusammen mit der mittleren Kursabweichung e^{Area} eingesetzt [BFHP10].

4.5 Evaluierung

Um die Kursprädiktion, Klassifikation und Warnfunktion sowie deren Zusammenspiel zu bewerten, muss die Qualität der einzelnen Komponenten zuverlässig und möglichst unabhängig ausgewertet werden. Dafür werden in diesem Abschnitt zunächst gängige generische Metriken vorgestellt, die universell anwendbar sind. Die Metriken werden davon ausgehend für die Auswertung von Zeitreihen allgemein, die Fahrerintention und die Warnfunktion spezialisiert.

4.5.1 Generische Auswertungen

Die Wahrscheinlichkeit, zu einem beliebigen Zeitpunkt t eine korrekte Klassifikation zu erhalten, wird als *Gesamtkorrektheit* bezeichnet; entsprechend umgekehrt ist die *Gesamtfehlerrate* definiert. Verschiedenen Klassen werden meist unterschiedlich gut detektiert; die Handschrifterkennung funktioniert zum Beispiel für die Ziffer 0 zuverlässiger als für die Ziffer 1, welche häufig mit der Ziffer 7 verwechselt wird.

Aufgrund dieser Erfahrung wird zumeist zusätzlich klassenspezifisch ausgewertet: Je Klasse werden durch die Sensitivität, Spezifität und Präzision die Detektionsrate sowie die Robustheit gegen Fehlalarme eines Klassifikators beschrieben (Tabelle 4.1). Die Metriken sind universell und einfach zu bestimmen; zudem sind die meisten Klassifikationsalgorithmen so konzipiert, dass während des Trainings genau diese Kriterien optimiert werden. Damit können verschiedene Klassifikationsprobleme und -algorithmen miteinander verglichen werden.

$p(\hat{\Psi} = \Psi)$	Gesamtkorrektheit, dass der Klassifikator zu einem beliebigen Zeitpunkt die korrekte Klasse schätzt
$p(\hat{\Psi} \neq \Psi) = 1 - p(\hat{\Psi} = \Psi)$	Gesamtfehlerrate
$p(\hat{\Psi} = c \mid \Psi = c)$	Bedingte Sensitivität der Klasse c : Wahrscheinlichkeit, dass ein Ereignis c korrekt erkannt wird
$p(\hat{\Psi} \neq c \mid \Psi \neq c)$	Bedingte Spezifität: Wahrscheinlichkeit, dass keine Fehldetektion der Klasse c stattfindet
$p(\Psi = c \mid \hat{\Psi} = c)$	Bedingte Präzision: Zuverlässigkeit des Schätzers, modelliert durch die Wahrscheinlichkeit, dass eine Detektion der Klasse c korrekt ist

Tabelle 4.1: Generische Auswertungen eines Schätzers $\hat{\Psi}$ bezüglich der Referenzklassifikation Ψ , welche als Ground Truth angenommen wird

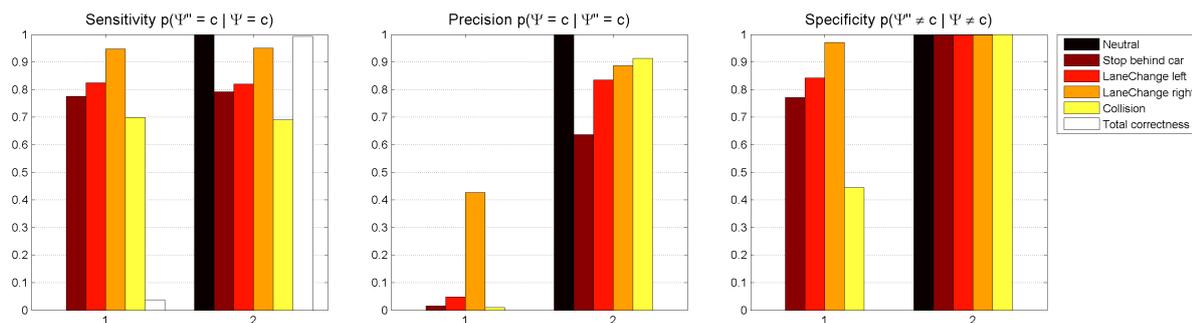


Abbildung 4.7: Generische Auswertungen von Klassifikatoren: Die Sensitivität je Klasse und die Gesamtkorrektheit bewerten die Detektionsrate, während die Präzision und Spezifität die Robustheit gegen Fehldetektionen beschreiben. Im Beispiel erzielen beide Klassifikatoren in den nicht-neutralen Klassen also eine ähnliche Detektionsleistung, jedoch ist Klassifikator 2 deutlich robuster gegen Fehlerkennungen.

4.5.2 Zeitliche Verzögerung

Bei sequentiellen Daten sind die Labels der einzelnen Zeitpunkte meistens nicht unabhängig voneinander. Zum Beispiel liegt nicht zu jedem Zeitpunkt eine unabhängige Fahrerintention vor, sondern diese erstreckt sich über den gesamten zeitlichen Verlauf eines Manövers inklusive Planung. Viele Anwendungen wie auch eine Warnfunktion sind daran interessiert, rechtzeitig die korrekte Schätzung zu erhalten - die Gesamtkorrektheit $p(\Psi = \hat{\Psi})$ allein ist nicht immer hinreichend aussagekräftig.

Häufig wird ausgewertet, ab welchem Zeitpunkt die korrekte Klasse erkannt wird (z.B. [OP00]). Es sei $S = (a, b)$ eine Sequenz der Klasse c (Kapitel 5.3). $D(a)$ ist die zeitliche Differenz zwischen dem wahren Klassenbeginn a und dem Zeitpunkt t , an welchem diese Klasse zum ersten Mal erkannt wird. Die Suche nach t wird auf eine ϵ -Umgebung von a eingeschränkt, weil ansonsten unterschiedliche Sequenzen miteinander assoziiert werden könnten. $D(a)$ wird schließlich über alle Szenenbeginne a gemittelt:

$$t = \min\{t : \hat{\Psi}(t) = c \wedge |t - a| < \epsilon\} \quad (4.8)$$

$$D(a) = t - a \quad (4.9)$$

$$D = \text{mean}\{D(a) | (a, b) = S_i, i = 1 \dots N_{\text{Scenes}}\} \quad (4.10)$$

Die so berechnete mittlere Verzögerung D besitzt zwar eine intuitive Bedeutung, jedoch nur geringen praktischen Nutzen, weil lediglich der Zeitpunkt der Erstdetektion, nicht die Sensitivität oder Robustheit ausgewertet werden. Ein randomisierter Klassifikator schätzt zum Beispiel in kurzen Abständen alle Klassen, so dass innerhalb der ϵ -Umgebung jedes Manöverbeginns diese Klasse garantiert einmal vorkommt. Die Verzögerungen $D(t)$ sind damit allesamt sehr gering, und die Auswertung erzielt scheinbar gute Resultate (Tabelle 4.2). Die Ursache liegt darin, dass keine Fehldetektionen bewertet werden, weshalb diese Metrik schnell verworfen wurde.

	Neutral	Spw. links	Spw. rechts	Anhalten
Zufallsgenerator	-0.01s	0.15s	-0.03s	0.10s
Random Forest	-0.04s	0.35s	-0.04s	0.24s

Tabelle 4.2: Mittlere zeitliche Verzögerung (s) zwischen dem gelabelten Beginn eines Manövers und dessen Erstdetektion. Der Zufallsgenerator erzielt die geringste Verzögerung, also die besten Werte. Dies hängt damit zusammen, dass alle Klassen regelmäßig geschätzt werden, und spricht eindeutig gegen diese Form der Auswertung.

4.5.3 Erweiterte bedingte Sensitivität

Um die vorherigen Auswertungsschwierigkeiten der zeitlichen Verzögerung zu umgehen, wird im Folgenden die Detektionsrate zu bestimmten Zeitpunkten bestimmt: Die erweiterte bedingte Sensitivität

$$p(\hat{\Psi} = c | \Psi = c, \mathbf{X}) \quad (4.11)$$

beschreibt die Detektionsrate einer Klasse c unter der zusätzlichen Bedingung \mathbf{X} . Für die Kollisionsvermeidung wird mit $\mathbf{X} := f_{\text{TTC}}$ z.B. die Klassifikationsrate bei gegebener TTC und damit bezüglich der Kritikalität analysiert. Statt der zeitlichen Verzögerung zum Manöverbeginn wird also zu fest gewählten Zeitpunkten die Korrektheit überprüft. Dies stellt besonders dann einen großen Vorteil dar, wenn der Manöverbeginn bei manuellem Labelling nicht genau definiert werden kann.

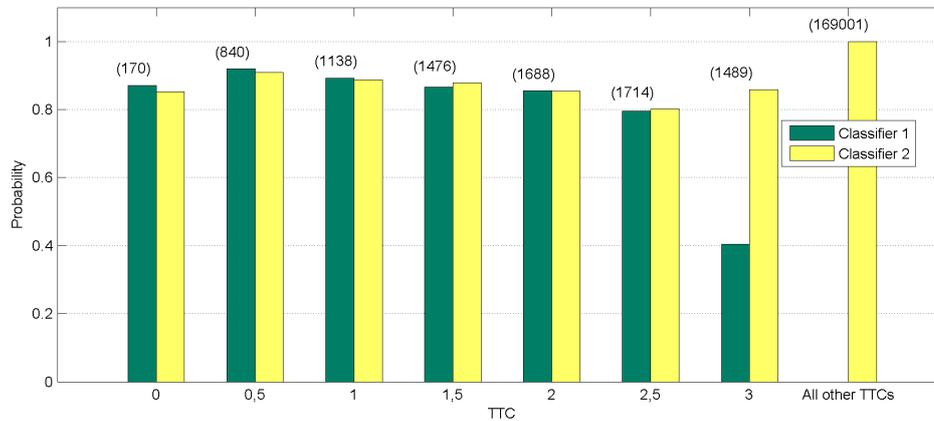


Abbildung 4.8: Erweiterte bedingte Sensitivität je Klasse und TTC. Mit fallender TTC steigt die Klassifikationsleistung an, wobei für $f_{\text{TTC}} \in [0s, 0.5s]$ deutlich weniger Zeitpunkte in die Auswertung einfließen konnten als für die anderen Bereiche.

Während die Formel in der Theorie durch ihre Einfachheit und Universalität besticht, gibt es in der Praxis folgende Besonderheiten zu beachten: Die zusätzliche Bedingung \mathbf{X} ist meist nicht gleichverteilt, z.B. existieren nur sehr wenige Szenen mit $f_{\text{TTC}} < 0.5s$. Weil in diesen Fällen das Ergebnis verfälscht werden könnte, hilft es, die absolute Anzahl ausgewerteter Zeitpunkte mit abzubilden (Abb. 4.8).

4.5.4 Auswertung der Kursprädiktion

Weil die Kursprädiktion eine kontinuierliche Modellierung der Fahrerintention ist, wird sie unterschiedlich zu den diskreten Schätzern ausgewertet. Grundsätzlich gibt es hierfür verschiedene Möglichkeiten, etwa den Längs- oder Querabstand der geschätzten zur wahren Position. Dabei muss jedoch berücksichtigt werden, dass im Fall einer Kurvenfahrt diese Größen ihre Bedeutung verändern, da sich das Koordinatensystem dreht.

Im Folgenden sei t_0 der Zeitpunkt der Prädiktion, Δt der Prädiktionshorizont, g der wahre und h der prädizierte Egokurs. Drei Auswertekriterien wurden als zuverlässig identifiziert (Abb. 4.9):

- Course distance e^{Course} : Das Lot des prädizierten Punktes $g(t + \Delta t)$ auf den tatsächlich gefahrenen Kurs h gibt Auskunft darüber, wie gut die Kursprädiktion in lateraler Richtung ist. Die Kursdistanz ist vor allem für die Kollisionsdetektion ein entscheidendes Maß, weil es darauf ankommt, ob der prädizierte Kurs eine bestimmte Position überdeckt.

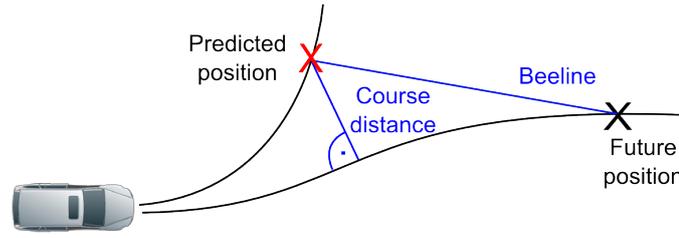


Abbildung 4.9: Auswertungskriterien für die Kursprädiktion: Die Kursdistanz und der Luftlinienabstand (Beeline) beschreiben unabhängig von eventuellen Kurvenfahrten, wie gut die Prädiktion in Längs- und Querrichtung zum tatsächlich gefahrenen Kurs ist.

- Beeline e^{Beeline} : Der Luftlinienabstand zwischen der geschätzten und wahren Position beinhaltet zusätzlich Informationen über die longitudinale Richtung. Angenommen, der Fahrer bremst oder beschleunigt, aber der prädizierte Kurs h spiegelt dies nicht wider. An dieser Stelle steigt dann der Luftlinienfehler, nicht jedoch die Kursdistanz.
- Area e^{Area} : Der mittlere Kursfehler berechnet sich als das Integral über den Luftlinienabstand während des Prädiktionszeitraums. Er entspricht damit ungefähr der Fläche, die zwischen g und h eingeschlossen wird. Laterale und longitudinale Fehler werden berücksichtigt; diese Größe ist jedoch am schwierigsten zu berechnen.

Mit diesen Kriterien kann auch in Kurvenfahrten die Prädiktionsgüte akkurat und invariant von der Drehung des Koordinatensystems berechnet werden. Sie sind leicht verständlich und berechenbar:

$$e^{\text{Course}}(g, h, t_0, \Delta t) = \min\{|g(t_0 + t) - h(t_0 + t)| : t = 0 \dots \Delta t\} \quad (4.12)$$

$$e^{\text{Beeline}}(g, h, t_0, \Delta t) = |g(t_0 + \Delta t) - h(t_0 + \Delta t)| \quad (4.13)$$

$$e^{\text{Area}}(g, h, t_0, \Delta t) = \int_{t=0}^{\Delta t} e^{\text{Beeline}}(g, h, t_0, t) dt \quad (4.14)$$

4.5.5 Nutzen des Warnsystems zur Unfallvermeidung

Für eine anwendungsnahe Auswertung wird der resultierende Nutzen des Klassifikators zur Unfallvermeidung geschätzt. Dazu wird gemäß der Konstruktionsannahmen des Warnsystems (Kapitel 3.3) simuliert, dass der Fahrer nach einer Warnung mit der Reaktionszeit c_{React} eine Notbremsung einleitet. Die Hilfsfunktionen $\Psi^{\text{Warn}}(t)$ und $\Psi^{\text{Coll}}(t)$ drücken aus, dass zum Zeitpunkt t eine Warnung geschieht und in den Daten eine Kollision stattgefunden hat. Die Fahrerreaktion auf die Warnung wird dann über die Beschleunigung a simuliert:

$$\Psi^{\text{Warn}}(t) \wedge \Psi^{\text{Coll}}(t) \rightarrow a(t + c_{\text{React}}) = a_{\text{Brake},E} \quad (4.15)$$

In einer Kollisionsszene kann so ermittelt werden, ob mit der Warnfunktion der Unfall komplett vermieden wird oder zumindest wieviel Geschwindigkeit abgebaut wird.

Für Szenen ohne Kollision darf umgekehrt nicht jede Warnung als Fehlalarm interpretiert werden: Aufgrund von Unaufmerksamkeit oder um das System zu testen steuern Fahrer

manchmal sehr spät und unkomfortabel. Auch in diesen Szenen würde eine Warnung zu einer früheren Reaktion und damit mehr Sicherheit und Komfort führen; die Akzeptanz des Fahrers steigt, weil die Funktionsweise des Systems transparent wird [LHH04].

Szene	Warnung	keine Warnung
Kollisionsszene	Nutzen	Unternutzen
Unkomfortable Szene	Nutzen	neutral
Neutrale Szene	Übernutzen	neutral

Tabelle 4.3: Auswertung der Warnfunktion in den möglichen Situationen der Anwendung

Tabelle 4.3 stellt die möglichen Kombinationen von Szenenausgang und dem Verhalten der Warnfunktion dar. Die finale Auswertung beinhaltet die Anzahl vermiedener Kollisionsszenen, Fehlalarmsszenen und das Einsparpotenzial an Kollisionsgeschwindigkeiten und unkomfortablen Beschleunigungen.

4.5.6 Diskussion der Auswertungsmöglichkeiten

Die generischen Auswertungen sind leicht verständlich, klar formalisiert und erlauben den Vergleich mit anderen Klassifikationsproblemen oder -algorithmen. Bedingte Auswertungen wie die Korrektheit in kritischen Fahrsituationen sind immer noch generisch durchführbar, beschreiben die Performanz aber bereits fokussiert auf bestimmte Szenen.

Dennoch berücksichtigen diese Auswertungen manche Schwierigkeiten nicht, die sich erst in der Anwendung zeigen. So wird im Allgemeinen die Robustheit oder Sensitivität auf allen Zeitpunkte ermittelt, eine Warnfunktion benötigt den untersuchten Klassifikator aber nur zu bestimmten Zeitpunkten. Für eine wirklich anwendungsnahe Bewertung muss deshalb das gesamte System betrachtet werden, indem zum Beispiel der Nutzen zur Kollisionsvermeidung such Simulation bestimmt wird.

Eine solch komplexe Auswertung ist jedoch ebenfalls mit Vorsicht zu genießen: Da ein Fahrermodell und eine Warnstrategie verwendet werden, ist im Fehlerfalle nicht klar feststellbar, welche Komponente verantwortlich ist. In einer Fehlalarmsszene kann z.B. die Fahrerreaktionszeit überschätzt sein, die Warnstrategie zu geringe Beschleunigungswerte annehmen oder letztendlich die Fahrerintentionsschätzung falsch liegen. Per definitionem sagen diese komplexen Auswertungen also nichts über die einzelnen Komponenten, sondern nur über das gesamte simulierte System aus.

Erfreulicherweise haben Versuche im Rahmen dieser Arbeit gezeigt, dass die Ergebnisse der verschiedenen Auswertungen stets dieselbe Tendenz aufweisen und sich gegenseitig bekräftigen. Um zugleich die starke Überzahl der Neutral-Klasse zu kompensieren (Abb. 8.1), wird zumeist die generische Metrik

$$p(\hat{\Psi} = \Psi \mid \Psi^{\text{Critical}}) \quad (4.16)$$

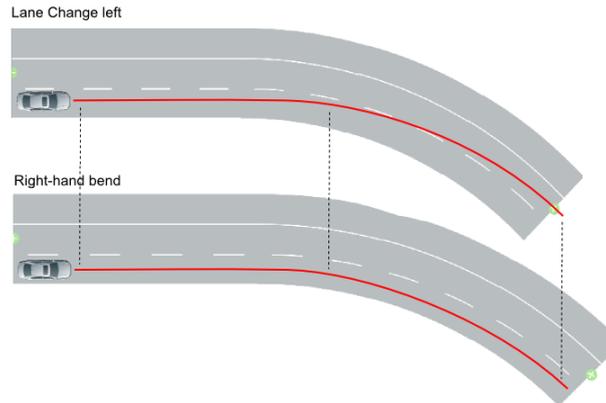


Abbildung 4.10: Mehrdeutigkeit der Highlevel-Manöverklassen: Dieselben physikalischen Fahrdaten werden oben als Spurwechselvorgang, unten als Kurvenfahrt (Klasse Neutral) bezeichnet. Diese Mehrdeutigkeit erschwert den Einsatz Maschinellen Lernens, da Prädiktionsfehler häufig hierauf zurückführbar sind.

verwendet - die Gesamtkorrektheit des Schätzers $\hat{\Psi}$ in kritischen Szenen (Kapitel 5.3). Fehlschätzungen in unkritischen Momenten spielen damit praktisch keine Rolle, was akzeptiert werden kann, da die die Kritikalität auch online feststellbar ist.

Die Kursschätzung wird geometrisch ausgewertet, und ein Vergleich beider Schätzer wird durch Ableitung der diskreten Fahrerintention sowie die anwendungsnahe Auswertung ermöglicht.

4.6 Zusammenfassung

Eine geschickte Modellierung der Fahrerintention ist eindeutig, leicht auswertbar und minimiert Abhängigkeiten zu anderen Größen, z.B. irrelevanten Objektdaten (Abb. 4.1). Highlevelmanöver sind weit verbreitet, weisen jedoch für Kollisionswarnsysteme und Maschinelles Lernen zu viele Nachteile auf: Beginn und Ende der jeweiligen Manöver sind nicht exakt definierbar, und auch die Klasse selbst kann nicht immer eindeutig benannt werden (Abb. 4.10). Diese nicht eindeutigen Labels, auch für Emotionserkennung wohl bekannt [Rus94, EF71], erschweren Training und Auswertung der Klassifikatoren.

Deeskalationsmanöver wurden als objektivere Klassen eingeführt, die auch eine Kollision modellieren können. Sie haben den Vorteil, dass die Ground Truth offline automatisch ermittelt werden kann. Es wurden hilfreiche Erkenntnisse zur Prädiktion gesammelt (Kapitel 8.4), dennoch ergaben sich einige Schwachpunkte: Zum Beispiel wird die Fahrerintention stets bezüglich eines einzelnen Objektes geschätzt; fährt das Egofahrzeug zwischen zwei Fahrzeugen durch, so liegt bei identischen Egodaten einmal die Klasse Ausweichen links, einmal Ausweichen rechts vor. Verschiedene Klassen in derselben Situation können bei Maschinellen Lernen schnell zu Overfitting führen.

Ein weiterer Nachteil ist die starke Ungleichverteilung der Klassen: Die Kollisionsszenen sind stark unterrepräsentiert und konnten häufig nur schlecht erkannt werden. Ebenfalls

sollte die Definition des Klassenbeginns anhand der Kritikalität vermieden werden, weil hierdurch verschiedene Faktoren miteinander vermischt werden, und nur ein geringe Teilmenge der Daten genutzt werden kann.

Die meisten dieser Schwierigkeiten werden durch die Kursprädiktion behoben: Von ihr kann auf einfache geometrische Art eine Schätzung der Deeskalationsklassen für Vergleichszwecke und Warnfunktion abgeleitet werden. Gleichzeitig ist sie leicht auswertbar und universeller anwendbar, denn die Modellierung enthält eine saubere Trennung der Ego- und Objektinformation.

5 Untersuchte Signaldaten und Vorverarbeitung

Dieses Kapitel beschreibt zunächst die verwendete Sensorik und davon abgeleitet wichtige Vorverarbeitungsschritte. Zunächst werden die Daten serialisiert, indem alle detektierten Objekte unabhängig voneinander verarbeitet werden; dieser Schritt ist nötig, um die Datenkomplexität und Overfitting zu vermeiden. Anschließend werden die sequentiellen Daten in kritische Teilszenen unterteilt, was für die Auswertung und Labelvergabe der Fahrerintention eine zentrale Rolle spielt.

Schließlich werden aus den Roh-Signaldaten, die von der Fahrzeugsensorik geliefert werden, weitere Merkmale durch Anwendung von Expertenwissen extrahiert. Dieser Schritt ist bei der Verwendung Maschinellen Lernens üblich und bringt Algorithmen zur Mustererkennung i.A. bessere Ergebnisse.

Die Definition und Berechnung der Vermeidungsbeschleunigung und verbleibender Reaktionszeit ermöglicht es, den Eskalationsgrad einer Szene subjektiv und objektiv zu beschreiben. Diese Kennzahlen stellen damit eine entscheidende Brücke zwischen dem Kritikalitätsempfinden des Fahrers und der Konzeption des Warnsystems dar.

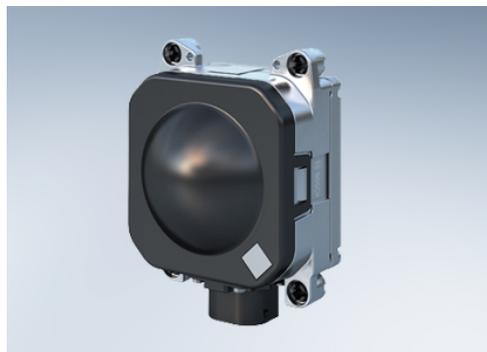


Abbildung 5.1: Fernbereichsradarsensor von Bosch: LRR3, Long-range Radar der 3. Generation [Rob03]. Mit einer Kantenlänge von unter 8cm wird er meist in der mittleren Fahrzeugfront verdeckt und nach vorn gerichtet verbaut.

5.1 Sensorik und Signale

Der LRR3-Radarsensor von Bosch ist ein Umfeldsensor. Er arbeitet im Frequenzbereich 77GHz, hat einen Öffnungswinkel von etwa 30° und kann Objekte in bis zu 250m Entfernung detektieren (Abb. 5.1). Zusätzlich stehen auch fahrzeuginterne Signale über den CAN-Bus in jedem Serienfahrzeug zur Verfügung. Die in dieser Arbeit verwendeten Signale und daraus extrahierte Merkmale sind in Tabelle 5.1 zusammengefasst.

Fahrdynamische Signale

a_x	EgoAx	Egolängsbeschleunigung
a_y	EgoAy	Egoquerbeschleunigung
v_x	EgoVx	Egolängsgeschwindigkeit
κ	Kappa	Krümmung des geschätzten Egokurses
	KappaDt	Krümmungsänderung

HMI-Signale

DirIndL/R	Blinker links/rechts aktiv
WheelAngle	Egovorderradwinkel
WheelAngleDt	Ableitung des Egovorderradwinkels, ist korreliert mit der Drehgeschwindigkeit des Steuerrades
GasPedal	Gaspedalwert
GasPedalDt	Veränderung der Gaspedalstellung
Braking	Bremse aktiv
PedalFeature	Kombinierte Gaspedal- und Bremsaktivität

Objektdaten

d_x	ObjectDx	Längsabstand des Objektes
d_y	ObjectDy	Querversatz des Objektes
$d_{y,C}$	ObjectDyCirc	Kursversatz
$v_{x,Rel}$	ObjectVxRel	Relative Geschwindigkeit des Objektes
$v_{y,Rel}$	ObjectVyRel	Relative Quergeschwindigkeit
	ObjectAxRel	Relative Längsbeschleunigung des Objektes
	ObjectVx	Absolute Längsgeschwindigkeit des Objektes
	ObjectVy	Absolute Quergeschwindigkeit des Objektes
	ObjectAy	Absolute QuerLängsbeschleunigung des Objektes
	TTC	Time to collision

Tabelle 5.1: Signale im Messfahrzeug und jedem Serienfahrzeug, für welches das untersuchte System ausgelegt ist. Ein Symbol wird nur für die Signale eingeführt, die im Detail behandelt werden.

5.2 Verwendung der Multiobjektdate

Die meisten Umfeldsensoren können eine Vielzahl von Objekten gleichzeitig detektieren. Um diese adressieren zu können, wird jedem Objekt eine eindeutige Nummer (ID)

zugewiesen. Die IDs werden ähnlich einer Warteschlange an die gerade detektierten Objekte verteilt, und da dies normalerweise nur endlich viele sind, können die IDs als nach oben hin beschränkt aufgefasst werden.

Dieser Aspekt erscheint zunächst rein technisch, er hat jedoch große Auswirkungen auf den Einsatz Maschinellen Lernens. Praktisch gesehen werden die Daten schnell sehr hochdimensional, da zu jedem detektierbaren Objekt eine große Anzahl relevanter Eigenschaften existiert. Gleichzeitig sind in dem Fall, dass nur eine geringe Anzahl Objekte detektiert wird, viele dieser Signale nicht gesetzt bzw. haben den Wert 0. Die Abwesenheit des Objekts wird meist dadurch realisiert, dass je Objektkanal ein binäres Signal angibt, ob es im aktuellen Messzyklus detektiert wurde oder nicht.

Für Maschinelles Lernen werden Merkmalsvektoren f benötigt, die Objektdateien müssen also auf die f_i verteilt werden. Da unter den Objekten keine Reihenfolge definiert ist, wäre der nächstliegende technische Ansatz, gemäß der Objekt ID vorzugehen. Sei M die Anzahl objektspezifischer Merkmale und O die maximale Anzahl an Objekt IDs. Unter Verwendung der Merkmalsreihenfolge aus Tabelle 5.1 ergäbe sich die Zuordnung

$$\begin{aligned} f_1 &= \text{ID 1, ObjectDx} \\ f_2 &= \text{ID 1, ObjectDy} \\ &\dots \\ f_{M+1} &= \text{ID 2, ObjectDx} \\ f_{M+2} &= \text{ID 2, ObjectDy} \end{aligned}$$

Weil die ID eines Objektes keinerlei Information trägt, werden die Objektdateien also willkürlich auf die Merkmale aufgeteilt. Damit entsteht die Gefahr des *Overfittings*, ein Klassifikator könnte zum Beispiel lernen, dass mit Objekt 1 stets kollidiert wird und Objekt 2 stets ausgewichen wird. Gleichzeitig ist die Verallgemeinerung unmöglich, denn die Muster tauchen bei unterschiedlichen IDs auch in verschiedenen Merkmalen f_i auf. Beide Beobachtungen könnten nur vermieden werden, indem zu jeder möglichen Permutation von Objekten und IDs dieselben Trainingsdaten verwendet würden, was offensichtlich sämtliche theoretischen Grenzen übersteigt.

Eine Vorsortierung der Objekte, z.B. nach ihrer relativen Position zum Egofahrzeug, wäre hier hilfreich - sie benötigt jedoch Wissen über die Relevanz der Objekte und ist damit schwer modellierbar. Stattdessen wurde ein einfacherer Ansatz verfolgt und jedes Objekt separat behandelt. Die gemeinsamen Merkmale, die nicht objektspezifisch sind (Fahrodynamik und HMI) wurden jeweils vervielfältigt. Die Daten haben also dieselbe Gestalt, als ob dieselben Fahrten mehrmals stattgefunden hätten, wobei stets nur ein Objekt existiert hätte (Abb. 5.2).

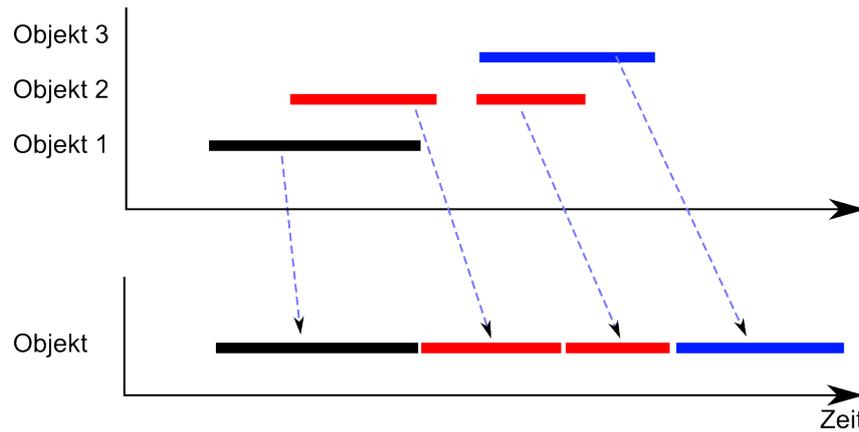


Abbildung 5.2: Serialisierung von Multiobjektdaten: Um die Dimensionalität der Daten niedrig zu halten und die variierende Anzahl gleichzeitig detektierter Objekte zu behandeln, werden die Multiobjektdaten in Singleobjektdaten umgeformt. Hierzu wird stets nur ein Objekt separat betrachtet, und die nicht objektrelevanten Daten vervielfältigt.

5.3 Szeneneinteilung und Kritikalität

Die Sensordaten können als zeitkontinuierliche Funktionen betrachtet werden (Kapitel 1.5). Zur praktischen Verwendung und Auswertung der Klassifikatoren werden häufig Szeneneinteilungen benötigt:

- S_{Measures} : Eine Partitionierung der Offline-Daten in jeweils unabhängig voneinander aufgenommene Messdaten. Aufgrund dieser Szenen wurde z.B. die Aufteilung in Trainings- und Testdaten durchgeführt. Die Partitionierung entsteht i.A. beim Importieren der Daten.
- S_{Objects} : Eine Objektszene beginnt mit dem Zeitpunkt der Erstdetektion eines Objektes und endet mit dem Zeitpunkt, zu dem das Objekt das letzte Mal detektiert wurde. Diese Szene entspricht also genau der Objektlebensdauer, formuliert aus Perspektive des Sensors.
- S_{Critical} : Auswahl der Szenen, die anhand des TTC-Kriteriums als subjektiv kritisch eingestuft werden. Genau diese Szenen sind für die Fahrerintentionsschätzung von hoher Relevanz.

Eine einzelne Szene ist ein Tupel $S = (a, b)$, bestehend aus Start und Endzeitpunkt; eine Szeneneinteilung ist entsprechend eine Folge $\mathbf{S} = (S_1, S_2, \dots)$ disjunkter Szenen S_i .

Kritische Szenen

Die Unterteilung S_{Critical} wird für Verwendung historischer Informationen und für die Auswertung der Fahrerintentionserkennung benötigt. Zum Beispiel fährt der Egofahrer einem anderen Fahrzeug im Stau mehrmals dicht auf, wodurch kritische TTCs erreicht

werden; schließlich überholt er es links (Abb. 5.3). Die gesamte Szene ist also mehrmals warnwürdig (Kapitel 3.3) und muss folglich in zwei kritische Teilszenen zerlegt werden.

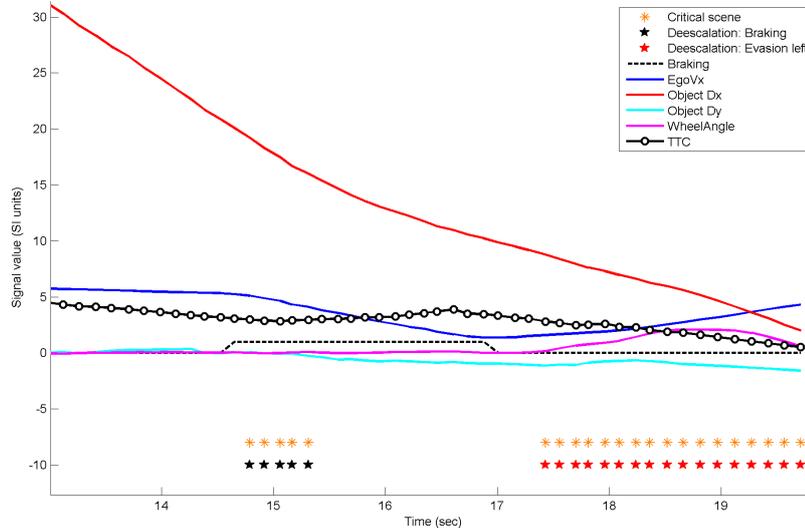


Abbildung 5.3: Definition kritischer Szenen. In der ersten kritischen Teilszene (14.7s, 15.4s) liegt eine schnelle Annäherung vor, weil die TTC die kritische Grenze von 3 Sekunden leicht unterschreitet. Diese Eskalation wird durch Bremsen behoben, was an der fallenden Geschwindigkeit f_{EgoVx} erkennbar ist. Die zweite kritische Szene liegt im Bereich (17.4s, 19.8s) vor - diesmal fährt der Fahrer links am Objekt vorbei, was am Objektversatz $f_{ObjectDy}$ ersichtlich wird.

Zur genauen Definition kritischer Szenen dienen die folgenden Kriterien:

- Das potenzielle Gefahrenobjekt muss sich im Fahrschlauch befinden, dh. dessen Kursversatz lässt mit den Fahrzeugbreiten auf eine Kollision schließen.
- Der Kursversatz kann verursacht durch leichte Lenkbewegungen schwanken; umgekehrt kann er auch erst spät eine Kollision anzeigen, etwa wenn der Fahrer aufgrund eines Fahrfehlers sehr spät auf das Kollisionsobjekt zusteuert. Deshalb genügt eine einmalige Erfüllung dieses Kriteriums während der gesamten Szene (Abb. 4.4).
- Der Szenenanfang wird über den Zeitpunkt definiert, an dem die Situation subjektiv als kritisch empfunden wird, also $f_{TTC} < 3s$ (Kapitel 2.9). Umgekehrt kann man folgern, dass der Fahrer spätestens ab diesem Zeitpunkt einen Handlungsentschluss gefasst hat, was den Szenenanfang und damit Labelbeginn der Fahrerintention begründet.
- Als Ausgang einer eskalierten Szene muss genau eines der möglichen Deeskalationsmanöver definiert sein; das Szenenende liegt also genau an dem Zeitpunkt vor, an dem der Abschluss eines Deeskalationsmanövers festgestellt wird (Kapitel 4.3).

Die booleschen Hilfsfunktionen Ψ^{Approx} und Ψ^{Critical} formalisieren das Kollisions- und Kritikalitätskriterium:

$$\Psi^{\text{Approx}} \equiv |d_{y,C}| < 1.9m \quad (5.1)$$

$$\Psi^{\text{Critical}} \equiv f_{\text{TTC}} < 3s \quad (5.2)$$

Damit kann nach einem Zeitpunkt t_0 die nächste Szene $S = (a, b)$ bestimmt werden:

$$a = \min\{a > t_0 : \Psi^{\text{Approx}}(a) \wedge \Psi^{\text{Critical}}(a)\} \quad (5.3)$$

$$b = \max\{b : b > a \wedge \forall t \in [a, b] : \Psi^{\text{Critical}}(t)\} \quad (5.4)$$

Die Menge aller Szenen $\mathbf{S}_{\text{Critical}}$ wird iterativ berechnet. Die Definition bietet somit eine obere Schranke für die Menge aller kritischen Szenen - es ist ein notwendiges, aber nicht hinreichendes Kriterium für eine warnwürdige Szene.

5.4 Scheinziele

Neben den Vorzügen eines Radarsensors wie weitgehender Wetter- und Lichtunabhängigkeit, verdeckter Einbaumöglichkeit und präziser longitudinaler Auflösung gibt es auch Schwächen, die in der Praxis berücksichtigt werden müssen: Mit dem meist zugrundeliegenden FMCW-Verfahren sind laterale Messungen nur indirekt möglich [Rob03]; die Erkennung der Objektbreite und -höhe und damit deren Klassifikation ist häufig noch schwierig [DDM⁺09]. Während zur Lösung dieses Problems derzeit Fusionskonzepte erforscht und umgesetzt werden, ist eine grundsätzliche Ungewissheit der Objekterkennung nie völlig ausschließbar.

Ein häufiger resultierender Effekt sind *Scheinziele* - Messungen von Objekten, die nicht real existieren oder die keinen Einfluss auf das Fahrgeschehen haben. Sie werden jedoch im Gefahrenbereich oder mit einer Bewegung in dessen Richtung gemessen und beeinflussen damit die weiteren Rechenschritte. In der Praxis treten besonders die folgenden Arten von Scheinzielen regelmäßig auf:

- Überfahrbare Ziele (Gullideckel, Eisenbahnschienen, Bordsteinkanten), die aufgrund ihrer Beschaffenheit und hohem Metallanteil eine hohe Rückstrahlleistung haben und damit fast immer von einem Radarsensor detektiert werden (Abb. 5.4).
- Unterfahrbare Ziele (Autobahnschilder, Brücken)
- Reflexpunktwanderung: Während der Vorbeifahrt an Fahrzeugen kann systematisch beobachtet werden, dass sich der Überlappungsbereich des Objektes mit dem Sensorerfassungsbereich verringert. Der Reflexpunkt und damit die gemessene Position des Fahrzeugs bewegt sich also paradoxerweise in Richtung des Egofahrzeugs, so dass ein präzisierendes Warnsystem einen Einscherer vermuten würde (Abb. 5.5).

- Trackingannahmen: Um vereinzelte Fehlmessungen zu kompensieren, werden Objektpositionen mit dem Kalman-Filter getrackt. Im Falle eines schnellen Abbiegens an eine Kreuzung kann eine zu konservative Schätzung dazu führen, dass das Objekt noch im Gefahrenbereich vermutet wird, obwohl es diesen schon verlassen hat.
- „Geisterobjekte“: Dieser Begriff hat sich für Detektionen von Objekten eingebürgert, die nicht existieren. Sie können z.B. durch Radarreflexion im Schnee, Spiegelungen an der Leitplanke oder sonstige sensorspezifische Phänomene verursacht sein.

Für die meisten Arten von Scheinzielen existieren effektive Gegenmaßnahmen; dennoch ist das Auftreten vereinzelter Situationen unvermeidbar, in welchen Scheinziele zu spät oder überhaupt nicht als solche klassifiziert werden können.

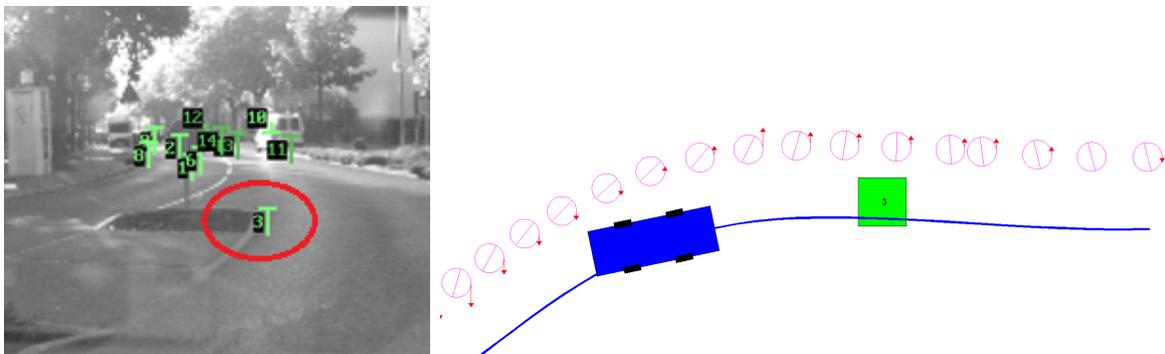


Abbildung 5.4: Scheinziele. Links: Die Kante der Verkehrsinsel wird als Objekt 3 detektiert und fälschlicherweise als stehendes Hindernis klassifiziert. Rechts: dieselbe Szene aus der Vogelperspektive: Die Breite dieses Hindernisses wird überschätzt und ragt in den zukünftigen Fahrzeugkurs hinein, was einen Fehleingriff hervorrufen würde.

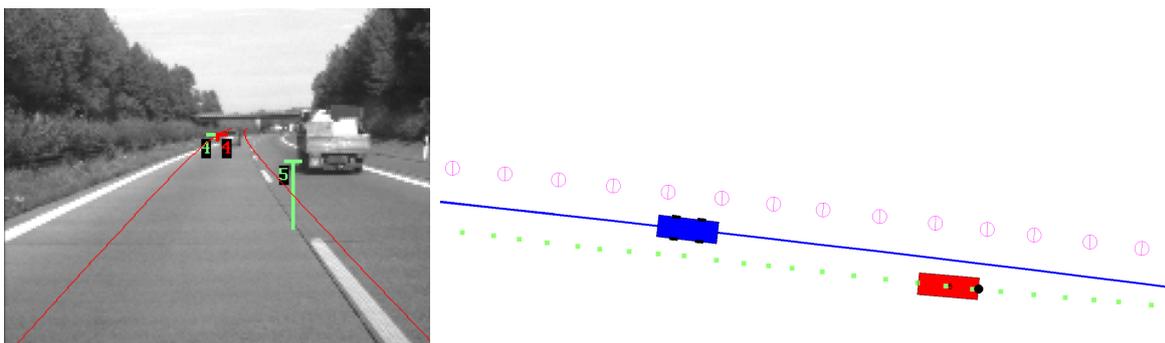


Abbildung 5.5: Scheinbewegungen. Links: Objekt 4 befindet sich auf der Fahrspur (schwer erkennbar) und wurde als Folgeobjekt für das ACC-System ausgewählt (rote Schrift). Objekt 5 bezeichnet den LKW auf der rechten Spur, mit hoher Varianz in longitudinaler Richtung gemessen (grüne Linien). Rechts: Im Verlauf der Fahrt scheint sich der LKW lateral anzunähern, Ursache dafür ist der systematisch wandernde Radar-Reflexpunkt auf der Stoßstange. Diese Annäherungen können fehlerhafte Unterstützungen verursachen.

Bedeutung für die Fahrerintentionserkennung

Weil die Klassifikation von Scheinzielen im Voraus nicht perfekt möglich ist, wurde eine bewusste Entscheidung getroffen: Unter der Annahme, dass der Egofahrer im Kollisionsfall auf das Kollisionsobjekt nicht reagiert, unterscheiden sich die sensorischen Daten von echten und scheinbaren Kollisionsszenen nicht. Damit können prinzipiell alle Daten verwendet werden.

Neben der entfallenden Vorfilterung steigt damit auch der Datenanteil in der stark unterrepräsentierten Kollisionsklasse. Bei der Prädiktion ergeben Scheinzieldaten jedoch meistens schlechtere Ergebnisse, weil diese Objekte teilweise erst sehr spät auftauchen, z.B. nach dem Warnzeitpunkt, und damit eine unvermeidbare Kollision darstellen. Deshalb wurden Scheinziele nur für das Modelltraining verwendet und aus den Prädiktionsergebnissen ausgefiltert.

5.5 Transformation der Kursparameter

Die zur Kursprädiktion verwendeten Parameter (a_x, κ) (Kapitel 4.4) hängen systematisch mit der Geschwindigkeit des Fahrzeugs zusammen: Bei höheren Geschwindigkeiten nehmen Kurskrümmungen und Beschleunigung ab. Diese Beobachtung müsste durch maschinelles Lernen explizit erkannt und modelliert werden, was dessen Komplexität und Fehleranfälligkeit unnötig steigern würde. Gleichzeitig steigt der systematische Diskretisierungsfehler mit den kleineren Werten.

Im Folgenden werden Transformationen beschrieben, welche diese Zusammenhänge durch einfache expertenkodierte Transformationen eliminieren; dadurch wird der Diskretisierungsfehler gering gehalten und das Maschinelle Lernen vereinfacht.

5.5.1 Zielkoordinaten

Diese Transformation basiert auf der psychologischen Erkenntnis, dass der Fahrer zunächst die zu erreichende Position (x, y) entscheidet und davon abhängig die motorischen Schritte umsetzt [PL99]. Es wird also angenommen, dass die Zielkoordinaten aus Perspektive des aktuellen Koordinatensystems die gesamte Information über die Fahrerintention beinhalten. Die Parameter (a_x, κ) werden folglich in die Koordinaten umgerechnet, die der entsprechende Kurs nach einer festen Zeit Δt erreicht:

$$(a_x, \kappa) \rightarrow (x, y) = g^{\text{Circ}}(\kappa, v_x, a_x, \Delta t) \quad (5.5)$$

5.5.2 Normierung der Kurskrümmung anhand der Komfortgrenzen

Die Kurskrümmung κ , Geschwindigkeit v_x und Querbewegung a_y weisen den physikalischen Zusammenhang aus (5.6) auf. a_y ist zusammen mit der Geschwindigkeit v_x ein direktes Maß für die physikalisch mögliche Kurskrümmung und die Belastung des Fahrers [SH82]. Der maximal tolerierte Wert wurde empirisch ermittelt und als Funktion $a_{y,Max}(v_x)$ in Abhängigkeit von der Geschwindigkeit modelliert. Wird a_y ins Verhältnis zu diesem Maximum gesetzt, ergibt sich die relative Querbewegung (5.7):

$$a_y = \frac{v^2}{r} = v^2 \kappa \quad (5.6)$$

$$a_{y,Rel} = \frac{a_y}{a_{y,Max}(v_x)} \quad (5.7)$$

$$(a_x, \kappa) \rightarrow (a_x, a_{y,Rel}) \quad (5.8)$$

Die Abbildung der Kursparameter auf die relative Beschleunigung ist damit bijektiv, und der Parameter $a_{y,Rel}$ beschreibt das Einhalten der Komfortgrenzen unabhängig von v_x . Die Kurskrümmung wird über den gesamten Geschwindigkeitsbereich fast gleichmäßig auf das Intervall $[-1, 1]$ umgerechnet (Abb. 5.6).

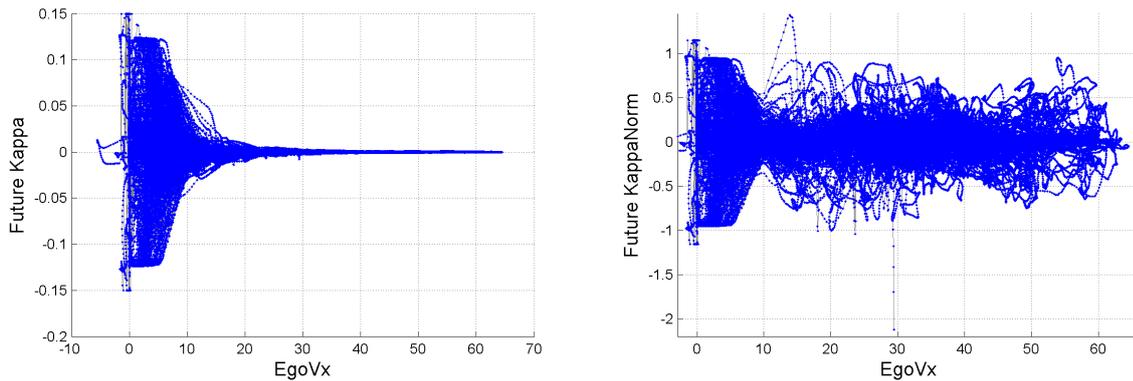


Abbildung 5.6: Normalisierung der Kurskrümmung. Links ist die systematische Abhängigkeit der Kurskrümmung κ von der Geschwindigkeit V_x zu sehen. Rechts wurde diese Abhängigkeit verringert, indem durch die maximal vom Fahrer tolerierte Kurskrümmung dividiert wurde. Damit liegen die meisten Werte im Intervall $[-1, 1]$, einzelne Ausreißer zeigen Szenen, in denen die Toleranzgrenzen überschritten wurden.

5.5.3 Differenzkodierung der Parameter

Die Kursparameter (a , κ) sind zu den meisten Zeitpunkten konstant und enthalten damit eine große Redundanz. Durch Differenzkodierung kann die Information komprimiert werden: Es wird nicht der eigentliche Wert, sondern die Differenz zum vorherigen Wert verwendet - eine Vorgehensweise, die in der Sprachverarbeitung häufig zur Datenkompression angewandt wird [PK08]. Es ergibt sich die Transformation

$$(a_x, \kappa) \rightarrow (\Delta x_x, \Delta \kappa) = (a_x - a_0, \kappa - \kappa_0) \quad (5.9)$$

wobei (a_0, κ_0) die aktuell gemessenen Parameter und (a_x, κ) die Parameter des zukünftigen Kurses sind. Die Differenzkodierung kann äquivalent auch auf die normierte Kurskrümmung angewandt werden, woraus sich die Größen $(\Delta a_x, \Delta a_{y,Rel})$ ergeben.

5.6 Merkmalsextraktion

Merkmalsextraktion ist ein beliebtes Hilfsmittel, um Klassifikationsalgorithmen durch Expertenwissen ihre Arbeit zu erleichtern. Durch Kombination oder Umrechnung vorhandener Signale werden neue Merkmale hinzugefügt, die direktere Rückschlüsse auf die Schätzklasse erlauben als die Rohsignale. In der Theorie sind die Grenzen zwischen der Merkmalsextraktion und der anschließenden Klassifikation fließend: Wenn ein Merkmalsextraktor bereits die zu schätzende Klasse selbst berechnet, so wird der Klassifikator trivial [DHS01]; umgekehrt könnte ein perfekter Klassifikationsalgorithmus die Merkmalsextraktion mitlernen und so überflüssig machen.

Weit verbreitete Merkmale, die in der Sprach- oder Bilderkennung eingesetzt werden, sind Frequenzanalysen, Hauptkomponentenanalyse, Glättungen, Kantendetektoren und affine Abbildungen zur Rotation oder Größenänderung von Bildbereichen. Häufig werden bei der Merkmalsextraktion großzügig neue Merkmale berechnet und dem Trainingsalgorithmus die Auswahl der signifikanten überlassen. Dieses Vorgehen bietet zusätzliche Erkenntnis über die Daten, denn zu wissen, dass bestimmte Merkmale zur Klassifikationsverbesserung beitragen oder nicht, ist ein wertvolles Analyseergebnis. Die in dieser Arbeit untersuchten Merkmale werden im Folgenden vorgestellt.

5.6.1 Generische Merkmale

Zunächst wurden die zur Mustererkennung üblichen Merkmale aus den Daten berechnet. Damit wurden folgende Erfahrungen gemacht (Kapitel 8.4):

- Die *Fourieranalyse* berechnet zu den Signaldaten das Frequenzspektrum. Die am stärksten vertretene Frequenz könnte prinzipiell Informationen über die aktuelle Steuergeschwindigkeit enthalten. Allerdings weisen die Fahrdaten keine regelmäßigen Schwingungen auf, und die Resultate dieses Schritts waren unverwendbar.
- Mit der *Wavelet-Transformation* werden Signale mit einem prototypischen Muster gefaltet; dadurch ergeben sich praktisch Gradienten und Mittelwerte zu verschiedenen Zeitbereichen. Experimentell konnten dadurch für die Fahrerintentionserkennung keine Vorteile gewonnen werden, weil die entsprechende Information der Gradienten bereits in den Merkmalen enthalten war.
- Die Hauptachsentransformation (*PCA*) wurde nur kurz untersucht: Sie berechnet lineare Kombinationen von Merkmalen, um deren Varianzen zu minimieren. Die

Bedeutung dieser Linearkombinationen ist jedoch nur schwer zu interpretieren, und aufgrund ebenfalls keiner Verbesserungen wurde PCA nicht weiter verfolgt.

Die Berechnung von Gradienten und deren Glättung hingegen brachte große Vorteile, weil dadurch die zeitliche Veränderung der Daten auf einfache Weise ausgedrückt werden konnte. Diese Gradienten wurden nur für die Signale κ , f_{GasPedal} , $f_{\text{WheelAngle}}$ bestimmt, da zu den anderen Signalen entweder bereits der Gradient mitgemessen wird oder die Signale diskret sind.

5.6.2 Kursversatz

Ein effektives Merkmal für die Verwendung in Fahrerassistenzsystemen ist die Verwendung des Kursversatzes, zum Beispiel für die Spurzuordnung. Die fest montierte Umfeldsensorik misst normalerweise die Lateralabstände von Objekten bezüglich der Fahrzeugachse. Ein Fahrzeug auf der linken Nachbarspur hat damit bei Geradeausfahrt z.B. einen Lateralabstand von $d_y = 3m$; in einer Linkskurve würde sich der Abstand zusätzlich vergrößern, während in einer Rechtskurve das Objekt sogar vor dem Sensor erscheinen und $d_y = 0$ erzielen kann.

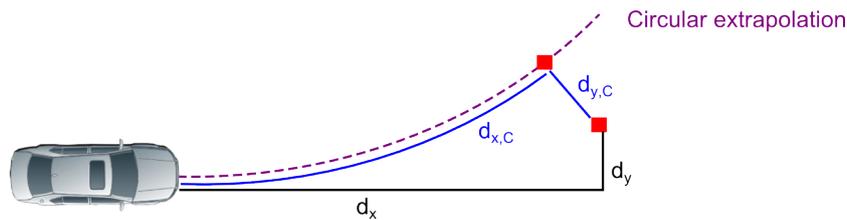


Abbildung 5.7: Ermittlung des Kursversatzes $d_{y,C}$ durch zirkuläre Kursprädiktion.

Leichte Kurven verursachen also bereits große Varianzen von d_y , welche eine direkte Spurzuordnung unmöglich machen. Dieses Problem wird umgangen, indem die Objektkoordinaten umgerechnet werden: Statt des Lateralabstandes zur Fahrzeugachse wird der Abstand zum zukünftigen Fahrzeugkurs bestimmt. Er wird in diesem Kontext bewusst sehr einfach durch zirkuläre Prädiktion geschätzt (Abb. 5.7) und folglich als $d_{y,C}$ bezeichnet. Entsprechend kann der Längsabstand auch umgerechnet werden zu $d_{x,C}$, womit eine bijektive Abbildung hergestellt wird.

5.6.3 Time to collision (TTC)

Das menschliche Kritikalitätsempfinden korreliert stark mit der geschätzten Kollisionszeit, was wiederholt in Studien gezeigt werden konnte (Kapitel 2.9). In dieser Untersuchung wird das Merkmal TTC deshalb an verschiedenen Stellen genutzt: Zum Einen wird daran definiert, wo eine potenziell kritische Szene beginnt oder endet (Abschnitt 5.3); zum anderen dient es als wichtiges Indikatorsignal für Maschinelles Lernen.

Die geschätzte Kollisionszeit f_{TTC} beschreibt die Zeit, nach der zwei Fahrzeuge unter der Annahme konstanter Geschwindigkeiten kollidieren würden. Der Lateralversatz spielt bei der Berechnung keine Rolle, so dass in die Berechnung lediglich der aktuelle Abstand d_x und die Relativgeschwindigkeit v_x eingehen:

$$f_{\text{TTC}} = \frac{d_x}{v_x} \quad (5.10)$$

Unter Umständen wird neben der TTC auch die erweiterte TTC (f_{ETTC}) verwendet, in deren Berechnung zusätzlich die relative Beschleunigung eingeht. Dadurch erhält die TTC prädiktive Anteile: Wenn während einer schnellen Annäherung z.B. stark gebremst wird, nimmt f_{ETTC} schnell wieder hohe und damit unkritische Werte an. f_{TTC} bleibt hingegen solange im kritischen Bereich, bis die Relativgeschwindigkeit v_x gesunken ist. Durch die Verwendung der Beschleunigung ist f_{ETTC} zwar realistischer, jedoch auch unetwiger als die f_{TTC} ; die Wahl des richtigen Merkmals hängt also vom Einsatzzweck ab.

5.6.4 Vermeidungsbeschleunigung

Für ein Manövers ψ bezeichnet die Vermeidungsbeschleunigung a_ψ die benötigte Intensität, um eine Kollision zu verhindern (Abb. 5.8). Sie ist definiert als

$$a_\psi(t) = \min\{a \mid \text{Ausführung von Manöver } \psi \text{ zum Zeitpunkt } t \text{ mit Intensität } a \text{ vermeidet eine Kollision}\} \quad (5.11)$$

In eskalierenden Situationen sinkt die Distanz zum Kollisionsobjekt mit fortschreitender Zeit t , womit die Vermeidungsbeschleunigung wächst. Für deren Berechnung werden die maximalen Beschleunigungen, die zwischen Fahrzeug und Straße aufgebracht werden können, berücksichtigt: Sie werden durch den Kamm'schen Kreis ausgedrückt und ermöglichen eine Herleitung der optimalen Ausweichtrajektorie [SOB06, SSI08].

Bei einer unschärfebehafteten Umfeldwahrnehmung ist eine exakte Bestimmung der Trajektorie kaum möglich, so dass Vereinfachungen gemacht werden können: Die Bewegung des Egofahrzeugs und des potenziellen Kollisionsobjektes wird durch ein parabelförmiges Längs- und Querbewegungsmodell approximiert; damit können die Vermeidungsbeschleunigungen $a_\psi(t)$, $\psi \in \{\psi_{\text{Brake}}, \psi_{\text{Left}}, \psi_{\text{Right}}\}$ mit einer geschlossenen Formel berechnet werden. Durch eine kreisförmige Ausweichtrajektorie wird eine höhere Genauigkeit erreicht; eine geschlossene Berechnung ist dann aber nur für stehende Kollisionsobjekte möglich.

5.6.5 Reaktionszeitpunkte

Der Zeitpunkt, an dem der Fahrer spätestens ein Deeskalationsmanöver ψ einleiten muss, um eine Kollision zu vermeiden, wird als *Reaktionszeitpunkt* bezeichnet. Er hängt von der Intensität a ab, mit der das Manöver durchgeführt wird:

$$t_{\psi,a} = \max\{t \mid a_\psi(t) \leq a\} \quad (5.12)$$

Mit steigender Intensität a verlagert sich der Reaktionszeitpunkt nach hinten. Um den letztmöglichen Reaktionszeitpunkt zu bestimmen, wird also die Intensität auf den maximal möglichen Wert $a_{x,E}$ oder $a_{y,E}$ gesetzt, der durch das Fahrermodell gegeben ist (Kapitel 3.2). Abb. 5.8 demonstriert diese Zeitpunkte an den Beispielen Bremsen und Ausweichen.

Die Berechnung des Reaktionszeitpunktes ist keineswegs trivial, denn die Formel beinhaltet Informationen über das *zukünftige* Verhalten der beteiligten Fahrzeuge. In einem Online-System werden dafür die benötigten Bewegungen durch das Kreismodell (Kapitel 4.4) mit konstanter Beschleunigung und Kurskrümmung prädiziert. Laterale Informationen können optional gänzlich ignoriert werden, es wird dann vereinfacht angenommen, dass alle Objekte sich hintereinander in derselben Spur befinden und geradeaus fahren. Mit diesen Annahmen kann $t_{\text{Brake},a}$ geschlossen berechnet werden.

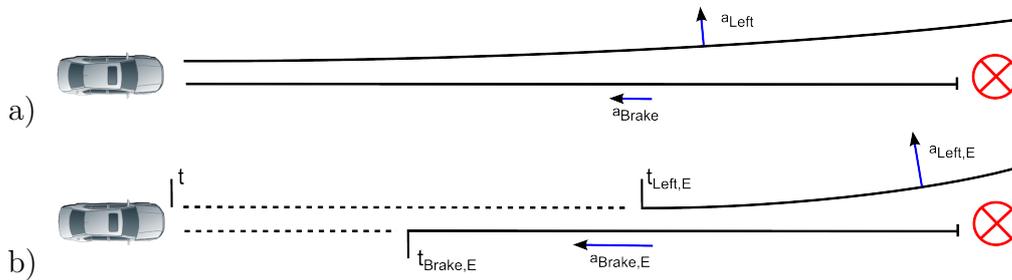


Abbildung 5.8: a) Vermeidungsbeschleunigungen: In der vorliegenden Szene nähert sich das Egofahrzeug einem stehenden Objekt mit konstanter Geschwindigkeit. Wenn der Fahrer zum aktuellen Zeitpunkt mit der Querbeschleunigung a_{Left} nach links steuert oder mit der negativen Längsbeschleunigung a_{Brake} bremst, wird eine Kollision vermieden. b) Reaktionszeitpunkte: Wenn mit maximaler Kraft $a_{\text{Brake},E}$ gebremst wird, so muss der Bremsvorgang zum Zeitpunkt $t_{\text{Brake},E}$ stattfinden. Analog dazu ist ein Ausweichvorgang im letztmöglichen Moment $t_{\text{Left},E}$ dargestellt.

5.6.6 Time To React (TTR)

Während der Online-Anwendung werden die Reaktionszeitpunkte $t_{\psi,a}$ in jedem Rechenschritt neu geschätzt, um zu jedem Zeitpunkt alle bis dahin bekannten Informationen auszunutzen. Der absolute Reaktionszeitpunkt selbst dient vor allem zur Formalisierung und Veranschaulichung. Für die praktische Anwendung ist es sinnvoller, relative Angaben zu verwenden - also die zum Zeitpunkt t verbleibende Zeit bis zum Reaktionszeitpunkt:

$$\Delta t_{\psi,C} = t_{\psi,C} - t \quad (5.13)$$

$$\Delta t_{\text{React}} = \max\{t_{\psi,E}\}_{\psi=1\dots N_C} \quad (5.14)$$

Damit ist z.B. $\Delta t_{\text{Brake},E} = f_{\text{TTRBrake}}$ die verbleibende Zeit, bis ein Notbremsmanöver eingeleitet werden muss. Spätestens nach $\Delta t_{\text{React}} = f_{\text{TTR}}$ muss eine Reaktion erfolgen, damit die Szene deeskaliert werden kann. Dies ist also die maximale Zeitspanne, die der Fahrer inaktiv bleiben darf, und wird als *Time To React (TTR)* bezeichnet.

5.6.7 Vergleich

Sowohl f_{TTC} als auch f_{TTR} schätzen den Eskalationsgrad einer Situation auf einer Zeitskala. Für f_{TTC} konnte empirisch der Zusammenhang zur subjektiven Kritikalität belegt werden (Kapitel 2.9); so kann von dem aktuellen Wert auf die innere Eskalationsphase und den Entscheidungszeitpunkt des Fahrers zurückgeschlossen werden. Das Merkmal wird auf einfache Weise berechnet und zeigt einen stetigen Verlauf (Abb. 5.9).

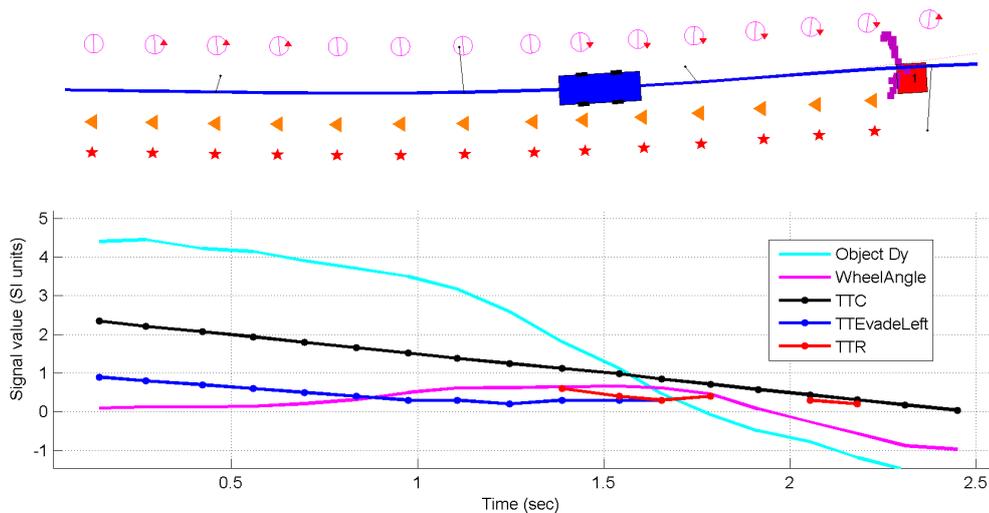


Abbildung 5.9: Schätzung des zeitlichen Fortschritts eines Ausweichmanövers. f_{TTC} verläuft gleichmäßig fallend. $f_{TTEvadeLeft}$ schätzt die übrige Zeit bis zu einem komfortablen Ausweichvorgang nach links; sie wird fast 0, da der Fahrer im letzten Moment ausweicht. f_{TTR} ist die Zeit bis zum letztmöglichen Ausweichzeitpunkt. Liegt das Objekt außerhalb des Fahrschlauchs, wie vor und während des Ausweichvorgangs, sind die entsprechenden Zeiten nicht definiert.

Im Gegensatz dazu modelliert f_{TTR} die physikalische Kritikalität; mit $f_{TTR} = 0$ liegt z.B. der letztmögliche Zeitpunkt einer Deeskalation vor. Dieses Merkmal eignet sich vor allem zur Formalisierung einer Warnfunktion. Da es von jedem Brems- oder Lenkvorgang beeinflusst wird, verläuft es deutlich unstetiger als f_{TTC} . Tabelle 5.2 gibt einen zusammenfassenden Überblick.

Bezeichnung	Bedeutung
f_{TTC}, f_{ETTC}	Geschätzte Zeit zur Kollision ohne bzw. mit Berücksichtigung der Beschleunigung
f_{TTR}	Verbleibende Zeit, bis die spätest mögliche Handlung eingeleitet werden muss
$f_{TTBrake},$ $f_{TTEvadeLeft},$ $f_{TTEvadeRight}$	Verbleibende Zeit, bis das entsprechende Manöver mit der maximaler Intensität, die das Fahrermodell zulässt, begonnen werden muss

Tabelle 5.2: Indikatoren für den zeitlichen Fortschritt oder die Kritikalität des Verkehrsgeschehens.

5.7 Zusammenfassung

Durch Serialisierung werden Multiobjektdaten zu Singleobjektdaten vereinfacht und damit tauglich für Maschinelles Lernen gemacht. Scheinziele werden bewusst wie reguläre Objekte verwendet, um mehr Kollisionsszenen zu erhalten. Weitere notwendige Vorverarbeitungsschritte sind die Einteilung der Daten in kritische Szenen und die Transformation der Kursparameter in möglichst gut präzifizierbare Größen.

Mit der Merkmalsextraktion wird durch Kombination bestehender Signale das Maschinelle Lernen verbessert. Anwendungsspezifische Merkmale wie TTC oder Vermeidungsbeschleunigung beschreiben den Eskalationsgrades einer Situation lassen Rückschlüsse auf die subjektive oder objektive Kritikalität zu; sie werden mitunter für die Formalisierung der Warnfunktion verwendet.

6 Grundlagen der Datenverarbeitung

Um Maschinelles Lernen für die Schätzung der Fahrerintention oder Kursprädiktion zu verwenden, stellt dieses Kapitel die Grundlagen von Hidden Markov Modellen sowie des Random Forest Algorithmus vor. Beide Verfahren werden im weiteren Verlauf der Arbeit methodisch miteinander kombiniert und detailliert ausgewertet.

6.1 Hidden Markov Modelle

Hidden Markov Modelle sind generative Modelle, welche Wahrscheinlichkeitsverteilungen von Sequenzen modellieren und darauf optimiert werden können. Nach den folgenden Grundlagen sei als exzellente Weiterführung auf Rabiners Tutorial [Rab89] verwiesen; die verwendeten Algorithmen werden in [DHS01, PK08] graphisch anhand des Trellis-Diagramms sehr gut veranschaulicht.

Stochastischer Prozess

Der zeitliche Verlauf einer Zufallsvariablen X wird als *stochastischer Prozess* bezeichnet. Dies kann zum Beispiel der Verlauf der Größe einer Pflanze sein, und $X(t)$ beschreibt die Größe zum Zeitpunkt $t \in \mathbf{R}^{\geq 0}$. X wird in (6.1) zunächst deterministisch modelliert, unter den vereinfachten Annahmen eines konstanten und unbegrenzten Wachstums:

$$X(t) = at + b \tag{6.1}$$

$$X(t) \sim N(at + b, \Sigma) \tag{6.2}$$

Unsicherheiten und Messschwankungen werden modelliert, indem X nun als Zufallsvariable betrachtet wird. In (6.2) ist dies unter Verwendung der Normalverteilung mit Erwartungswert $at + b$ und Standardabweichung Σ beschrieben. X ist ein *zeitdiskreter stochastischer Prozess*, wenn $t \in \mathbf{N}^{\geq 0}$ - in realen Anwendungen ist dies meistens der Fall.

Markov Kette

In (6.2) stellt t die bisherige Dauer des gesamten Wachstumsprozess dar - diese Information steht jedoch im Allgemeinen nicht zur Verfügung bzw. ist bei beschränkter Historie

nicht von Belang. Sie wird eliminiert, indem die Veränderung gegenüber dem letzten diskreten Zeitpunkt modelliert wird:

$$X_0 = N(\mu, \Sigma) \quad (6.3)$$

$$X_{t+1} = N(\mu + a\Delta t, \Sigma) \quad (6.4)$$

Nun liegt eine Markov Kette erster Ordnung vor, in welcher die Größe der Pflanze die Rolle des Systemzustandes spielt, und nur der letzte Zustand relevant ist. Allgemein ist eine *Markov Kette k-ter Ordnung* definiert als ein zeitdiskreter Zustand X_t , dessen Wert nur von den letzten k Zeitschritten, also den vorherigen Zuständen S_{t-1}, \dots, S_{t-k} abhängt:

$$X_t \perp X_{t-2}, X_{t-3}, \dots | X_{t-1} \quad (6.5)$$

$$p(X_t | X_{t-1}, X_{t-2}, \dots) = p(X_t | X_{t-1}, \dots, X_{t-k}) \quad (6.6)$$

Markov Modelle

Sind die Zustandsübergangswahrscheinlichkeit $p(X_t | X_{t-1})$ für alle t konstant, spricht man von einem *zeitstationären Prozess*. Ist der Zustand darüberhinaus eine diskrete Größe mit endlichem Wertebereich $S_t \in \{1, \dots, Q\}$, werden die Übergangswahrscheinlichkeiten in Form einer Matrix $a_{i,j} = p(S_{t+1} = j | S_t = i)$ dargestellt. Die initiale Zustandsverteilung wird durch $\pi_i = p(S_1 = i)$ beschrieben. Ein solches *Markov Modell* kann äquivalent als gerichteter Graph dargestellt werden, dessen Knoten die Zustände darstellen und Kanten mit den $a_{i,j}$ beschriftet sind (Abb. 6.3).

Hidden Markov Modelle

Im bisherigen Beispiel war der Systemzustand direkt beobachtbar. Häufig ist die beobachtbare Größe jedoch nicht von unmittelbarer Bedeutung, sondern sie ist abhängig von einer zusätzlichen verborgenen Größe, deren Wert gesucht ist. Das bekannte Urnen und Kugeln Beispiel verdeutlicht das Zusammenspiel beobachtbarer und versteckter Größen. Hierbei befinden sich in einem nicht einsehbaren Raum drei Urnen, jeweils zu verschiedenen Anteilen mit farbigen Kugeln gefüllt; in jedem Zeitschritt wählt ein Gini (mystisches Zauberesen, in Beispielen häufig verwendet) zuerst eine Urne und zieht dann aus dieser zufällig eine Kugel. Die Farbe ist die einzige Information, die nach außen dringt, und die wahre gewählte Urne soll geschätzt werden.

Das Beispiel besteht aus zwei stochastischen Prozessen: Zum Zeitpunkt t bezeichnet S_t die unbekannt gewählte Urne und damit den versteckten Zustand. O_t beschreibt die Farbe der gezogenen Kugel, also die Beobachtung. Offensichtlich ist O_t nur von S_t abhängig. Falls die Urnenwahl nicht rein zufällig ist, sondern von der zuletzt gewählten Urne abhängt, ist S ein Markov Prozess erster Ordnung. S und O zusammen ergeben dann ein *Hidden Markov Modell* (Abb. 6.1).

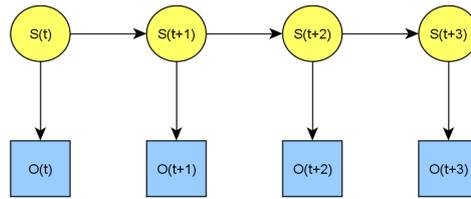


Abbildung 6.1: Ein Hidden Markov Modell, dargestellt als dynamisches Bayessches Netzwerk (Lattice View). Die Zustände S sind versteckt, während die Beobachtungen O in den Daten vorhanden sind.

Im diskreten Fall sei $\mathbf{O} = \{1, \dots, M\}$ die Menge möglicher Beobachtungen, die auch als Symbole bezeichnet werden. Die Beobachtungswahrscheinlichkeiten $p(O_t|Q_t)$ werden als Matrix $b_{i,k} = p(O_t = k|S_t = i)$ dargestellt. Ein HMM ist also zusammenfassend ein Tupel $\lambda_c = (Q, \mathbf{O}, \pi, A, B)$ und besteht aus den Elementen aus Tabelle 6.1.

Q	Anzahl oder Menge der Zustände
\mathbf{O}	Domäne der Beobachtungen
$\pi = \{\pi_i\}$	Anfangswahrscheinlichkeiten
$A = \{a_{i,j}\}$	Übergangswahrscheinlichkeit
$B = \{b_{i,k}\}$	Beobachtungswahrscheinlichkeit

Tabelle 6.1: Elemente eines Hidden Markov Modells mit diskretem Beobachtungsraum \mathbf{O} . Im kontinuierlichen Fall werden die Beobachtungswahrscheinlichkeiten $b_{i,k}$ meistens durch Gaussian Mixtures ersetzt.

Als wichtige Erweiterung kann die Einschränkung auf diskrete Beobachtungen aufgehoben werden, die bei der Verwendung realer Daten nur selten auftauchen. Damit können beliebige, mehrdimensionale kontinuierliche Beobachtungen $O_t \in \mathbf{R}^d$ modelliert werden. $p(O_t|S_t)$ wird meistens als Summe von Gaussverteilungen modelliert und in dieser Arbeit durch den Random Forest Algorithmus verbessert (Kapitel 7.3).

6.1.1 Algorithmen

Im Zusammenhang mit HMMs sind vor allem Problemstellungen mit den typischen Algorithmen für die Sequenzanalyse und zum Modelltraining verbreitet:

Alpha-Beta-Algorithmus

Im Analysefall ist ein HMM λ sowie eine Sequenz von Beobachtungen $O = o_1, \dots, o_T$ gegeben, $p(O|\lambda)$ wird gesucht. Die Bedingung über das Modell λ kann häufig entfallen zwecks verkürzter Schreibweise $p(O)$. Weil die zugrundeliegende Sequenz der versteckten Zustände unbekannt ist, müssen hierfür alle möglichen Fälle gewichtet addiert werden:

$$p(o_1, \dots, o_T) = \sum_{q_1, \dots, q_T} p(o_1, \dots, o_T | q_1, \dots, q_T) p(q_1, \dots, q_T) \quad (6.7)$$

$$= \sum_{q_1, \dots, q_T} \prod_{i=1}^T p(o_i | q_i) p(q_i | q_{i-1}) \quad (6.8)$$

In dieser Umformung wurde die stochastische Unabhängigkeit der O_t sowie die Markov Eigenschaft des versteckten Prozesses Q_t verwendet; ferner ist $p(q_1|q_0) := \pi_1$ die Startverteilung der Zustände. Naiv implementiert sind für diese Berechnung exponentiell viele Schritte nötig; der *Alpha-Beta-Algorithmus* basiert auf dynamischer Programmierung und löst das Problem in der Laufzeit $O(Q^2T)$, also in Linearzeit bezüglich der Länge der Sequenz. Als Zwischenergebnisse werden unter anderem die Variablen $\alpha_{q,t} := p(O, Q_t = q)$ berechnet, welche die Aktivität eines Zustands q zum Zeitpunkt t darstellen.

Viterbi-Algorithmus

Mitunter ist nicht die Wahrscheinlichkeit $p(O|\lambda)$ gesucht, sondern anhand einer Sequenz sollen Rückschlüsse auf die versteckten Zustände gemacht werden. Das Problem wird präzisiert durch die Suche nach einer Folge versteckter Zustände q_1, \dots, q_T , deren Wahrscheinlichkeit zusammen mit den Beobachtungen maximiert wird:

$$(q_1, \dots, q_T) = \operatorname{argmax}_{(q_1, \dots, q_T)} p(q_1, \dots, q_T, o_1, \dots, o_T) \quad (6.9)$$

$$= \operatorname{argmax}_{(q_1, \dots, q_T)} \prod_{i=1}^T p(o_i|q_i)p(q_i|q_{i-1}) \quad (6.10)$$

Hiermit wird die Wahrscheinlichkeit des Pfades selbst sowie der Beobachtungen maximiert. Die Berechnung gleicht dem Alpha-Beta Algorithmus, aber der Summen-Operator wird durch *argmax* ersetzt. Der *Viterbi-Algorithmus* löst diese Berechnung ebenfalls in der Komplexitätsklasse $O(Q^2T)$.

Training mit dem Baum Welch Algorithmus

Die komplizierteste Berechnung ist die Parametrierung eines HMMs anhand vorhandener Beobachtungssequenzen. Es seien (O_1, \dots, O_K) Trainingssequenzen von potenziell verschiedener Länge, dann ist ein HMM λ gesucht, welches diese Sequenzen bestmöglich generieren kann. Diese Fragestellung wird als Optimierungsproblem formuliert, indem die A-Posteriori-Wahrscheinlichkeit maximiert wird:

$$\lambda = \operatorname{argmax}_{\lambda} \prod_{i=1 \dots K} p(O_i|\lambda) \quad (6.11)$$

Hierbei wird das Prinzip *Expectation Maximization* angewandt, welches davon ausgeht, dass (O_1, \dots, O_K) eine repräsentative Stichprobe der von λ erzeugten Sequenzen sind. Der *Baum Welch Algorithmus* ist das geläufigste Verfahren zur Berechnung dieses Optimums; es ist ein *Greedy-Verfahren* und optimiert in jedem Iterationsschritt ähnlich *Gradient Descend* die Parameter des gesuchten HMMs unter Verwendung des *Alpha-Beta-Algorithmus*.

6.1.2 Modellierung eines Zeitschritts

Im Allgemeinen ist die aufgenommene Datenrate höher, als Informationen für Klassifikationszwecke enthalten sind. Akustische Daten werden z.B. häufig mit der Frequenz $\omega = 44\text{kHz}$ aufgenommen; für die Spracherkennung ist jedoch nur die Tonhöhe und Abfolge der Phoneme entscheidend, deren Artikulationsgeschwindigkeit etwa $\omega' = 2 \dots 16\text{Hz}$ beträgt [SSP09]. Ähnlich dem Prinzip der Merkmalsextraktion kann die Abtastrate also explizit modelliert und reduziert werden, um die Dichte an relevanter Information zu steigern.

Indem die Daten auf eine Abtastfrequenz ω_λ reduziert werden, wird eine Teilsequenz S der Länge $T = \omega_\lambda^{-1}$ in einen einzelnen, diskreten Zeitschritt t komprimiert. Um möglichst die relevante Information zu extrahieren, wird der Beobachtungsvektor o_t auf den Mittelwert von S gesetzt. Zusätzlich können Regressionskoeffizienten und Standardabweichungen als weitere Merkmale hinzugefügt werden. Die Sequenzen S können sich auch überlappen, um gegenseitige Einflüsse zu berücksichtigen - dies gleicht praktisch einer Glättung der Daten (Abb. 6.2).

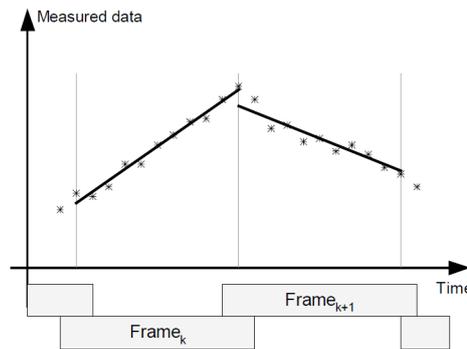


Abbildung 6.2: Segmentierung in einzelne, überlappende Frames, die jeweils einem Zeitschritt eines HMMs entsprechen. Die lineare Regression sowie die Standardabweichung je Frame können als zusätzliches Merkmal verwendet werden [Mit04].

Zustandsverweildauern eines HMMs

Der zeitliche Fortschritt in einem HMM ist primär durch die Zustandsübergangswahrscheinlichkeiten bestimmt. Ohne Berücksichtigung von Beobachtungen ist die A-Priori-Zustandsverweildauer D eines Zustands q geometrisch verteilt, mit Parameter $p = p(q \rightarrow q)$. Der Erwartungswert $\mathbb{E}(D)$ berechnet sich wie folgt:

$$p(D = d) = p^d(1 - p) \quad (6.12)$$

$$\mathbb{E}(D) = p/(1 - p) \quad (6.13)$$

Weil die geometrische Verteilung nicht immer zu den erwarteten Zustandsverweildauern passt, gibt es verschiedene Ansätze, D variabler zu modellieren:

- Globale zeitliche Streckung durch Variation der Abtastsequenz ω_λ . Eine solche Streckung wirkt sich global auf alle Zustände aus, Verweildauern in einzelnen Zuständen können dadurch nicht variabler gestaltet werden.
- Erweiterung des HMMs durch explizit modellierte, z.B. Gauss-verteilte Verweildauern $p(D)$. Diese werden im Viterbi und Alpha Beta Algorithmus berücksichtigt, wodurch jedoch die Komplexität stark ansteigt und das Training inhaltlich erschwert wird [Rab89].
- Simulation einer beliebigen Verteilung der Verweildauer durch Hintereinanderschalten mehrerer Zustände (Abb. 6.3). Die Zustände besitzen dieselben Beobachtungswahrscheinlichkeiten, was als *Parameter Tying* bezeichnet wird [Bil06].

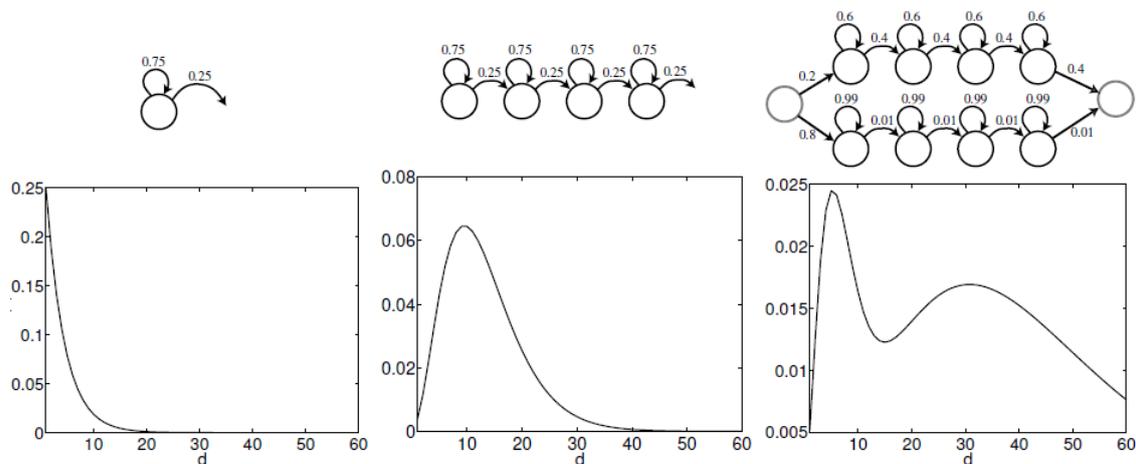


Abbildung 6.3: Zustandsverweildauern in einem HMM. 1) Die erwartete Dauer ist geometrisch verteilt. 2, 3) Durch Hintereinander- oder Parallelschalten mehrerer Zustände mit identischen Beobachtungswahrscheinlichkeiten können fast beliebige Verweildauern modelliert werden [Bil06].

Interpretation

Die Modellierung des zeitlichen Fortschritts und die resultierende Verweildauer $E(D)$ hat einen hohen Einfluss auf die Güte der Anwendung eines HMMs. Ist $E(D)$ zu hoch, wird ein Zustandswechsel sehr unwahrscheinlich - im Extremfall bleibt das HMM dauerhaft in einem Zustand, womit die zeitliche Information überhaupt nicht genutzt wird. Ist $E(D)$ hingegen zu niedrig, verliert der Zustand an Einfluss, da er im Extremfall nur je einen Zeitschritt aktiv ist (Kapitel 8.9.3). Die verbleibende Sequenz muss dann entweder auf andere Zustände mit wiederum hoher Verweildauer abgebildet werden, oder Zustände wechseln sich häufig ab und können damit nicht spezialisiert sein.

Mit den zuvor genannten Ansätzen können die Zustandsverweildauern zwar beliebig gestaltet werden, der Komplexitätsgrad des Modells steigt jedoch, und die Parameter werden schwieriger zu trainieren. In diesen Untersuchungen wurde deshalb strukturell die Basis-Form verwendet, unter systematischer Modellierung der erwarteten Verweildauern

(Kapitel 7.4.2). Als Verbesserung wurde der Zeitschritt an den Wert zusätzlicher Beobachtungen gekoppelt, wodurch sich eine präzise Modellierbarkeit ergab (Kapitel 7.4.3).

6.1.3 Ensemble-HMMs zur Klassifikation

HMMs sind per definitionem *generative Modelle*, sie erzeugen Datensequenzen oder analysieren deren Wahrscheinlichkeit. Bei der Klassifikation hingegen soll anhand gegebener Daten die Klasse geschätzt werden. Der übliche Ansatz, ein diskriminatives Framework zu erstellen, ist die Verwendung eines Ensembles von HMMs [Rab89]: Auf jede Klasse c wird ein HMM λ_c spezialisiert, welches mit Sequenzen genau dieser Klasse trainiert wird (Abb. 6.4).

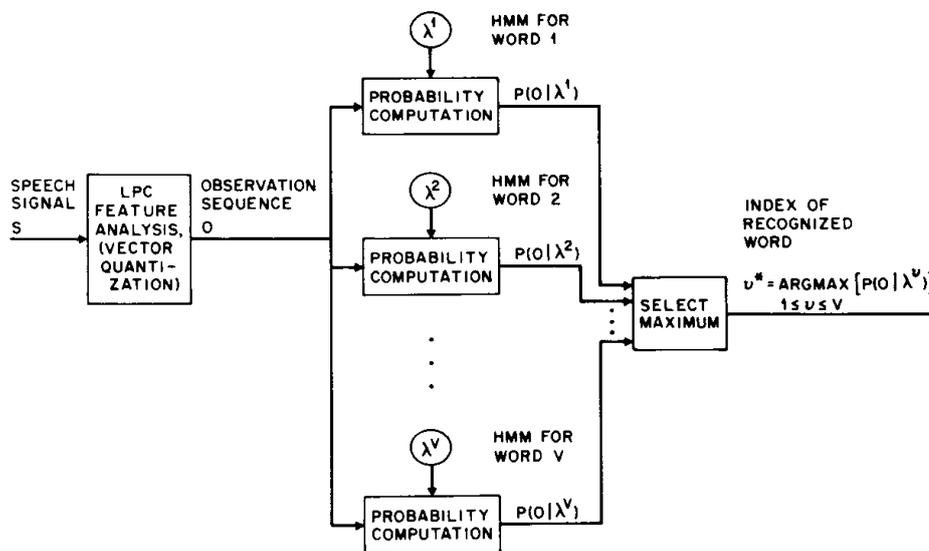


Abbildung 6.4: Worterkennung mittels eines Ensemble von HMMs. Mit den Repräsentanten jeder Klasse wird jeweils ein Modell trainiert. Für die anschließende Klassifikation wird die A-Posteriori-Wahrscheinlichkeit gegeben jedes einzelnen Modells berechnet und der das HMM mit der größten Schätzung verwendet [Rab89].

Zur Klassifikation einer Sequenz $O = o_1, \dots, o_T$ werden zunächst mit allen HMMs die A-Posteriori-Wahrscheinlichkeiten $p(O|\lambda_c)$ berechnet, mit dem Viterbi- oder Alpha-Beta-Algorithmus. Die geschätzte Klasse entspricht dem Modell, welches $p(O|\lambda_c)$ maximiert:

$$\hat{\Psi}(O) = \operatorname{argmax}_c \{p(o|\lambda_c)\} \quad (6.14)$$

Weil seltene Sequenzen aufgrund des *Expectation Maximization*-Modelltrainings auch geringe Wahrscheinlichkeiten besitzen, ist $\hat{\Psi}(O)$ also stark von den A-Priori-Wahrscheinlichkeiten $p(\lambda_c)$ der Trainingsklassen abhängig. Zudem wird meist eine saubere Klassenverteilung mit $\sum_c p(c|O) = 1$ gesucht. Beide Forderungen werden durch die *Bayes-Formel* gelöst [BED08]:

$$p(\lambda_c|o) = p(o|\lambda_c) \frac{p(\lambda_c)}{p(o)} \quad (6.15)$$

$$= p(o|\lambda_c) \frac{p(\lambda_c)}{\sum_c p(o|\lambda_c)p(\lambda_c)} \quad (6.16)$$

Bei der Verwendung von Ensemble-HMMs muss betont werden, dass zu jeder Sequenz nur eine Klassenschätzung entsteht - ein unendlicher Datenstrom wie z.B. die Fahrdaten mit abwechselnden Klassen muss hierfür in Teilsequenzen zerlegt werden.

In verschiedenen Untersuchungen wird diese Zerlegung anhand eines Neuheitskriteriums durchgeführt, welches eine neue Sequenz beginnt, sobald die Daten eine starke Abweichung der vorherigen Daten zeigen [TVL06, VPS07]. In Kapitel 7.4.3 wird für dieses Problem eine neuartige alternative Lösung erarbeitet, basierend auf absoluten Zeitpunkten in sequentiellen Daten, die zusammen mit der subjektiven Kritikalität besonders für die Fahrerassistenz geeignet ist.

6.1.4 Diskriminatives Training mit Mutual Information

Bei der Klassifikation mit HMMs ist eine maximale Posteriori-Wahrscheinlichkeit der Datensequenzen, wie sie durch *Expectation Maximization* trainiert wird, meistens nur von sekundärem Interesse: Vielmehr interessiert das Ensemble von Modellen, welches eine möglichst hohe Klassifikationsrate erzielt. Mutual Information bietet eine Möglichkeit, das Training entsprechend zu gestalten.

Gegeben seien Sequenzen $O_i, i = 1 \dots N$ und zugehörige Klassen $c_i \in 1 \dots C$. Die Mutual Information MI zwischen den Beobachtungen und der Klasse ist das Optimierungskriterium, hier in einer vereinfachten Form (6.17). Die Verwendung der *Bayes-Formel* erlaubt die Berechnung mit den zu trainierenden HMMs [Kap98]:

$$\text{MI} \approx \sum_{i=1}^N \log \frac{p(O_i, c_i)}{p(O_i)p(c_i)} \quad (6.17)$$

$$= \sum_{i=1}^N \log p(O_i | c_i) - \sum_{i=1}^N \log p(O_i) \quad (6.18)$$

$$= \sum_{i=1}^N \log p(O_i | \lambda_i) - \sum_{i=1}^N \log \sum_{m=1}^C p(O_i, \lambda_m) \quad (6.19)$$

Das Zusammenspiel der beiden Terme ergibt für jede Sequenz O_i eine Maximierung der Posterioriwahrscheinlichkeit genau auf dem HMM λ_i , welches der Trainingsklasse c_i entspricht, und einer Minimierung auf den anderen HMMs. Diese Wahrscheinlichkeiten in Gleichung (6.19) beeinflussen stark den Optimierungsschritt der Trainingsphase und können sehr kleine Werte annehmen. Aus diesem Grund können Parameter während der Modellbildung schnell degenerieren, und es wird eine große Menge an Trainingsdaten benötigt [Kap98, Bis06], die aufgrund der Komplexität der Fahrdaten nicht vorliegen kann.

In Voruntersuchungen wurde diskriminatives Training von HMMs für die Erkennung der Deeskalationsmanöver verwendet. Aufgrund des komplizierteren Trainingsverfahrens und schlechter Resultate wurden diese Experimente jedoch eingestellt: Die erstellten Modelle waren häufig degeneriert und erzielten nur schlechte Klassifikationsergebnisse [Ste11].

Ein weiterer Grund, diskriminatives Training in der vorliegenden Arbeit nicht weiter zu verwenden, liegt an der Ensemble-Klassifikation: Wie auch bei regulärem Training geschieht die Klassifikation stets für eine gesamte Sequenz, weil nicht die aktiven Zustände, sondern die resultierende A-Posteriori-Wahrscheinlichkeit verwendet wird. In dieser Arbeit wurde die Bedeutung einzelner Zustände herbeigezogen, wofür mit Klassifikations-HMMs eine andere Form der Modellierung gefunden wurde (Kapitel 7.2).

6.1.5 Bedeutung der Zustände

Weil die Zustände q_i eines HMMs jeweils anhand ihrer Beobachtungswahrscheinlichkeiten auf eine bestimmte Datenmenge spezialisiert sind, besitzen sie eine (zumindest indirekte) Bedeutung. Bei der Klassifikation mit Ensemble-HMMs spielt die Bedeutung keine Rolle, da die geschätzte Klasse nur von der Posteriori-Wahrscheinlichkeit der gesamten Sequenz abhängt. Unter Umständen möchte man jedoch die Zustände gezielt mit zusätzlichen Informationen anreichern oder deren Aktivitäten genau analysieren.

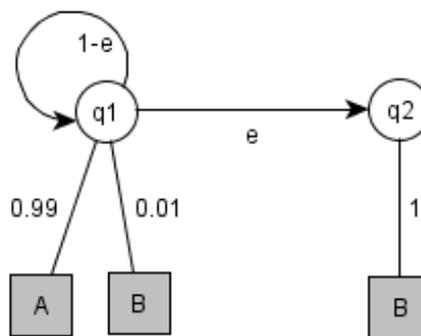


Abbildung 6.5: Ein einfaches HMM zur Analyse von Sequenzen mit den diskreten Beobachtungen $O = \{A, B\}$. Zustand q_1 kann fast nur das Symbol A , q_2 ausschließlich das Symbol B beobachten.

Eine solche Betrachtung verdeutlicht den Baum Welch Algorithmus: Zustandsübergänge, die in den Trainingsdaten selten auftauchen, erhalten im Modell automatisch eine geringe Wahrscheinlichkeit. Anhand des Beispiels in Abb. 6.5 wird dies verdeutlicht: Wird die Sequenz $S = AAAB$ eingegeben, so würde man gemäß der Beobachtungswahrscheinlichkeiten und damit Spezialisierungen die versteckte Zustandsfolge q_1, q_1, q_1, q_2 erwarten - schließlich ist q_2 genau auf die Beobachtung B spezialisiert. Dies ist die *diskriminative* Sichtweise.

Sobald jedoch die Transitionswahrscheinlichkeit mit $e < 0.01$ zu gering wird, verbleibt das HMM während der gesamten Sequenz in q_1 , was den *generativen Charakter* des Modells widerspiegelt: Das Modell muss also mit Trainingssequenzen erstellt worden sein, in denen die Teilfolge AB entsprechend selten vorkam. Zur Analyse solcher Sequenzen ist das HMM damit ungeeignet.

Die A-Priori-Wahrscheinlichkeiten der Beobachtungen und Sequenzen wirken sich also per Konstruktion stark auf die Modellstruktur und die resultierenden Aktivitäten der

Zustände aus; dies muss berücksichtigt werden, wenn die Zustandsaktivitäten im Detail betrachtet werden (Kapitel 7.2). Damit einzelne Transitionen und damit auch Zustände nicht zu unwahrscheinlich werden, werden in Kapitel 7.4 Maßnahmen für eine ausgeglichene Modellstruktur erarbeitet.

6.1.6 Zusammenfassung

Hidden Markov Modelle sind ein vielfach verwendetes Verfahren mit Ursprung in der Sprachverarbeitung [DHS01]. Sie wurden bereits auf diverse Problemstellung der Klassifikation von Sequenzen übertragen, zum Beispiel zur Fahrmanövererkennung [TVL06, Mit05] oder Gestenerkennung [BOP97]. Mit der üblichen Verwendung von Ensemble-HMMs kann zu jeder Sequenz nur eine Klasse geschätzt werden, was durch Unterteilung langer Sequenzen gelöst werden muss.

Die generative Eigenschaft von HMMs erlaubt eine Modellierung der Sequenzen, statt nur deren Klassifikation. Da während der Modellbildung jedoch die Auftretenshäufigkeiten einzelner Sequenzen und Zustände mit gelernt wird, können bei ungleich verteilten Trainingsdaten auch schlechte Modelle entstehen. Damit einzelne Zustände eine Bedeutung erhalten und somit auf bestimmte Beobachtungen und Bedeutungen spezialisiert werden können, werden Maßnahmen für eine ausgeglichene Modellstruktur benötigt.

6.2 Random Forest

Der Random Forest Algorithmus wurde 2001 von Breiman [Bre01] vorgestellt und ist damit verhältnismäßig neu. Mit seiner Effizienz und Stabilität auch in hochdimensionalen Merkmalsräumen konnten bereits in vielen Gebieten der Mustererkennung bestmögliche Resultate erzielt werden [CKY08].

Zum Training eines Modells werden unter Berücksichtigung von *Occam's Razor* mehrere Entscheidungsbäume erstellt. Durch geschickte Randomisierung unterscheiden sich die Bäume, was zur Stabilität der Klassifikation beiträgt. Die folgenden Abschnitte stellen die einzelnen Schritte vor.

6.2.1 Occam's Razor

Grundlage vieler Verfahren des Maschinellen Lernens ist *Occam's Razor*, eine Regel, welche nach dem englischen Philosophen Wilhelm von Ockham benannt wurde. Vereinfacht besagt sie, dass für die Erklärung einer Beobachtung stets die einfachste Hypothese zu bevorzugen ist [RN02].

Uhrzeit	Qualität	Preis	Bewertung
14:00	gut	teuer	positiv
20:00	mittel	teuer	negativ
18:00	gut	preiswert	positiv
16:00	mittel	preiswert	positiv
15:00	schlecht	teuer	negativ
15:00	gut	preiswert	positiv

Tabelle 6.2: Beispiel für einen Klassifikationsdatensatz, hier eine Variante des bekannten Restaurantbesuchers. Für jedes Restaurant sind die Uhrzeit des Besuchs, Qualität und Preis des Essens sowie die zu erlernende Bewertung gespeichert.

Tabelle 6.2 veranschaulicht das Prinzip anhand eines Beispiels. Einem Restaurantbesucher sollen neue Restaurants empfohlen werden, wofür seine Bewertungskriterien erkannt werden sollen. Formal wird also ein Modell benötigt, welches die Bewertung des Besuchers anhand der drei Beobachtungen Uhrzeit, Qualität und Preis schätzt. Hierfür lassen die Beispieldaten u.A. die folgenden zwei Hypothesen zu:

1. Bei guter Qualität *oder* günstigen Preisen wird das Restaurant positiv bewertet, ansonsten negativ.
2. Bei guter Qualität *und* günstigen Preisen, sowie zwischen 16:00 und 19:00 oder um 14:00, fällt die Bewertung positiv aus, ansonsten negativ.

Beide Modelle sagen für die Beispieldaten die richtige Bewertung voraus. Offensichtlich beinhaltet das erste Modell eine einfache, plausible Logik und kann für weitere Empfehlungen verwendet werden. Das zweite Modell erlernt jedoch nur den Spezialfall gute Qualität und günstige Preise; darüberhinaus würde es abhängig von der Uhrzeit entweder alle oder kein Restaurant empfehlen. Es modelliert also nicht die wahren Bewertungskriterien, sondern erklärt umständlich mittels eines redundanten Merkmals die Beispieldaten.

Occam's Razor findet sich also in der ersten, nicht aber der zweiten Hypothese wieder. Übertragen auf Maschinelles Lernen wird klar, wie wichtig die Bildung der korrekten und universellsten Hypothese durch Identifizierung der relevanten Merkmale ist. Wird dieses Ziel verfehlt, entsteht eine Hypothese, die meist nur auf den Trainingsdaten gültig ist; in der Praxis kommt dieser Effekt häufig vor und wird als *Overfitting* bezeichnet.

6.2.2 Konstruktion eines Entscheidungsbaums

Als Konstruktionsalgorithmus für einen einzelnen Entscheidungsbaums wird das Verfahren *Classification and Regression Trees (CART)* eingesetzt, das bereits 1984 von Breiman publiziert wurde [BFOS84]. Anhand von Trainingsdaten wird ein einzelner Entscheidungsbaum mit den folgenden Schritten erstellt:

1. Gegeben ist eine Menge von Trainingsbeispielen $\mathcal{T} = (f_i, c_i)_{i=1\dots N}$ mit Beobachtungen f_i und dazugehörigen Klassen c_i sowie der Wurzelknoten v .
2. Rekursionsabbruch trivial: Falls alle Beispiele zu derselben Klasse c gehören, also $|\{c_i\}| = 1$, ist die Klasse eindeutig und kein weiterer Split nötig.
3. Rekursionsabbruch wegen Nicht-Unterscheidbarkeit: Sind die Indikatoren $|\{f_i\}| = 1$ identisch, ist ein weiterer Split unmöglich. Die empirische Klassenverteilung $p(c|\mathcal{T})$ wird im Knoten gespeichert und später als Konfidenzmaß verwendet.
4. Split: Ansonsten werden ein Indikatorsignal j und ein Grenzwert d gewählt. (j, d) bildet eine orthogonale Ebene im Merkmalsraum. Für beide Halbräume wird je ein neuer Unterknoten v_1, v_2 eingefügt und die Trainingsdaten entsprechend in $\mathcal{T}_1, \mathcal{T}_2$ partitioniert.
5. Die Rekursion wird in beiden Unterknoten v_i mit der jeweiligen Partition \mathcal{T}_i in Schritt 1 fortgesetzt.

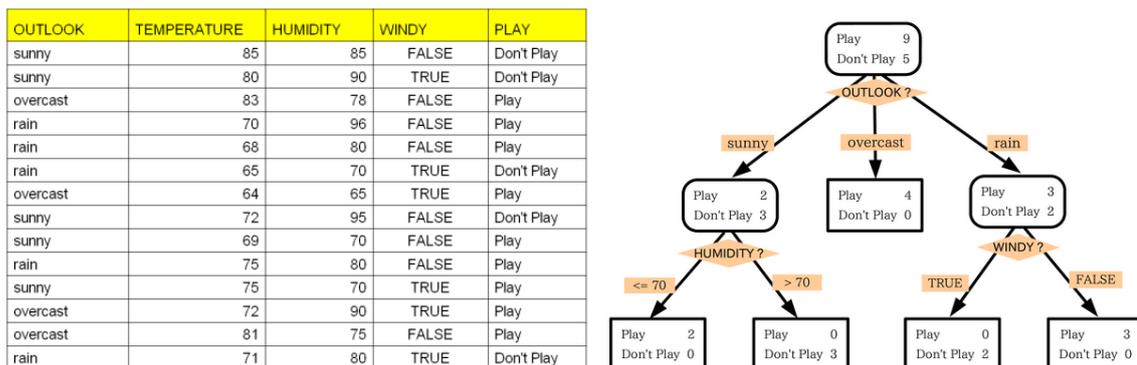


Abbildung 6.6: Golf-Datensatz und ein daraus erstellter Entscheidungsbaum. Links: Der Datensatz besteht aus jeweils vier unabhängigen Indikator-Variablen, anhand derer die abhängige Variable *Play* geschätzt werden soll. Rechts: Der resultierende Entscheidungsbaum. In jedem Knoten ist die empirische Verteilung der zu schätzenden Variable *Play* dargestellt [Enc12].

Die Wahl des Indikatorsignals in Schritt 4 ist entscheidend für die Güte der Prädiktion. Um mit dem Ziel eines einfachen Modells möglichst wenige Knoten einzufügen, wird in jedem Schritt eine Heuristik angewandt: Es wird der Split (j, d) gewählt, welcher die Heterogenität der Klassen minimiert - ausgedrückt durch den *Gini-Index*, *Entropie* oder *Impurity*.

Abb. 6.6 zeigt abschließend den Trainingsdatensatz für das häufig zitierte Golf-Beispiel und einen möglichen, daraus erstellten Entscheidungsbaum.

6.2.3 Randomisierung

Der Random Forest Algorithmus wirkt Effekten des *Overfittings* (Abschnitt 6.2.1) entgegen, indem bei der Merkmalsselektion und der Wahl der Trainingsdaten Randomisierung eingesetzt wird:

1. Durch Anwendung von *Bagging* wird ein Ensemble von Klassifikatoren erstellt, wobei jeder einzelne auf einer randomisierten Teilmenge der Trainingsdaten (*bootstrap selection*) aufgebaut wird [Bre96].
2. In jedem Knoten wird nur eine randomisierte Teilmenge aller Merkmale betrachtet, um den nächsten Split zu finden.

Diese Randomisierung führt nicht nur Stabilität gegen Messfehler und Overfitting ein; die verschiedenen Klassifikationsergebnisse der einzelnen Bäume können gleichzeitig als Konfidenz- oder Wahrscheinlichkeitsmaße interpretiert werden und modellieren damit die Sicherheit der Prädiktion (Abb. 6.7).

6.2.4 Prädiktion

Zur Prädiktion eines Datensatzes f mit dem trainierten Modell $\lambda = (\lambda_1, \dots, \lambda_N)$ werden die einzelnen Entscheidungsbäume λ_i bis in ein Blatt v_i durchlaufen. Die dort gespeicherten Klassenverteilungen $p^{(v_i, \lambda_i)}(c|\mathcal{T})$ werden über das Ensemble von Bäumen gemittelt. Das Ergebnis ist damit formal eine Wahrscheinlichkeitsverteilung

$$p^{\text{RF}}(c|f) = \frac{1}{N} \sum_i p^{(v_i, \lambda_i)}(c) \quad (6.20)$$

Abb. 6.7 verdeutlicht das Prinzip des Random Forest Algorithmus. Die wahren Klassengrenzen - die diagonalen Linien - werden durch orthogonal verlaufende Klassengrenzen angenähert. Dies resultiert aus der sukzessiven Teilung des Beobachtungsraums in Halbräume, so dass die finale Wahrscheinlichkeitsfunktion eine Summe gewichteter Rechteckverteilungen ist. Gleichzeitig sieht man, dass im Bereich der wahren Klassengrenzen die Prädiktion mit Unsicherheit behaftet sind.

6.2.5 Zusammenfassung

Durch geschickte Randomisierung erzielt Random Forest in vielen Domänen der Mustererkennung gute Resultate: Die resultierende Wahrscheinlichkeitsverteilung $p(c|o)$ entspricht einer gewichteten Summe von Rechteckverteilungen, durch die beliebige Verteilungen approximiert werden können. Durch die Einfachheit der resultierenden, orthogonalen Klassifikationsgrenzen werden besonders hochdimensionale Merkmalsräume stabil beherrscht, jedoch konstruktionsbedingt mit systematischem Fehler. Dieser steht einer Prädiktion

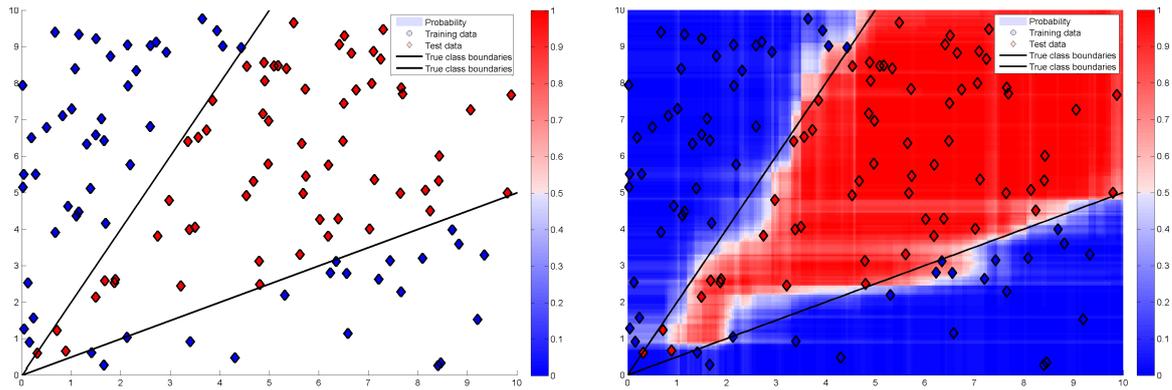


Abbildung 6.7: Prädiktion mit Random Forest. Links: Zweidimensionale Trainingsdaten, deren Klasse (rot oder blau) durch zwei Diagonale im Raum bestimmt sind. Rechts: Ergebnis der Prädiktion auf dem gesamten Wertebereich. Der Farbton stellt die geschätzte Klasse, die Intensität deren Wahrscheinlichkeit dar

kontinuierlicher Daten u.U. im Wege, weshalb für die Kursprädiktion Gegenmaßnahmen durch Transformation der Schätzgrößen ergriffen wurden (Kapitel 5.5).

Es muss betont werden, dass Random Forest wie viele Klassifikatoren (Adaboost, SVM, Regression...) auf Merkmalsvektoren arbeitet und nicht für sequentielle Daten konzipiert ist; Merkmale werden zwar in Kombination miteinander betrachtet, jedoch ist eine zeitliche Entwicklung der Daten nicht direkt abbildbar. Die *Moving Window* Technik bietet einen einfaches Hilfsmittel dazu, welches in unseren Experimenten allerdings wenig erfolgreich war (Kapitel 6.3).

In einzelnen Untersuchungen wurden dazu Erweiterungen vorgeschlagen: Der *Temporal Decision Tree* z.B. verarbeitet in jedem Entscheidungsknoten v die historischen Signalwerte des Alters Δt_v . Im Gegensatz zu den festen Abtastpunkten bei Moving Window hängt hier das Alter Δt_v vom Entscheidungsknoten v ab und kann damit situativ variieren. [CPD03]. Eine Erweiterung auf *Temporal Random Forest* wurde mit Daten der Gestenerkennung untersucht [DV09], es wurden jedoch nur eine geringe Verbesserung bzw. in manchen Settings sogar Verschlechterungen berichtet, was unter Anderem auf die Schwierigkeit des Trainings zurückgeführt wurde.

Zur Vereinfachung des Trainings wurde in dieser Arbeit deshalb die situative Variation der Abtastpunkte durch lineare Streckung untersucht (Kapitel 7.5).

6.3 Moving Window

Im Beispiel von Abb. 6.8 sind sequentielle Daten gezeigt, deren Klassifikation vom zeitlichen Verlauf des Indikatorsignals f abhängt. Die Klasse Ψ drückt Steigung oder Gefälle des Indikatorsignals aus:

$$\Psi(t) = \begin{cases} a & f(t) \leq f(t-1) \\ b & \text{sonst} \end{cases} \quad (6.21)$$

Ein Klassifikator der Form $\hat{\Psi}(f(t))$, der zu jedem Zeitpunkt t nur die aktuellen Werte erhält, kann die Klasse unmöglich erlernen, da die Information der Steigung in dem isoliert betrachteten Wert $f(t)$ nicht enthalten ist.

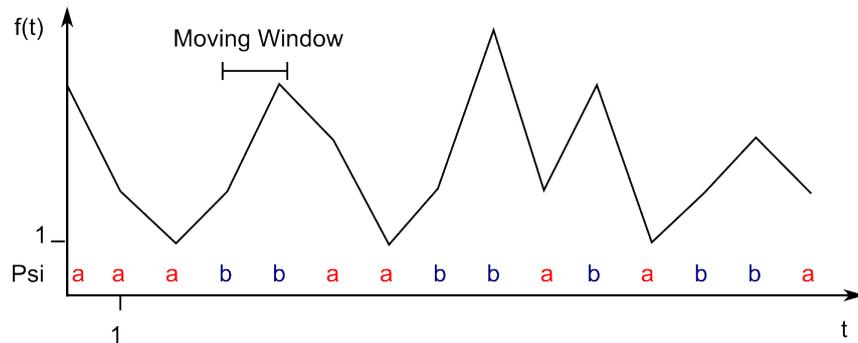


Abbildung 6.8: Sequentieller Datensatz. Die Klasse gibt wieder, ob die Werte zuletzt gefallen (a) oder gestiegen (b) sind. Ein *Moving Window* mit den Abtastpunkten $h = (-1, 0)$ erfasst den vorhergehenden und aktuellen Wert des Signals f .

Mit der *Moving Window* Technik wird ein Zeitfenster an den Zeitpunkt t gelegt, und alle in diesem Fenster enthaltenen Funktionswerte werden dem Ursprungssignal in Form zusätzlicher Dimensionen hinzugefügt [Die02, Mit04]. Formal sei $h = (h_1, \dots, h_H)$ ein Vektor der Abtastpunkte, bestehend aus relativen Zeitangaben, dann entsteht die Sequenz

$$S_h(t) = (f(t + h_1), \dots, f(t + h_H)) \quad (6.22)$$

aus der Erweiterung des Signals f um die historischen Zeitpunkte h . Für das Beispiel kann anhand von $h = (-1, 0)$ offensichtlich eine perfekte Schätzfunktion $\hat{\Psi}(S_h(t))$ gelernt werden: Alle relevanten Informationen sind nun in der Sequenz $S_h(t)$ enthalten.

Für aufeinanderfolgende Zeitpunkte entstehen mit *Moving Window* überlappende Sequenzen (Abb. 7.2) - dies wird in Kapitel 7.1 durch den neuen Ansatz der *Growing Windows* umgangen.

6.4 Bayes-Fehler

Der Bayes-Fehler liefert eine Abschätzung, wie gut ein beliebiger Klassifikator auf einem Datensatz mit bekannten Labels maximal werden kann. Klassifikationsfehler liegen nämlich nicht zwangsläufig beim Klassifikator: Auch kann es sein, dass die Indikatorsignale fehlerhaft oder die Labels ungenau vergeben sind. Abb. 6.9 stellt das Prinzip des Bayes-Fehlers anhand eines eindimensionalen Klassifikationsproblems dar.

Die bedingten Wahrscheinlichkeitsverteilungen der Beobachtungen werden als bekannt angenommen, also $p(x|\omega_1), p(x|\omega_2)$. Falls beide Klassen dieselbe A-Priori-Wahrscheinlichkeit besitzen und die Kosten einer Fehlklassifikation symmetrisch sind, ist die optimale Klassifikation

$$\Psi'(x) = \operatorname{argmax}_i p(x|w_i) \quad (6.23)$$

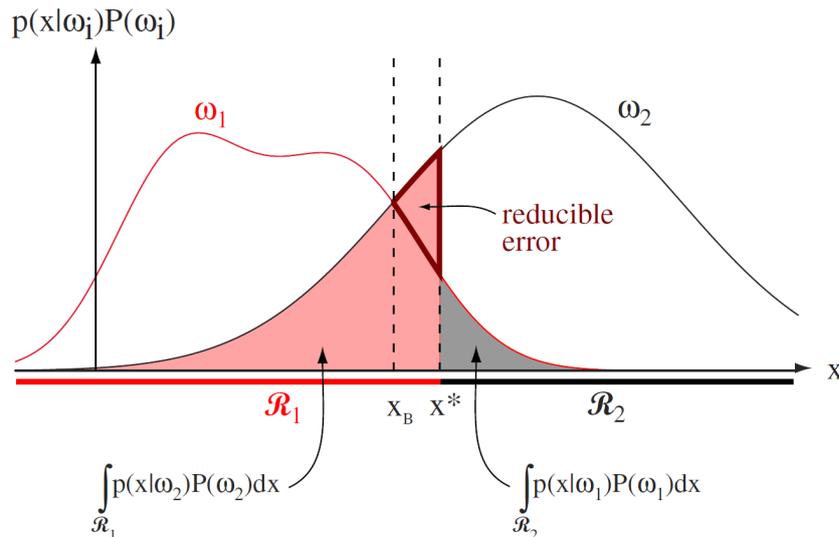


Abbildung 6.9: Bayes-Fehler für die Klassifikation anhand eines eindimensionalen Merkmals x . Da die Klassen sich überlappen, ist eine perfekte Klassifikation nur aufgrund dieses Merkmals nicht möglich. Für die Entscheidungsgrenze x^* entspricht die rot und grau kolorierte Fläche dem erwarteten Klassifikationsfehler. Die Grenze x_B wäre optimal, da sie diesen Klassifikationsfehler minimiert und so den Bayes-Fehler erreicht [DHS01].

Diese Klassifikation ist optimal in dem Sinne, dass sie das erwartete Risiko minimiert, definiert als die Kosten der erwarteten Fehlklassifikationen [DHS01]. Der erwartete Fehler wird durch die falsch klassifizierte Datenpunkte bestimmt. Er berechnet sich also zu

$$E = \int p(\Psi'(x) \neq \Psi(x)) dx \quad (6.24)$$

Anwendung in der Praxis

Die Berechnung der bedingten Beobachtungswahrscheinlichkeiten $p(x|w_i)$ ist nicht trivial, vor allem wenn die Beobachtungen mehrdimensional sind. Dies läuft auf das bekannte Problem der Kerndichteschätzung hinaus [Par62], wofür verschiedene Lösungen existieren. In unseren Arbeiten wurden Histogramme verwendet, was einer gewichteten Summe von Rechteckverteilungen entspricht; sie können leicht bestimmt werden und konvergieren asymptotisch gegen die wahre Verteilung. Je Merkmalsdimension wurden $n = 30$ Rechteckverteilungen verwendet, um eine hinreichend feine Abtastung des Merkmalsraums bei gleichzeitig genügend Datenpunkten zu erzielen.

Ein großer Nachteil der Histogramme ist die hohe Anzahl von Parametern und die resultierende Nicht-Stetigkeit der Klassifikation; zudem funktionieren die Histogramme in Bereichen mit wenigen Datenpunkten sehr schlecht. Eine bekannte Lösung sind *Parzen Windows*, wobei Gaussverteilungen statt Rechteckverteilungen verwendet werden, die sich in Folge weich überlappen [DHS01, HTF01].

6.5 Bestimmung der Abtastfrequenz

Bei der Verwendung zeitlicher Daten allgemein stellt sich früh die Frage, mit welcher Datenrate diese analysiert werden. HMMs und die Moving Window Technik betrachten meist nur diskrete Zeitpunkte, so dass eine geeignete Abtastfrequenz ω gefunden werden muss.

Zur Wahl von ω gibt es keine universelle Lösung - nach dem *Nyquist-Shannonschen Abtasttheorem* muss aber mindestens die doppelte Frequenz der untersuchten Information gewählt werden, damit keine Aliasing-Effekte auftauchen. Um zusätzlich die Amplitude zu erfassen, müssen Abtastpunkte sowohl bei den Nulldurchgängen als auch auf den Maxima und Minima liegen; die vierfache Nyquist-Frequenz wird also benötigt [Uns00].

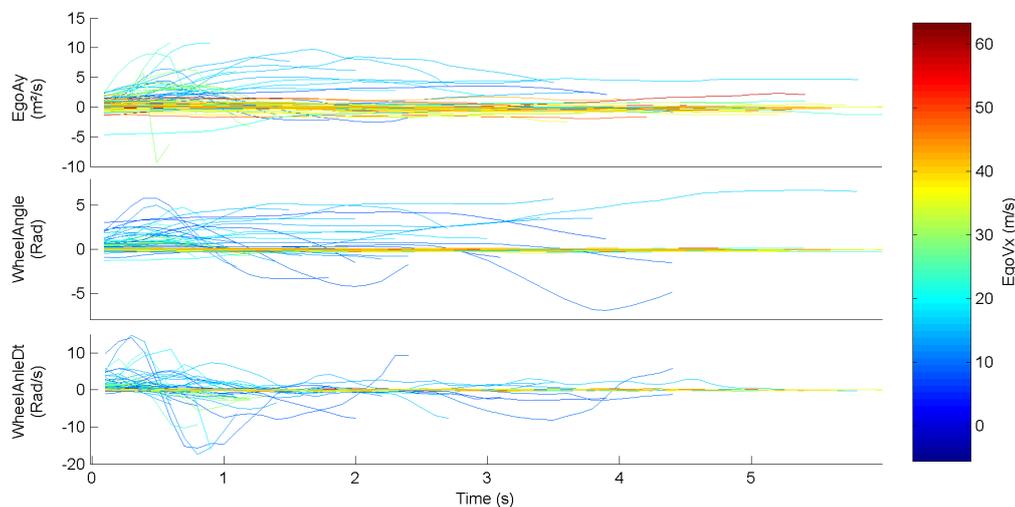


Abbildung 6.10: Zeitlicher Verlauf relevanter Signale während 104 Ausweichmanövern nach links. Von oben nach unten: Querbeschleunigung, Radwinkel und Radwinkelgradient. Die Fahrzeuggeschwindigkeit f_{EgoVx} wurde zur Einfärbung der Signalverläufe verwendet.

Für die vorliegenden Daten wurde eine Heuristik angewandt: Die untersuchte Fahrerintention macht sich besonders in den Signalen erkenntlich, die vom Fahrer direkt manipulierbar sind, also Lenkrad- und Pedalstellung. Abb. 6.10 zeigt Ausweichszenen nach links und den Verlauf dreier typischer Signale der Querbewegung.

Zum Zeitpunkt $t = 0.5s$ ist für einige Sequenzen das Maximum einer Sinus-förmigen Bewegung zu sehen, die durch das Steuerverhalten des Fahrers erklärt werden und für Ausweichbewegungen typisch ist (Kapitel 2.4). Mit steigenden Geschwindigkeiten verringern sich sowohl die Frequenz als auch Amplitude der Signale - die Fahrer steuern also langsamer und mit kleineren Bewegungen. Die schnellsten Sequenzen weisen eine ungefähre Dauer von $T = \lambda = 2s$ auf, die vierfache Nyquist-Frequenz beträgt also $\omega = 2Hz$ bzw. $0.5s$. Diese Bedutung stimmt mit der in [Mit05] genannten Filterfrequenz überein (Kapitel 2.7).

6.6 Zusammenfassung

Die Prinzipien der Hidden Markov Modelle und des Random Forest Algorithmus unterscheiden sich von Grund auf: HMMs sind für die Modellierung zeitlicher Zusammenhänge in Eingabesequenzen konzipiert. Die einzelnen Datenpunkte werden meist durch Gaussian Mixtures modelliert, wobei Schwierigkeiten des Overfittings und degenerierten Komponenten entstehen können.

Random Forest verarbeitet keine zeitliche Daten, kann aber durch die Randomisierung auch mit hochdimensionalen oder schlecht gemessenen Daten umgehen. Hilfsmittel wie Moving Window ermöglichen es, zeitliche Information bedingt zu verwenden. Beide Prinzipien könnten sich aufgrund dieser Eigenschaften also perfekt ergänzen, um zeitliche Daten hoher Dimensionalität zu verarbeiten.

Der Bayes-Fehler dient der Vorab-Abschätzung des Klassifikationsfehlers; er wurde zudem verwendet, um die Relevanz einzelner Merkmale und deren Historie zu bewerten.

7 Methodik

Im methodischen Fokus dieser Arbeit werden *Hidden Markov Modelle* mit *Random Forest* kombiniert: Die Vorteile von HMMs, zeitliche Abläufe modellieren zu können, werden zusammen mit der Stärke von Random Forest ausgenutzt, komplexe und hochdimensionale Wahrscheinlichkeitsverteilungen lernen zu können (Kapitel 6). Für diese Anwendung als Kerndichteschätzer wird Random Forest auf gewichteten Labels trainiert.

Weil HMMs ursprünglich für die Analyse gesprochener Wörter, also abgeschlossener Sequenzen konzipiert sind, ist für den kontinuierlichen Datenstrom der Fahrerassistenzdaten zunächst eine strukturelle Anpassung erforderlich. Dazu werden Ensemble-HMMs äquivalent in ein einzelnes Klassifikations-HMM überführt. Es ermöglicht eine variable Modellierung, Verbesserung und Analyse der Modellstruktur (Abb. 7.1).

Zur Einbindung historischer Daten im alleinstehenden Random Forest Algorithmus wird eine Erweiterung des *Moving Window* Prinzips erarbeitet, welche die variierende Ausführungszeit der Sequenzen - speziell menschlicher Handlungen - abbildet. Die Anwendung zur Kursprädiktion ergibt sich schließlich aus der Kombination der geschätzten Parameter mit dem Kursmodell.

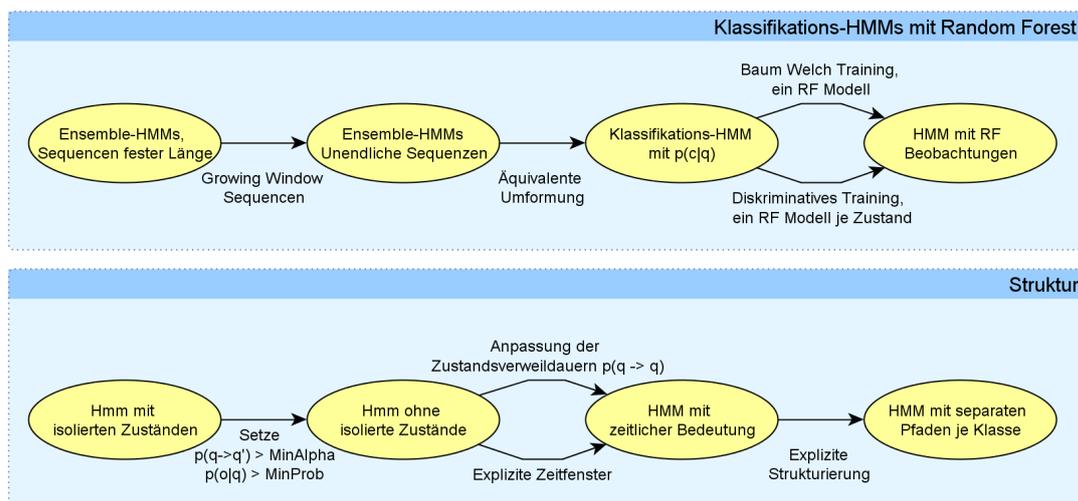


Abbildung 7.1: Überblick der Methoden. Oben: HMMs werden für die Klassifikation unendlich langer Datensequenzen weiterentwickelt. Die Beobachtungswahrscheinlichkeiten werden durch Random Forest anstelle von Gaussian Mixtures berechnet. Unten: Strukturelle Anpassungen werden explizit für das Anwendungsgebiet Fahrerassistenz erarbeitet. Die Maßnahmen sind größtenteils auch unabhängig voneinander anwendbar.

7.1 Growing Window Sequenzen

Sequentielle Daten liegen häufig als (unendlicher) Datenstrom vor, der sich aus verschiedenen Klassen zusammensetzt. So bestehen z.B. gesprochene Sätze aus verschiedenen Wörtern und Fahrsequenzen aus Fahrmanövern. Weil die meisten Klassifikatoren jeweils eine Klasse je gesamter Sequenz schätzen, muss ein solcher Datenstrom in Teilsequenzen zerlegt werden. Dabei gilt es, werden möglichst kurze Sequenzen zu finden, die dennoch alle relevanten Informationen enthalten.

Üblicherweise werden mit dem *Moving Window* Prinzip überlappende Sequenzen fester Länge erstellt (Kapitel 6.3). Das nun vorgestellte *Growing Window* Prinzip vermeidet die entstehende Redundanz [Dit09]. Es ist eine Vorstufe zur Einführung der Klassifikations-HMMs, die völlig ohne eine Unterteilung auskommen.

Training

Durchgängig werden nur kritische Zeitpunkte betrachtet, da ansonsten kein Deeskalationsmanöver definiert ist. Zum ersten Eskalationszeitpunkt haben die Fahrer sich per definitionem für eine Handlung entschieden (Kapitel 5.3), so dass die gesamte kritische Szene genau den relevanten Daten entspricht.

Es seien $\mathbf{S}_{\text{Critical}} = \{(a, b)\}$ die Start- und Endpunkte aller kritischen Szenen. Mit *Moving Window (MW)* und *Growing Window (GW)* ergeben sich die folgenden Mengen von Trainingssequenzen (Abb. 7.2):

$$\mathbf{S}_{\text{Train}}^{\text{MW}} = \{S_t = o_{t-\Delta t} \dots o_t, t = 1 \dots T\} \quad (7.1)$$

$$\mathbf{S}_{\text{Train}}^{\text{GW}} = \{S_{a,b} = o_a \dots o_b, (a, b) \in \mathbf{S}_{\text{Critical}}\} \quad (7.2)$$

Prädiktion

Für die Prädiktion zum Zeitpunkt $t = 1 \dots T$ werden analog zum Training Sequenzen ab dem ersten Eskalationszeitpunkt dieser Szene verwendet; mit fortschreitendem Zeitpunkt t sind dies also genau die Sequenzen wachsender Länge (Abb. 7.2):

$$S_t^{\text{MW}} = o_{t-\Delta t} \dots o_t \quad (7.3)$$

$$S_t^{\text{GW}} = o_a \dots o_t \quad \exists (a, b) \in \mathbf{S}_{\text{Critical}} : t \in [a, b] \quad (7.4)$$

Da jede Sequenz S_{t+1}^{GW} die vorhergehende S_t^{GW} beinhaltet, ist auf den ersten Blick noch Redundanz enthalten. Bei der Klassifikation mit dem Alpha Beta Algorithmus (Kapitel 7.2.2) genügt es jedoch, nur die letzte Sequenz S_T^{GW} zu berechnen: Die resultierenden $\alpha_{t,q}, t_0 \leq t \leq T$ entsprechen genau dem Ergebnis der jeweiligen Teilsequenz S_t .

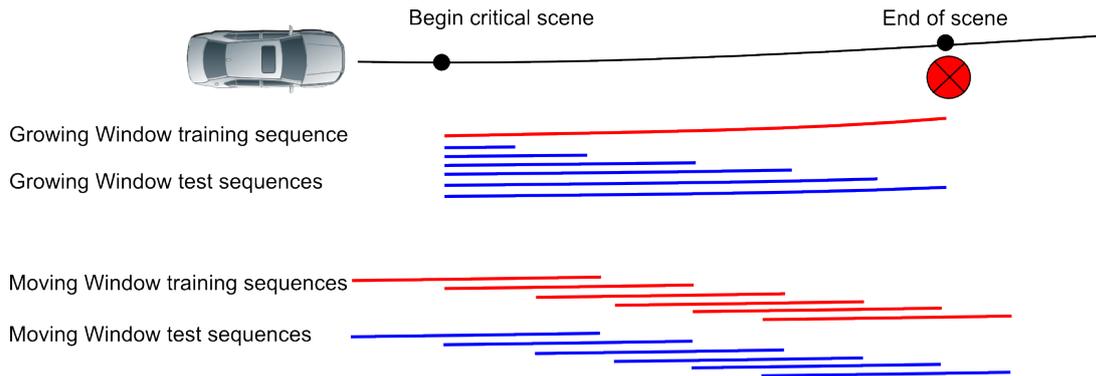


Abbildung 7.2: Unterteilung der Fahrdaten in Teilsequenzen für die Klassifikation und Prädiktion mit Hidden Markov Modellen. Die Moving Window Sequenzen besitzen stets eine konstante Länge, während die Growing Window Sequenzen zum ersten kritischen Zeitpunkt beginnen.

Vorteile

Schon rein informativ unterscheiden sich *Moving* und *Growing Window*: Bei MW wird angenommen, dass die relevanten Indikatoren innerhalb einer konstanten vorherigen Zeitspanne vorlagen; bei GW werden sie ab einem bestimmten Ereignis vermutet, welches als Sequenzbeginn verwendet wird. Für die Wahl dieses Ereignisses ergibt sich als naheliegende Lösung der Klassenbeginn - es entsteht jedoch die Gefahr, frühere Indikatoren zu verpassen. Klassifikations-HMMs adressieren dieses Problem, indem die Gesamtsequenz ohne weitere Unterteilung verwendet wird (Kapitel 7.2).

Mit *Growing Window* Sequenzen kann die Klassifikation besonders effizient berechnet werden: Die α -Variablen können iterativ berechnet werden, so dass je Zeitschritt maximal Q^2 Berechnungen nötig sind (Q Anzahl der HMM-Zustände).

7.2 Klassifikations-HMMs

Als wichtige methodische Veränderung für die Klassifikation mit HMMs wird die bekannte Ensemble-Struktur (Kapitel 6.1.3) äquivalent in ein einzelnes HMM überführt. Mit diesem kann nicht nur für ganze Sequenzen, sondern für jeden einzelnen Zeitpunkt eine Klasse geschätzt werden. Dadurch vereinfacht sich nicht nur die Handhabung der Datensequenzen (Kapitel 7.1), sondern es ist auch eine wichtige Voraussetzung für die folgende Integration des Random Forest Algorithmus.

7.2.1 Konstruktion

Zu einem Klassifikationsproblem mit den Klassen $1..C$ sei ein Ensemble von HMMs $(\lambda_c)_{c=1..C}$ gegeben, \mathbf{O} bezeichnet die Domäne der Beobachtungen. Jedes HMM ist ein

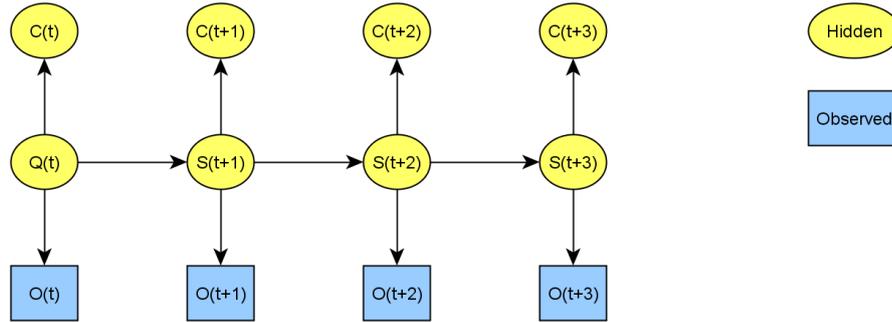


Abbildung 7.3: Klassifikations-HMM: Neben dem versteckten Zustand Q und den Beobachtungen O ist die Klasse C ein weiterer Prozess. Die Klasse hängt vom Zustand ab und ist nur während des Trainings beobachtbar.

Tupel $\lambda_c = (Q^c, \mathbf{O}, \pi^c, A^c, B^c)$ mit den üblichen Komponenten (Tabelle 6.1). Für die Kombination des Ensembles zu einem einzelnen Klassifikations-HMM λ werden alle Zustände miteinander vereint, wobei die Beobachtungen und Transitionen erhalten bleiben (Abb. 7.4). Die π_q^c werden mit der A-Priori-Klassenwahrscheinlichkeit $p(c)$ gewichtet, wodurch das Modell stochastisch gültig bleibt:

$$Q = Q^1 \cup \dots \cup Q^C \quad (7.5)$$

$$\pi(q) = p(c)\pi^c(q) \quad \forall q \in Q^c \quad (7.6)$$

$$b_q(o) = b_q^c(o) \quad \forall q \in Q^c, o \in \mathbf{O} \quad (7.7)$$

$$a_{q,q'} = \begin{cases} a_{q,q'}^c & \text{falls } q, q' \in Q^c \\ 0 & \text{sonst} \end{cases} \quad (7.8)$$

Für die Klassifikation erhält jeder Zustand eine zusätzliche diskrete Wahrscheinlichkeitsverteilung $p(c|q)$. Er wird mit der Klasse des HMMs assoziiert, aus dem er ursprünglich stammt:

$$p(c|q) = \begin{cases} 1 & \text{falls } q \in Q^c \\ 0 & \text{sonst} \end{cases} \quad (7.9)$$

Damit bekommen die Zustände eine neue Bedeutung: Waren sie ursprünglich nur dazu gedacht, um den sequentiellen Zusammenhang von Beobachtungen zu modellieren, so erhalten sie nun die zusätzliche Bedeutung einer vorliegenden Klasse (Abb. 7.3). Über eine Gewichtung kann sogar eine Klassenverteilung modelliert werden.

7.2.2 Klassifikation und Beweis der Äquivalenz

Mit dem Klassifikations-HMM λ resultiert die geschätzte Klasse aus der einfachen Anwendung des Alpha-Beta Algorithmus (Kapitel 6.1.1). Die durch ihn berechneten Zustandswahrscheinlichkeiten $\alpha_{q,t}$ multipliziert mit den jeweiligen Klassenbeobachtungen liefern direkt die geschätzte Verteilung der Klassen.

Dieses Ergebnis stimmt bei der vorherigen Konstruktion mit der Ensemble-Schätzung überein, was sicherstellt, dass das neue Klassifikations-HMM mindestens gleichmächtig

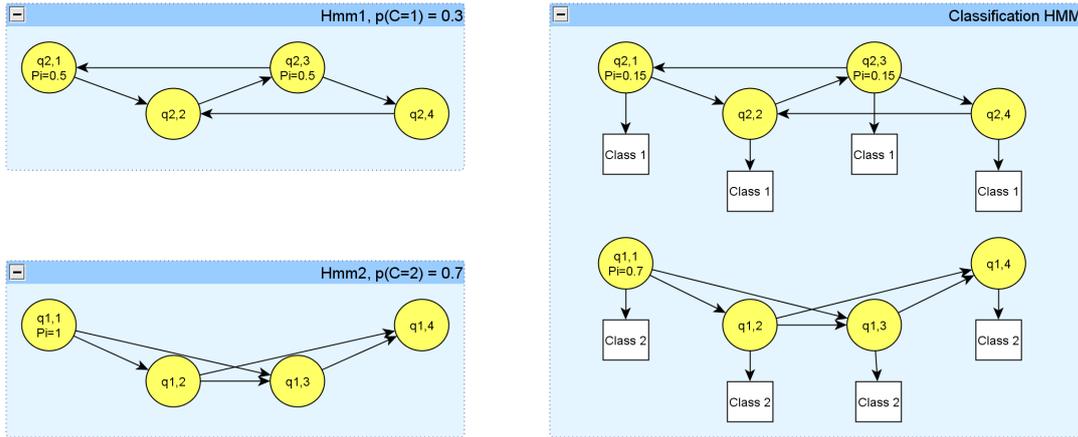


Abbildung 7.4: Kombination eines Ensembles von HMMs zu einem äquivalenten Klassifikations-HMM: Die Startwahrscheinlichkeiten werden mit der ursprünglichen Klassenwahrscheinlichkeit gewichtet, und die Zustände erhalten die Klasse in Form einer zusätzlichen Beobachtung. Durch die neue Struktur können unendlich lange Sequenzen mit unterschiedlichen Klassen problemlos analysiert werden.

sind. Zum Beweis sei $O = o_1 \dots o_T$ die betrachtete Sequenz von Beobachtungen, dann berechnet das HMM-Ensemble $(\lambda_c)_{c=1 \dots C}$ unter Verwendung der Bayes-Formel die Klassenwahrscheinlichkeiten

$$p(c|O) = p(O|c) \frac{p(c)}{p(O)} \quad (7.10)$$

$$= p^c(O) \frac{p(c)}{p(O)} \quad (7.11)$$

$$\sim p^c(O)p(c) \quad (7.12)$$

$$= \sum_{q_1 \dots q_T \in Q^c} p^c(O|q_1 \dots q_T) p^c(q_1 \dots q_T) \pi^c(q_1) p(c) \quad (7.13)$$

Ab Gleichung (7.12) wurde der Normierungsfaktor $p^c(O)$ zur Übersichtlichkeit entfernt, weil er nicht von der geschätzten Klasse abhängt.

Um diese Berechnung nun äquivalent mit dem Klassifikations-HMM λ auszuführen, wird die Summe in (7.13) zunächst auf die Zustände des ursprünglichen HMMs eingeschränkt. Ab (7.15) wird diese Berechnung durch die neuen Klassenwahrscheinlichkeiten $p(c|q)$ (7.9) und Startwahrscheinlichkeiten $\pi(q)$ (7.6) ausgedrückt, wodurch die Einschränkung aufgehoben werden kann:

$$= \sum_{q_1 \dots q_T \in Q^c} p(O|q_1 \dots q_T) p(q_1 \dots q_T) \pi(q_1) \quad (7.14)$$

$$= \sum_{q_1 \dots q_T \in Q} p(c|q_1) \dots p(c|q_T) p(O|q_1 \dots q_T) p(q_1 \dots q_T) \pi(q_1) \quad (7.15)$$

Da in λ zwischen Zuständen q_1, q_2 unterschiedlicher Klassen keine Verbindung besteht, sind alle entsprechenden Pfadwahrscheinlichkeiten $p(q_1 \dots q_2) = 0$. Für alle Pfade $q_1 \dots q_T$ mit von 0 verschiedener Wahrscheinlichkeit stimmen also die Klassenwahrscheinlichkeiten

überein: $p(c|q_1) = \dots = p(c|q_T)$; da diese per Konstruktion nur 0 oder 1 sind, vereinfacht sich die Gleichung weiterhin zu

$$p(c|O) = \sum_{q_1 \dots q_T \in Q} p(c|q_T) p(O|q_1 \dots q_T) p(q_1 \dots q_T) \pi(q_1) \quad (7.16)$$

$$= \sum_{q_1 \dots q_T \in Q} p(c|q_T) p(O, q_1 \dots q_T) \pi(q_1) \quad (7.17)$$

$$= \sum_{q \in Q} p(c|q) p(O, Q_T = q) \quad (7.18)$$

$$= \sum_{q \in Q} p(c|q) \alpha_{q,T} \quad (7.19)$$

Abschließend wird die in (7.12) entfernte Normierungskonstante wieder hergestellt anhand der Bedingung, dass die Klassenwahrscheinlichkeiten zu 1 addieren müssen.

7.2.3 Training mit dem Baum Welch Algorithmus

Die Klassenwahrscheinlichkeiten je Zustand $p(c|q)$ werden als zusätzliche Dimension der Beobachtungen modelliert, wodurch der Baum Welch Algorithmus fast unverändert angewandt werden kann. Die resultierenden *korrigierten Posteriors* $p(o_t|q)$ sind per Konstruktion nur ungleich 0, wenn die Klasse c_t im Zustand q möglich ist. Der Baum Welch Algorithmus trainiert die Beobachtungswahrscheinlichkeiten $p(o|q)$ eines Zustandes also gemäß der Beobachtungen o_t , deren Klasse c_t mit $c(q)$ übereinstimmt.

Neben den Eingangssignalen fließt also zusätzliches Wissen über die Klasse ins Training ein. Die formalen Optimierungskriterien verdeutlichen dies: Üblicherweise wird das HMM λ gesucht, welches die Wahrscheinlichkeiten der Sequenzen $p_\lambda(O)$ maximiert. Bei Klassifikations-HMMs λ^C hingegen wird eine hohe Sequenzwahrscheinlichkeit unter Berücksichtigung der Klasse $p_{\lambda^C}(O, C(O))$ benötigt.

$$\lambda = \operatorname{argmax}_{\lambda} \prod_O \sum_{q_1 \dots q_T \in Q} p_\lambda(O, q_1 \dots q_T) \quad (7.20)$$

$$\lambda^C = \operatorname{argmax}_{\lambda^C} \prod_O \sum_{q_1 \dots q_T \in Q} p_{\lambda^C}(O, C(O), q_1 \dots q_T) \quad (7.21)$$

7.2.4 Vorteile

Mit geringen Modifikationen der Modellierung in Algorithmen wurden Ensemble-HMMs äquivalent in ein Klassifikations-HMM λ umgeformt. Dieses bietet nun verschiedene neue Möglichkeiten:

- Flexible Modellierung: Mit Transitionen zwischen Zuständen unterschiedlicher Klassen können nun auch Klassenübergänge dargestellt werden. Dies kann Unsicherheiten der Schätzung verbessern (Abschnitt 7.4.4).
- Klassifikation langer Sequenzen: Da λ jeden einzelnen Zeitpunkt klassifiziert, können beliebig lange Sequenz mit Klassenwechseln verarbeitet werden; eine Zerlegung in Teilsequenzen entfällt (Abschnitt 7.1).
- Explizite Modellierung der HMM-Struktur: Durch manuelle Initialisierung der Zustandsübergänge kann Expertenwissen über die Verweildauer oder Übergänge von Klassen modelliert werden.
- Training der Klassen: Die Klassenwahrscheinlichkeit $p(c|q)$ können ebenfalls mit dem Baum Welch Algorithmus trainiert oder verbessert werden. Bei einer stark ungleichen Verteilung in den Trainingsdaten geschieht es allerdings schnell, dass seltene Klassen die Wahrscheinlichkeit $p(c|q) = 0$ erhalten somit auch nicht mehr geschätzt werden können.
- Verwendung zukünftiger Information: Für die Klassifikation können anstatt der α -Variablen wie beim Training auch γ -Variablen oder der Viterbi-Pfad verwendet werden. In beiden Fällen geht die gesamte Sequenz in das Ergebnis ein und somit auch Zeitpunkte, die in der Zukunft des zu klassifizierenden Zeitpunktes liegen. Dies kann für Offline-Anwendungszwecke hilfreich sein.

Diese Möglichkeiten erscheinen besonders für die Erkennung von Aktivitäten von Vorteil, wobei unendlich lange Datensequenzen mit Klassenwechseln vorliegen. Durch die zusätzlichen Modellierungsoptionen kann wertvolles Wissen über den Informationsgehalt und den zeitlichen Ablauf der Daten gewonnen werden (Kapitel 8.12).

7.2.5 Verwandte Arbeiten

Klassifikation-HMMs weisen Ähnlichkeiten zu *Conditioned HMMs* auf [GSS11]: Hierbei wird die Klasse durch einen zweiten versteckten Prozess C modelliert, welcher den versteckten Zustand durch $p(Q|C)$ beeinflusst. In unserer Struktur hängt umgekehrt die Klasse vom Zustand ab durch $p(C|Q)$. Während beide Methoden ineinander überführbar sind, erlaubt das Klassifikations-HMM durch entsprechend mehr Zustände eine explizite Modellierung, was in den folgenden Untersuchungen ausgenutzt wird (Abschnitt 7.4.4). Bei *Conditioned HMMs* berichten die Autoren zudem über die Einschränkung, aus Effizienzgründen keine Klassenwechsel zuzulassen; dies ist für die Worterkennung korrekt, für den Anwendungszweck der Fahrerassistenz jedoch genau einer der treibenden Faktoren.

Eine noch universellere Möglichkeit für die Klassifikation mit HMMs bieten *Input Output HMMs* [BB96], bei denen weitere Abhängigkeiten zwischen Beobachtungen und Klasse modelliert werden. Sie wurden bisher zur Erkennung von Grammatiken verwendet und erschienen für die hochdimensionalen Fahrdaten dieses Anwendungsbereichs zu komplex.

Diskriminatives Training mit Mutual Information [Kap98] optimiert die einzelnen HMMs eines Ensembles ebenfalls in Richtung einer hohen Klassifikationsrate, führte in diesen Untersuchungen aufgrund der Komplexität der Daten jedoch nicht zu brauchbaren Ergebnissen (Kapitel 6.1.4)

7.3 Beobachtungen mit Random Forest

Die Komplexität der vorliegenden Sensordaten motiviert es, zusätzlich zur HMM-Struktur auch die Modellierung der Beobachtungen zu untersuchen; diese werden üblicherweise durch Gaussian Mixtures berechnet. Es wurde vor allem das folgende Verbesserungspotenzial aufgedeckt:

- Gaussian Mixtures können degenerieren, indem aufgrund von zuwenig Trainingspunkten die Varianz gegen 0 konvergiert. An dieser Stelle nimmt die Dichtefunktion dann unkontrollierbar hohe Werte an und verfälscht alle weiteren Berechnungen.
- Besonders diskrete oder binäre Merkmale können eine solche Degenerierung verursachen; um die Merkmale dennoch zu verwenden, muss künstliches Rauschen hinzugefügt werden oder die trainierten Mixtures nachträglich angepasst werden.
- Abhängigkeiten zwischen Merkmalen können durch die Gaussian Mixtures stets nur linear bzw. als Summe gewichteter Mixtures modelliert werden. Der Zusammenhang zwischen der Fahrzeuggeschwindigkeit und tolerierten Querschleunigung (Kapitel 5.5.2) ist damit zum Beispiel nicht korrekt abbildbar.
- Für die Bestimmung von $p(o)$ werden stets alle Merkmale verwendet; es ist nicht möglich, in bestimmten Situationen einzelne Merkmale zu ignorieren. Eine große Varianz in der entsprechenden Dimension der Gaussverteilung ist die einzige Möglichkeit, durch die jedoch die gesamte Beobachtungswahrscheinlichkeit sinkt.

Genau in diesen Situationen kann der Random Forest seine Stärke beweisen, komplexe Daten stabil verarbeiten zu können (Kapitel 6.2). Die grundsätzliche Idee ist es, die Zustandsbeobachtungen nicht mehr durch Gaussian Mixtures oder diskrete Verteilungen zu modellieren, sondern an dieser Stelle ein trainiertes Random Forest Modell zu verwenden. Im Konzept der beiden Verfahren liegt jedoch ein bedeutender Unterschied vor, was bereits formal sichtbar wird:

$$\begin{array}{ll} \text{generativ:} & p^{\text{GM}}(o|q) : \text{Zustand} \rightarrow \text{Beobachtung} & (7.22) \\ \text{diskriminativ:} & p^{\text{RF}}(c|o) : \text{Beobachtung} \rightarrow \text{Klasse} & (7.23) \end{array}$$

Gaussian Mixtures modellieren also je Zustand q die Beobachtungswahrscheinlichkeiten $p(o|q)$ (7.22), während Random Forest zu den Beobachtungen die Klassenwahrscheinlichkeiten $p^{\text{RF}}(c|o)$ berechnet (7.23). Um den HMMs die $p(o|q)$ zur Verfügung zu stellen, werden im Folgenden zwei Möglichkeiten erarbeitet.

7.3.1 Ein Random Forest Modell für alle Zustände

Mit der Bayes-Umkehrformel wird die Random Forest Klassenschätzung zu den gesuchten Beobachtungswahrscheinlichkeiten umgerechnet:

$$p^{\text{RF}}(o|q) := p^{\text{RF}}(q|o) \frac{p(o)}{p(q)} \quad (7.24)$$

Die $p(o)$ sind konstant und beeinflussen das Ergebnis des Alpha-Beta oder Viterbi Algorithmus nicht - sie können ignoriert werden. Die A-Priori-Zustandswahrscheinlichkeiten $p(q)$ werden anhand der Trainingsdaten ermittelt; alternativ können sie ebenfalls ignoriert werden, wobei die Bayes-Umkehrformel dann nicht in Erscheinung tritt. Random Forest schätzt dann den Zustand bezüglich der Beobachtungen und tendiert aufgrund seines diskriminativen Prinzips dazu, einzelne Zustände zu bevorzugen oder isolieren.

Zum Training der $p^{\text{RF}}(q|o)$ wird das Grundprinzip des Baum Welch Algorithmus aufgegriffen: Er bestimmt im *Expectation*-Schritt die Beobachtungs- und Transitionswahrscheinlichkeiten empirisch und übernimmt sie im *Maximization* Schritt als neue Parameter. Dies entspricht einem Greedy-Algorithmus mit Gradientenschritt und führt beweisbar zu einer Steigerung der A-Posteriori-Wahrscheinlichkeit der Trainingsdaten [Rab89].

Die empirischen Zustandswahrscheinlichkeiten drücken aus, mit welcher Wahrscheinlichkeit ein Zustand q bei der Beobachtung o aktiv ist und berechnen sich zu

$$p(q|o) = \frac{\sum_{t:o_t=o} \gamma_{t,q}}{|\{t : o_t = o\}|} \quad (7.25)$$

Diese Größe könnte trainiert werden, indem während des Modelltrainings die Beobachtungen o_t auf die $\gamma_{t,q}$ abgebildet werden. Der Random Forest Algorithmus benötigt jedoch diskrete Trainingsklassen c_i und kann die kontinuierlichen γ somit nicht berücksichtigen. Zunächst wird deshalb als Trainingsklasse der jeweils aktivste Zustand verwendet, und es ergibt sich die Trainingsmatrix

$$\boldsymbol{\tau} = (o_t, \operatorname{argmax}_q \gamma_{t,q})_{t=1}^T \quad (7.26)$$

Tabelle 7.1 verdeutlicht die Trainingsmatrix und die resultierende Prädiktion. Die Schätzung der Klassenverteilungen bei gleichen Features, wie bei $t = 3 \dots 6$, wird hier ausgenutzt, um weiche Schätzungen der Zustandswahrscheinlichkeiten zu erhalten.

Das Beispiel $t = 7$ stellt eine Grenze dieser Anwendung dar: Da $p^{\text{RF}}(q_2|d) = 0 \neq 0.4$, ist der Zustand q_2 im Alpha Beta Algorithmus bei der Beobachtung $o = d$ gänzlich deaktiviert. Sehr seltene Zustände können damit schnell isoliert werden; eine Gegenmaßnahme dazu wird in Kapitel 7.6 vorgestellt.

7.3.2 Ein Random Forest Modell je Zustand

Bei vielen Zuständen wird die vorherige Konstruktion schnell sehr komplex, da das Training mit zu vielen Klassen stattfindet. Alternativ wurde je Zustand q ein Random Forest

t	o	$\gamma_{t,1}$	$\gamma_{t,2}$	Ψ^{RF}	$p^{\text{RF}}(q_1 o)$	$p^{\text{RF}}(q_2 o)$
1	a	1	0	1	1	0
2	b	0	1	2	0	1
3	c	1	0	1	0.75	0.25
4	c	1	0	1	0.75	0.25
5	c	1	0	1	0.75	0.25
6	c	0	1	2	0.75	0.25
7	d	0.6	0.4	1	1	0

Tabelle 7.1: Trainingsdaten \mathcal{T} mit der zu lernenden Klasse Ψ^{RF} für Random Forest: p^{RF} ist das Resultat der Prädiktion und verdeutlicht das Erlernen von Klassenverteilungen; für $t = 7$ kann der Zustand q_2 jedoch nicht geschätzt werden.

Modell Φ_q erstellt, bei welchem konstant zwei Trainingsklassen verwendet werden. Während des Trainings werden dann nur die Zeitpunkte t verwendet, zu denen der Zustand aktiv ist, z.B. $\gamma_{t,q} > 0$. Da jeder Zustand separat trainiert wird, hängt das Resultat somit nicht mehr von der relativen Häufigkeit des Zustandes ab, und das Erlernen seltener Zustände (Abschnitt 7.3.1) wird ermöglicht.

Der Bayes-Ansatz wird umgangen, indem die Beobachtungswahrscheinlichkeiten neu interpretiert werden: Jedem Zustand werden wie zuvor zu aktiven Zeitpunkten $\gamma_{t,q} > 0$ die entsprechenden Beobachtungen zugewiesen. Die zu schätzende Größe ist aber nun eine binäre Trainingsklasse und besagt, ob der Zustand die richtige Klasse schätzt oder nicht.

Im Folgenden kennzeichnen $\alpha^{\text{GT}}, \gamma^{\text{GT}}$ und α, γ die Resultate des Alpha Beta Algorithmus unter Verwendung der korrigierten bzw. nicht korrigierten Beobachtungswahrscheinlichkeiten (Abschnitt 7.2.3). Die Grundideen zum Training sind:

- Die Aktivität $a_{t,q} = \max(\alpha_{t,q}, \gamma_{t,q}^{\text{GT}})$ ist definiert als die Wahrscheinlichkeit, dass ein Zustand während der Prädiktion oder des Trainings aktiv gesichtet wurde.
- Je Zustand werden alle Beobachtungen aktiver Zeitpunkte $t : a_{t,q} > 0$ beachtet. Das Training konzentriert sich damit jeweils auf eine Teilmenge von Beobachtungen, so dass eine Spezialisierung des Zustandes möglich wird.
- Beobachtungen o_t sollen akzeptiert werden (\oplus), wenn der Zustand die zu diesem Zeitpunkt korrekte Klasse erzeugt: $p(c_t|q) > 0$, ansonsten werden sie abgelehnt (\ominus).

Mit diesen Überlegungen wird für jeden Zustand q ein Random Forest Modell mit den folgenden Trainingsdaten \mathcal{T}_q erstellt:

$$\mathcal{T}_q = \{(o_t, \oplus) \mid a_{t,q} > 0 \wedge c(q) = c_t\} \quad (7.27)$$

$$\cup \{(o_t, \ominus) \mid a_{t,q} > 0 \wedge c(q) \neq c_t\} \quad (7.28)$$

Dabei kann es geschehen, dass die Trainingsdaten für einen Zustand trivial oder leer werden; beim Training hat der Zustand also entweder stets korrekt oder stets falsch klassifiziert. Der erste Fall stellt kein Problem dar - alle Beobachtungen können dann künftig

in diesem Zustand entstehen, der letztlich eine Gleichverteilung darstellt. Der zweite Fall würde bedeuten, dass der Zustand niemals aktiv war, wenn seine Klasse in einer Trainingssequenz vorkam. Mit einfachen strukturellen Gegenmaßnahmen (Abschnitt 7.4.1) kann gewährleistet werden, dass der Zustand bei der nächsten Iteration wieder aktiv wird und weiter verwendet werden kann.

t	Ψ	$a_{t,1}$	$a_{t,2}$	Ψ_1	Ψ_2
1	a	1	0	\oplus	-
2	a	0.1	0.9	\oplus	\ominus
3	a	0.7	0.3	\oplus	\ominus
4	a	0	1	-	\ominus
5	b	1	0	\ominus	-
6	b	0.4	0.6	\ominus	\oplus
7	b	0.6	0.4	\ominus	\oplus
8	b	0	1	-	\oplus

Tabelle 7.2: Wahl der Trainingsdaten τ_q zum Trainieren eines Random Forest Modells je Zustand q . Die Zustände repräsentieren die Klassen $c(q_1) = a, c(q_2) = b$; damit werden passend zur Ground Truth Klasse Ψ die Trainingssignale $\Psi_i \in \{\oplus, \ominus\}$ definiert, insofern der Zustand aktiv ist.

Tabelle 7.2 verdeutlicht die Wahl der Trainingsbeispiele und -klasse. Der Zustand q_1 ist für $t = 1...3, 5...7$ aktiv. Da er die Klasse $c(q_1) = a$ modelliert, wird für die Zeitpunkte $t = 1...3$ die Akzeptanz \oplus gelernt, für $t = 5...7$ die Zurückweisung \ominus . Der genaue Wert der Aktivität geht also nicht in das Training mit ein, wodurch sich für geringe Zustandsaktivitäten ein zu starkes Training ergeben kann. Wie zuvor kann Random Forest auch hierfür mit gewichteten Trainingsdaten angewandt werden (Kapitel 7.6).

7.4 Struktur des HMMs

Die Klassifikations-HMMs und das neue Training der Random Forest Beobachtungswahrscheinlichkeiten erfordern weitere Anpassungen, damit vor allem das Modelltraining mit dem Baum Welch Algorithmus zum gewünschten Erfolg führt. Dabei stehen besonders die Transitionen und die Modellstruktur im Vordergrund, damit alle Zustände aktiv werden können.

7.4.1 Vermeidung isolierter Zustände

Die Transitionswahrscheinlichkeiten wirken sich stark auf die resultierende Klassifikation aus und können leicht zu Problemen führen. Abb. 7.5 zeigt dies exemplarisch: In der Szene liegt die Klasse $\Psi = 4$ vor. Der Zustand q_4 entspricht der korrekten Klasse $c(q_4) = 4$, Er berechnet jedoch für $t = 5...10$ die Beobachtungen fälschlicherweise mit $p(o|q_4) = 0$, erzielt aber ab $t \geq 13$ korrekterweise den höchsten Wert aller Zustände.

Da alle Zustände isoliert sind, muss das HMM in einem einmal aktiven Zustand verbleiben; q_4 kann für $t \geq 5$ also in den α -Variablen nie wieder aktiv werden. Im Viterbipfad taucht er überhaupt nicht auf, weil dieser die gesamte Sequenz berücksichtigt, und kein durchgängiger Pfad mit q_4 möglich ist. Die korrekte Schätzung ab $t \geq 13$ kann somit nach der vorherigen Fehlschätzung nicht mehr berücksichtigt werden.

Diese Umstände können in zwei Ursachen zusammengefasst werden:

- Die Transitionen $p(q \rightarrow r)$ lassen nicht genügend Zustandswechsel zu. Die Ursache liegt in den Trainingssequenzen, die verhältnismäßig wenige Klassenwechsel aufweisen; entsprechend adaptiert der Baum Welch Algorithmus auch die Transitionen.
- Die Beobachtungswahrscheinlichkeiten $p(o|q) = 0$ isolieren einen Zustand q . Dies kann bei Random Forest Beobachtungen vorkommen, mit Gaussian Mixtures ist dieser Fall unmöglich, weil der Beobachtungsraum sich unendlich erstreckt.

Als Gegenmaßnahme werden die Wahrscheinlichkeiten jeweils auf ein Minimum c_* erhöht und anschließend die stochastische Korrektheit durch Normierung wiederhergestellt:

$$p'(q \rightarrow r) := \max(p(q \rightarrow r), c_a) \quad (7.29)$$

$$p'(o|q) := \max(p(o|q), c_o) \quad (7.30)$$

Abb. 7.6 zeigt das Ergebnis der Anpassung: Der Viterbi-Algorithmus wählt nun ab $t \geq 15$ den korrekten Zustand q_4 mit dem Umweg über den Neutralzustand q_1 . Die α -Variablen reagieren mit geringer Verzögerung.

7.4.2 Variation der Zustandsverweildauern

Um die Zustände des HMMs bestimmten Zeitpunkten in den Daten zuordnen zu können, wurden die Verweildauern angepasst (Kapitel 6.1.2). Unter Vorgabe der erwarteten Verweildauer D wurden die Selbsttransitionen auf den berechneten Wert gesetzt und die übrigen Transitionen entsprechend skaliert:

$$p^D(q \rightarrow q) := 1 - D^{-1} \quad \forall q \quad (7.31)$$

$$p^D(q \rightarrow r) := p(q \rightarrow r) \frac{D^{-1}}{\sum_{s \neq q} p(q \rightarrow s)} \quad \text{für } q \neq r \quad (7.32)$$

Mit dieser Initialisierung wird im Mittel die Zustandsverweildauer D erreicht; danach erfolgt ein Zustandswechsel $q \rightarrow r$. Die Beobachtungen für q, r unterscheiden sich somit entsprechend der Zeitspanne D ; mit wachsendem D ist eine Steigerung der Diversität zu erwarten, welche sich wiederum auf die Anpassungsfähigkeit an die Daten und die Klassifikationsrate auswirkt.

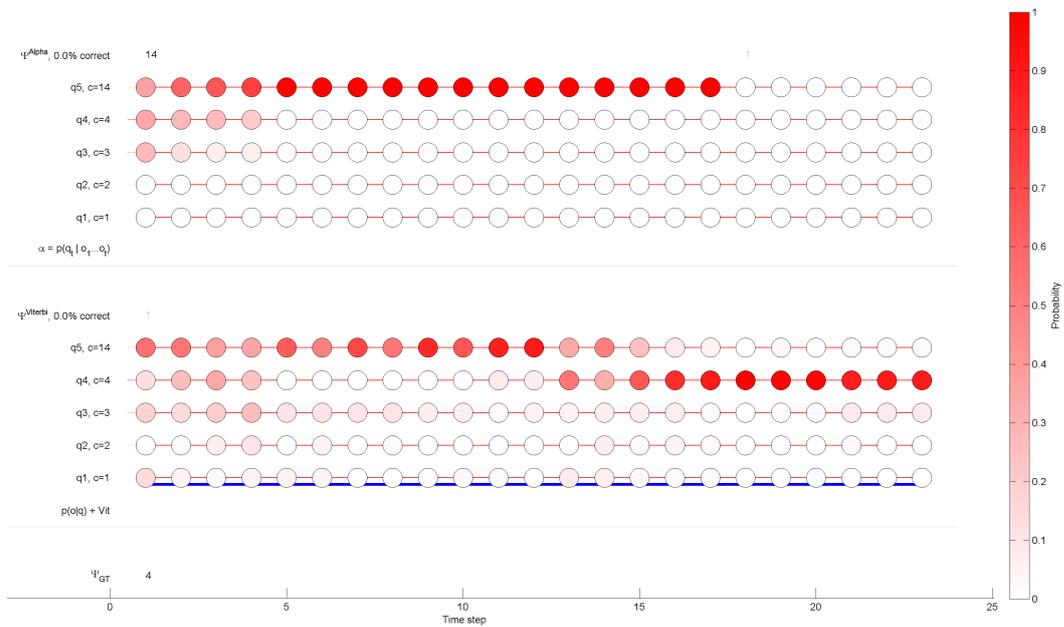


Abbildung 7.5: Trellis-Diagramm eines HMMs mit isolierten Zustände. Die Ground Truth Klasse dieser Sequenz ist durchgängig $\Psi^{GT} = 4$. Unten: Trellis-Diagramm mit Beobachtungswahrscheinlichkeiten $p(o|q)$ (Farbintensität) und Viterbipfad (blau). Linien zwischen den Zuständen visualisieren die Übergangswahrscheinlichkeiten, in diesem Beispiel ist also kein Zustandswechsel möglich. Oben: α -Variablen.

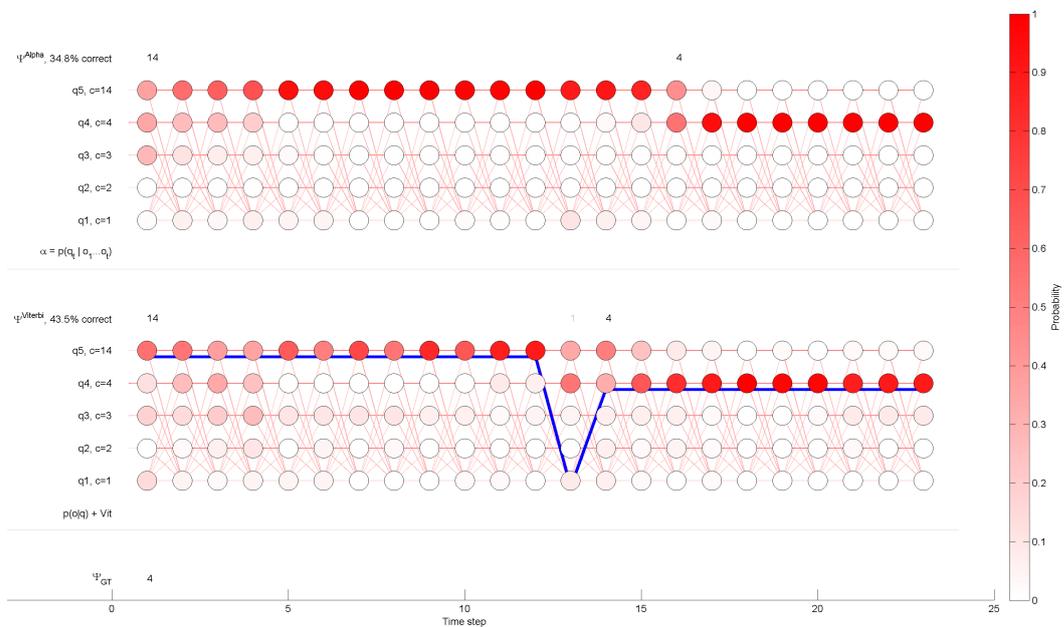


Abbildung 7.6: Trellis-Diagramm desselben HMMs, wobei Transitionen von und zu dem Neutralzustand q_1 nachträglich eingefügt wurden. Durch die erhöhte Konnektivität wird nun ab $t \geq 14$ bzw. 16 der korrekte Zustand q_4 erreicht.

7.4.3 Modellierung expliziter Ablaufphasen

Als Alternative zur Anpassung der Zustandswahrscheinlichkeiten wurden absolute Zeitpunkte untersucht. Die Idee ist es, Indikatoren für den zeitlichen Fortschritt in den Daten selbst zu finden: Zum Beispiel kann anhand der Lautstärke oder Stimmlage des Sprechers darauf geschlossen werden, an welcher Stelle im Satz dieser sich gerade befindet [PK08].

Zur Ermittlung eines geeigneten Indikatorsignals f_{Time} dienen folgende Überlegungen: Viele Sequenzen können in Ablaufphasen unterteilt werden - Gesten, Wörter oder Fahrmanöver z.B. in die Phasen Vorbereitung, Durchführung und Abschluss [PL99, BED08]. Innerhalb dieser Phasen ähneln sich Sequenzen gleicher Klasse besonders, so dass die Kenntnis über die aktuelle Phase die Klassifikation verbessern kann. Zwar ist es ohnehin die implizite Aufgabe eines HMMs, die Phasen zu finden und auf spezielle Zustände abzubilden - ein explizit bekanntes Indikatorsignal f_{Time} kann aber zusätzlich wertvolle Beiträge leisten.

Im Anwendungsfall kommen u.A. die geschätzte Kollisionszeit f_{TTC} oder die verbleibende Reaktionszeit f_{TBRake} in Frage (Kapitel 5.6). Hier fiel die Wahl auf f_{TTC} , was ein starker Indikator für die subjektive Kritikalität ist; es lässt damit Rückschlüsse auf den Entscheidungszeitpunkt und Manöverbeginn zu (Kapitel 2.9 und 4.3).

Die Verknüpfung des Indikatorsignals mit den Zuständen geschieht in Form zusätzlicher Beobachtungswahrscheinlichkeiten (Abb. 7.7): Zum Beispiel wird der Zustand q mit der folgenden Rechteckverteilung an die kritische Phase einer Annäherung gebunden:

$$p(f_{\text{TTC}}|q) = \text{uniform}(2s, 3s) \quad (7.33)$$

Wird f_{TTC} ohnehin auch als Beobachtung verwendet, so passt sich die entsprechende Dimension der $p(o|q)$ im Baum Welch Algorithmus der expliziten Initialisierung an. Die trainierten $p(o|q)$ erlauben dann wiederum Rückschlüsse auf die zeitliche Struktur der Daten und die Verweildauer in einzelnen Phasen.

7.4.4 Explizite Pfade im HMM je Klasse

Bei der vorherigen Modellierung von Ablaufphasen entsteht eine sequentielle Verbindung der Zustände gemäß der Phasenübergänge. Ein solches Modell lässt sich aber auch Top-Down herleiten: Der Zustand soll die Fahrerintention sowie den Fortschritt des Manövers widerspiegeln, wozu die Transitionen den Ablauf des Manövers und eventuelle Wechsel der Intention modellieren. Die Struktur ähnelt den Left-to-right-Models, die auch in verwandten Arbeiten zur Fahrmanövererkennung verwendet werden [Mit05, BED08], jedoch erhalten die Zustände und Transitionen hier eine explizite Bedeutung (Tabelle 7.3).

Mit dieser Modellierung können insbesondere mehrere Pfade je Klasse verwendet werden: Ein Zustand erhält dann als zusätzliche Bedeutung neben der Klasse und Phase auch

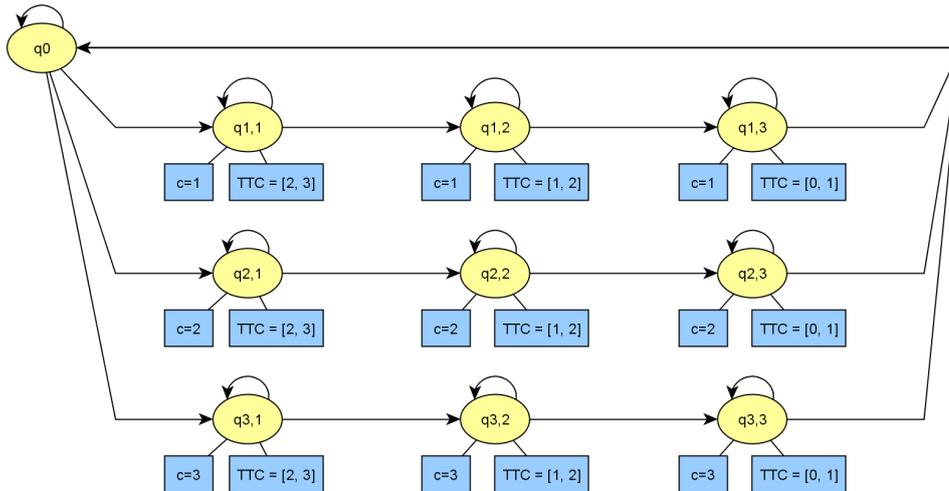


Abbildung 7.7: Top-Down-Erstellung eines HMMs für sequentielle Vorgänge verschiedener Klassen mit Unterteilung in mehrere Ablaufphasen.

die Ausprägung, was es erlaubt, ein HMM aus prototypischen Sequenzen zusammenzusetzen.

$q_{c,i}$	Zustände für Klasse c und Ablaufphase i
q_{Neutral}	Neutralzustand
$\pi(q_{c,i})$	Startwahrscheinlichkeit einer Klasse; meistens auf die Neutralklasse konzentriert
$q_{\text{Neutral}} \rightarrow q_{c,i}$	Sobald die Klasse c beginnt, soll das HMM in einen entsprechenden Zustand und die Initialphase $i = 1$ wechseln. Mit $i > 1$ können Phasen übersprungen werden.
$q_{c,i} \rightarrow q_{\text{Neutral}}$	Ein Wechsel zurück in den Neutralzustand sollte stets möglich sein, wenn Klassen abrupt enden, z.B. wenn ein Objekt im Sensor verloren geht.
$q_{c,i} \rightarrow q_{c,j}$	Wahrscheinlichkeit eines Wechsel in die nächste Ablaufphase j . Mit expliziter Initialisierung (Abschnitt 7.4.3) ergibt sich dieser Parameter während des Trainings automatisch

Tabelle 7.3: Komponenten und resultierende Variationspunkte eines temporalen HMMs. Mit $q \rightarrow q'$ wird hier die Transition und deren Bedeutung zwischen zwei Zuständen bezeichnet.

7.5 Adaptive Abtastpunkte für Moving Window

Für die *Moving Window* Technik (Kapitel 6.3) werden die Abtastpunkte h so gewählt, dass die benötigte Information im Fenster enthalten ist. Gleichzeitig soll es möglichst klein sein, um keine Redundanzen zu enthalten. Ziel ist es, das relevante Muster des untersuchten Signals zu erfassen. Je nach Art der Daten ist ein Fenster konstanter Länge

dafür nicht immer optimal, denn bei unterschiedlichen Geschwindigkeiten der Ausführung dauert das Muster mehr oder weniger lang.

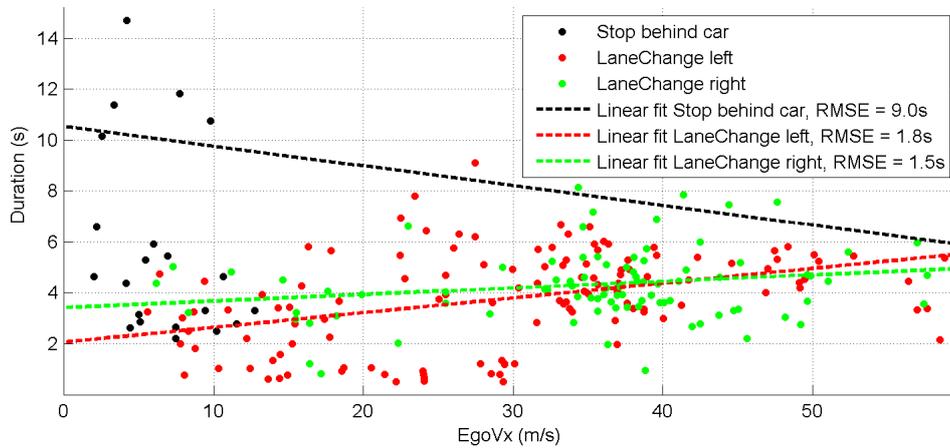


Abbildung 7.8: Dauer von gelabelten Highlevel-Manövern gegenüber der Geschwindigkeit. Trotz hoher Streuung, die zum Teil auch durch das manuelle Labelling bedingt ist, ist für Spurwechsellvorgänge eine klare Tendenz erkennbar, was die Regressionsgerade verdeutlicht (rot und grün).

Die Dauer von Fahrmanövern variiert zwar stark, was auch am manuellen Labelling liegen kann, trotzdem kann ein deutlicher Zusammenhang zur Geschwindigkeit gefunden werden (Abb. 7.8). Er ist unter anderem dadurch erklärbar, dass mit steigender Geschwindigkeit geringere Querbewegungen akzeptiert werden [SH82]. Ein lineares Regressionsmodell kann die mittlere Manöverdauer $d(v)$ abhängig von der Geschwindigkeit v schätzen (Tabelle 7.4):

Manöver	Mittl. Dauer	Korrelation	Regressionsgerade $d(v)$
Anhaltevorgang	10.02s	-0.03	$10.54s - 0.08s \cdot v / (m/s)$
Spurwechsel links	3.80s	0.39	$2.06s - 0.06s \cdot v / (m/s)$
Spurwechsel rechts	4.33s	0.18	$3.42s - 0.03s \cdot v / (m/s)$
Spurwechsel li/re	4.02s	0.34	$2.44s - 0.05s \cdot v / (m/s)$

Tabelle 7.4: Korrelation und lineare Regression zur Modellierung der Dauer verschiedener Fahrmanöver in Abhängigkeit von der Geschwindigkeit v zu Manöverbeginn.

Es erschien sinnvoll, zwecks einer größeren Datenmenge Spurwechsel als symmetrisch anzunehmen und links / rechts miteinander zu kombinieren. Es ergibt sich eine schwache Korrelation von 0.34. Bei Anhaltevorgängen ist der Zusammenhang nicht feststellbar, was dadurch zu erklären ist, dass sich die Geschwindigkeit während eines solchen Vorgangs verändert und die Vergleichbarkeit der Szenen somit schwierig wird.

Für adaptive Abtastpunkte wird zunächst eine Basis h_0 bezüglich einer Referenzgeschwindigkeit v_0 definiert (Kapitel 8.4). Während der Prädiktion wird anhand der gemessenen Geschwindigkeit v der Wert der Regressionsgerade $d(v)$ berechnet und damit die h_0 linear gestreckt:

$$h(v) = h_0 \frac{d(v)}{d(v_0)} \quad (7.34)$$

Durch die Modifikation der Abtastzeitpunkte wurde nur eine minimale Verbesserung der Klassifikation auf Deeskalationsmanövern erreicht; es wurde geschlussfolgert, dass die Variation der Manöverdauer doch stark von komplexeren Faktoren als nur der Egogeschwindigkeit abhängt. Sporrer et al. [SPBS98] kamen in einer detaillierten Betrachtung der Beschleunigungsverläufe zu ähnlichen Ergebnissen und nennen als weitere Einflussgröße z.B. die Verkehrsdichte, den Zielfahstreifen und die Strecke des Spurwechselvorgange.

7.6 Random Forest und Weighted Classes

Bei der Anwendung diskriminativer Klassifikatoren kann es in bestimmten Fällen sinnvoll sein, gewichtete Trainingsdaten oder gewichtete Klassen zu betrachten. Für die Modellierung von Beobachtungswahrscheinlichkeiten in einem HMM (Kapitel 7.3) soll z.B. eine Abbildung der Beobachtungen auf die α -Variablen geschehen. In diesem Fall liegt also zu jedem Trainingsdatum i keine deterministische Trainingsklasse c_i vor, sondern eine Klassenverteilung $c_{i,1} \dots c_{i,m}$ bei m Klassen. Ebenso können die Trainingsdaten selbst gewichtet sein mit dem Koeffizienten w_i (Tabelle 7.5), was ineinander überführt werden kann.

i	1	2	3	4	5
Beobachtung o_i	a	b	c	c	e
Klasse c_i	1	2	1	2	1
Gewicht w_i	1	1	0.2	0.8	1

Tabelle 7.5: Gewichtete Trainingsbeispiele: Zu jeder Beobachtung i gibt es ein Gewicht w_i , welches zum Beispiel die Aktivität eines HMM-Zustandes sein kann (Kapitel 7.3). Die Beobachtung $o = c$ gehört also mit 20% Wahrscheinlichkeit zu Klasse 1 und 80% zu Klasse 2.

Prinzipiell könnte hierfür die Regressions-Erweiterung für Random Forest eingesetzt werden, um direkt auf kontinuierlichen Größen trainiert zu werden [Bre01]; das Schätzergebnis modelliert dann jedoch keine Unsicherheiten, welche besonders im Zusammenhang mit der Bayes-Formel zum Einsatz kommen. Zudem tauchten einige technische Schwierigkeiten auf, weil die Erstellung eines Regression Forest aufwändiger und schwieriger nachvollziehbar ist als bei einem Klassifikator auf diskreten Klassen.

$$\overline{\mathcal{T}}^{\text{Weighted}} = \left\{ \begin{array}{ccccc} (a, 1) & (b, 1) & (c, 1) & (c, 2) & (e, 1) \\ (a, 1) & (b, 1) & & (c, 2) & (e, 1) \\ (a, 1) & (b, 1) & & (c, 2) & (e, 1) \\ (a, 1) & (b, 1) & & (c, 2) & (e, 1) \\ (a, 1) & (b, 1) & & & (e, 1) \end{array} \right\}$$

Tabelle 7.6: Äquivalente Reproduktion gewichteter Trainingsdaten zu ungewichteten Daten (o, c) , bestehend aus Beobachtung und Trainingsklasse. Die Auftretenshäufigkeiten der Trainingstupel entsprechen der zuvor gezeigten Gewichtung w_i .

Hier wurde eine einfache Methode gewählt, um gewichtete Trainingsdaten zu berücksichtigen: Die Beobachtungen werden entsprechend der Koeffizienten $c_{i,j}$ bzw. w_i reproduziert und mit jeweils der Klasse j trainiert (Tabelle 7.6). Somit wird die Eigenschaft

von Random Forest ausgenutzt, im Falle genau gleicher Trainingsdaten deren empirische Klassenverteilung wiederzugeben (Kapitel 6.2.2, Rekursionsabbruch).

Die Problemstellung ähnelt der Situation *Imbalanced data*, in welcher die Klassen zu stark verschiedenen Anteilen repräsentiert sind (z.B. Erkennung seltener Krankheiten). Hierfür ermöglicht Random Forest eine Lösung durch Anpassung der globalen Klassengewichte [CLB04]. Im vorliegenden Fall liegt aber keine globale Häufigkeitsungleichheit vor, sondern jedes Trainingsdatum besitzt eine eigene Gewichtung. Mit der gewählten Methode der Reproduktion entstehen viele redundante Trainingsdaten, in Untersuchungen hat sie aber teilweise zu verbesserten Resultaten beigetragen (Kapitel 8.8).

7.7 Kursprädiktion durch Random Forest

Bisher wurden methodische Erweiterungen vorgestellt für die Erkennung der diskreten Fahrerintention, modelliert durch Highlevel- oder Deeskalationsmanöver. Die als generische Alternative motivierte Kursprädiktion (Kapitel 4) arbeitet im Gegensatz dazu auf kontinuierlichen Größen. Dadurch ändert sich die Anwendung Maschinellen Lernens grundsätzlich: Die Kursprädiktion ist nicht an konkrete Manöver gekoppelt, wodurch auch die Auswertung sowie Optimierungskriterien des Modelltrainings generischer werden.

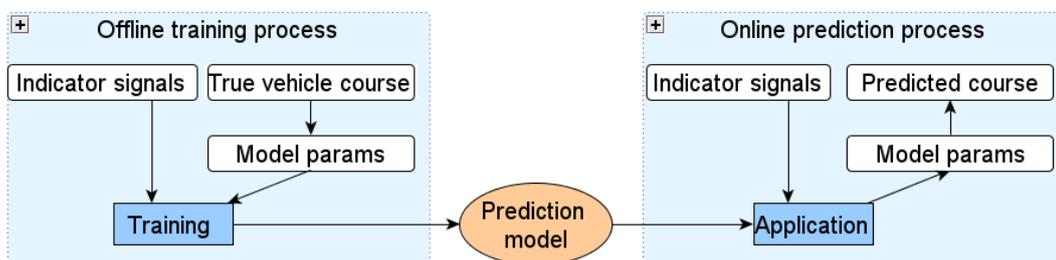


Abbildung 7.9: Framework der Kursprädiktion mit Maschinellem Lernen. Der zukünftige Fahrzeugkurs wird durch ein Fahrmodell approximiert und dessen Parameter transformiert, woraus anschließend unter Berücksichtigung von Indikatorsignalen ein Prädiktionsmodell trainiert wird. Zur Online-Prädiktion wird die Kette rückwärts angewandt.

7.7.1 Diskretisierung

Weil der Random Forest Algorithmus primär diskrete Größen schätzt, müssen die kontinuierlichen Kursparameter zunächst diskretisiert werden. Prinzipiell kann der Algorithmus zwar durch lineare Regressionsmodelle in den Blättern auch kontinuierliche Größen lernen; mit den gängigen Verfahren resultiert hieraus jedoch eine deterministische statt probabilistische Schätzung. Die zugleich höhere Komplexität des Verfahrens machte eine Verringerung der Trainingsdaten notwendig, was durch eine Halbierung der Abtastfrequenz durchgeführt wurde. In Summe ergaben sich schlechte Resultate, die zugleich schwerer verständlich und analysierbar waren.

Die benötigten diskreten Schätzgrößen werden also ermittelt, indem die Parameter des Kursmodells transformiert (Kapitel 5.5) und anhand eines logarithmischen Rasters diskretisiert werden. Mit diesem Verfahren wird die Auflösung um den Nullpunkt vergrößert und nimmt mit steigenden Werten logarithmisch ab [PK08]. Für den Anwendungsbereich der Kollisionsdetektion scheint dies sinnvoll, da schon leichte Lenkbewegungen den entscheidenden Unterschied zwischen einer Kollision und einem Überholvorgang ausmachen können. Bei starken Bewegungen genügt meist schon die Information, dass eine intensive Aktivität vorliegen wird, so dass hier ein höherer Diskretisierungsfehler akzeptiert wird.

7.7.2 Prädiktion der Kursparameter

Da durch den Random Forest Algorithmus jeweils nur eine einzelne Größe prädiziert werden kann, werden die Kursparameter a, b unter der Annahme stochastischer Unabhängigkeit durch zwei unabhängige Modelle berechnet. Mit den Ego- und Objektdaten S_E, S_O als Indikatoren ergibt sich:

$$p^{\text{RF}}(a, b | S_E, S_O) = p^{\text{RF}}(a | S_E, S_O) p^{\text{RF}}(b | S_E, S_O) \quad (7.35)$$

Die Parameter werden also unabhängig voneinander prädiziert und miteinander kombiniert. Grundsätzlich könnten hierdurch Einschränkungen entstehen: So kann z.B. nicht geschätzt werden, dass der Fahrer ausweicht oder bremst, nicht aber beides gleichzeitig tut. *Structured output learning* ist eine erweiterte Form des Maschinellen Lernens, die genau diese Möglichkeiten bietet [BL08, BS06]. Da jedoch weitestgehend davon ausgegangen wird, dass die meisten Fahrer vor allem in eskalierenden Situationen nur eine Handlung ausführen (Abb. 4.3), kann die beschriebene Vereinfachung akzeptiert werden.

Für das probabilistische Framework (Kapitel 4.1) wird die Wahrscheinlichkeit für eine Position x zum Zeitpunkt t gesucht. Sie entsteht durch Marginalisierung aller Parameter und jeweiliger Anwendung dieser Parameter mit dem Kursmodell:

$$p^{\text{RF}}(x_t | S_E, S_O) = \int_a \int_b p(x_t | a, b) p^{\text{RF}}(a | S_E, S_O) p^{\text{RF}}(b | S_E, S_O) \quad (7.36)$$

$$= \int_a \int_b g^{\text{Circ}}(a, v, b, t) p^{\text{RF}}(a | S_E, S_O) p^{\text{RF}}(b | S_E, S_O) \quad (7.37)$$

Hierbei ist v die Fahrzeuggeschwindigkeit, g^{Circ} das zirkuläre Kursmodell, und die zu schätzenden Parameter a, b sind Kurskrümmung und Beschleunigung des Kursmodells.

7.8 Zusammenfassung

Klassifikations-HMMs bieten flexiblere Strukturierungsmöglichkeiten, sind einfacher handhabbar als Ensemble-HMMs und dabei beweisbar mindestens gleichmächtig. In Kombination mit *Growing Window* Sequenzen wurde damit besonders für unendliche zeitliche

Daten eine nützliche Methode geschaffen, welche die effiziente Analyse ermöglicht und überlappende, redundante Sequenzen vermeidet.

Der Random Forest Algorithmus wurde zur Modellierung der Beobachtungswahrscheinlichkeiten bei HMMs eingesetzt, da er eine hohe Stabilität bei komplexen Beobachtungen aufweist. Hierfür wurde ein generativer und ein diskriminativer Ansatz vorgestellt und das Training auf gewichtete Klassen erweitert.

Die Fahrerintention, die im praktischen Teil dieser Arbeit untersucht wurde, wird zusammenfassend auf drei Arten geschätzt: Direkte Schätzung der diskreten Klasse mit Random Forest oder Hidden Markov Modellen sowie indirekt über die kontinuierliche Kursprädiktion. Die vorgestellten algorithmischen Erweiterungen können auch in anderen Bereichen der Verhaltens- und Mustererkennung eingesetzt werden.

8 Resultate

Im experimentellen Teil dieser Arbeit wird zunächst die Voruntersuchung des Bayes-Fehlers beschrieben und dadurch die Wichtigkeit historischer Daten begründet. Diese wurden mit der *Moving Window* Technik verwendet, um mit dem Random Forest Algorithmus Highlevel- bzw. Deeskalationsmanöver zu klassifizieren.

Anschließend wird als Schwerpunkt dieser Arbeit die Schätzung der diskreten Fahrerintention mit Hidden Markov Modellen ausgewertet, zunächst mit Ensemble-HMMs und Gaussian Mixture Beobachtungen. Darauf basierend wird zu der Struktur des vorgestellten Klassifikations-HMMs und dessen Kombination mit Random Forest übergeleitet, um die Vorteile dieser Erweiterungen auch in der Praxis zu belegen.

Die Kursprädiktion schätzt die zukünftige Trajektorie des Egofahrzeugs über kontinuierliche Modellparameter; sie wird deshalb zunächst generisch ausgewertet. Für den Gesamtvergleich der Methoden werden daraus gemäß dem probabilistischen Framework Deeskalationsmanöver abgeleitet. Eine anwendungsnahe Untersuchung vergleicht abschließend für die verschiedenen Schätzer, welcher Nutzen in einem darauf basierendes Warnsystem entsteht.

Die Verwendung verschiedener Modellierungen und Prädiktoren für die Fahrerintention, kombiniert mit generischen als auch anwendungsorientierten Auswertungen, erlaubt eine zusammenfassende Interpretation, was über die Methoden allgemein als auch im Hinblick auf Fahrerassistenz gelernt wurde. Daraus werden schließlich Erkenntnisse über die zeitliche Beschaffenheit und Indikatoren der Daten abgeleitet, die auch auf weitere Gebiete der Verhaltenserkennung übertragen werden können.

8.1 Datenbeschreibung

Die verwendeten Daten entstammen einem Versuchsträger vom Typ Audi A8 mit Automatikgetriebe. Insgesamt wurden 10.9h aufgezeichnete Daten mehrerer Fahrer verwendet, die zurückgelegte Strecke betrug 800.3km. 1.2h der Daten wurden auf einer Teststrecke aufgezeichnet, um auch kritische Szenen zu berücksichtigen: Bei Geschwindigkeiten zwischen 5 und 35m/s wurden von geschulten Fahrern 156 Szenen eingefahren, die knappes Ausweichen vor bewegten und stehenden Objekten, spätes Bremsen und Kollisionen mit stehenden Objekten beinhalten.

Die restlichen Daten wurden zu gleichen Anteilen im Stadtverkehr, auf Autobahnen und bei Überlandfahrt aufgezeichnet, was dem für Absicherungskonzepte üblichen 1/3-Mix entspricht. Durch diese explizite Auswahl konnte sichergestellt werden, dass die Datenmenge repräsentativ ist; gleichzeitig war sie bewusst klein gehalten, um effizientes Arbeiten zu ermöglichen.

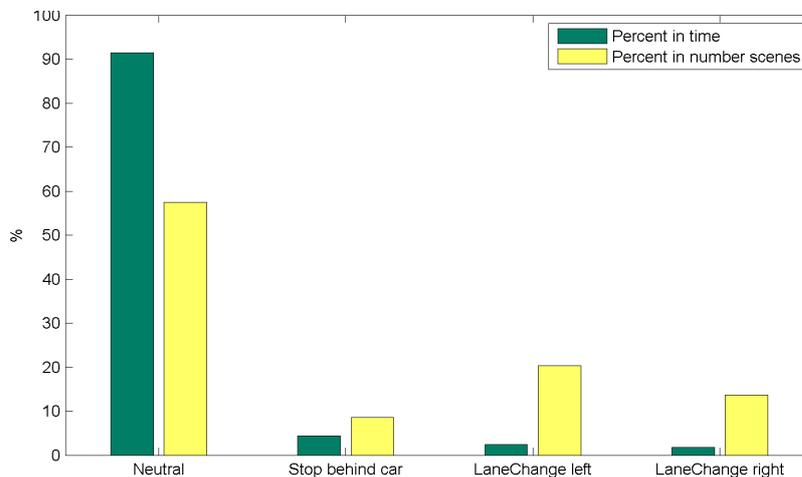


Abbildung 8.1: Verteilung der gelabelten Highlevel-Manöver in den untersuchten Daten (insg. 7.8h). Beim Labeln entsteht nach jeder Szene meistens eine Neutral-Szene, so dass diese Klasse etwa die Hälfte aller Szenen ausmacht. Weil die meisten Manöver sehr kurz sind, fällt deren Anzahl deutlich mehr ins Gewicht als die Gesamtdauer.

Für die Untersuchung der Highlevel-Klassen wurden aufgrund der Ähnlichkeit der Szenen weniger Autobahndaten verwendet, woraus sich eine Teilmenge von 6.9h Daten ergab. Die manuell gelabelten Manöver sind nicht gleich verteilt: Meistens liegt die Klasse Neutral vor, die Spurwechsel- und Anhaltvorgänge machen in der Summe nur 8.5% der Zeit aus. Aufgrund der meistens nach links gefahrenen Ausweichszenen liegen mehr solche Szenen vor, die als Spurwechsel links gelabelt sind als nach rechts. Anhaltvorgänge sind am seltensten, belegen aber verhältnismäßig viel Zeit, was durch viele Szenen an roten Ampeln erklärt wird (Abb. 8.1).

Die Deeskalationsklassen sind ähnlich ungleichmäßig verteilt (Abb. 8.2): Hier liegt zu ca. 96% der Zeit die Klasse Neutral vor, da die Deeskalationsmanöver nur in kritischen Situationen mit $TTC < 3s$ definiert sind (Kapitel 5.3).

8.2 Verfahren und Parametrierungen

Sofern nicht anders beschrieben, wurden in den Untersuchungen die folgenden Standardparametrierungen verwendet:

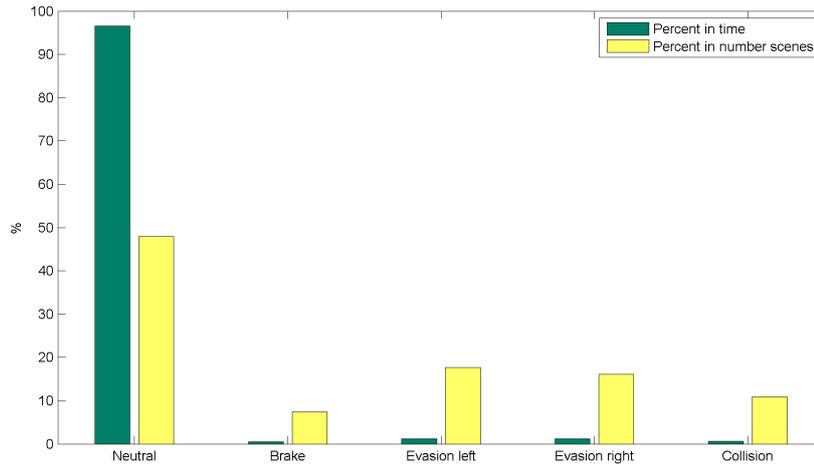


Abbildung 8.2: A-Priori-Wahrscheinlichkeiten der Deeskalations-Manöver in den untersuchten Daten (insg. 10.9h). Die Klasse ist nur definiert, wenn die Szene kritisch ist, weshalb bezüglich der Anzahl klassifizierter Zeitpunkte die Neutral-Klasse so stark überwiegt.

Random Forest

Die Anzahl der Bäume N_{Tree} wird meistens so groß wie möglich gewählt, weil die Entscheidungsgrenzen von Random Forest mit wachsendem N_{Tree} konvergiert [Bre01]. Eine obere Grenze entsteht allein praktisch durch den Speicherbedarf und die benötigte Rechenzeit. In unseren Untersuchungen wurden $N_{\text{Tree}} = 50 \dots 100$ verwendet, wobei ab $N_{\text{Tree}} \geq 30$ keine signifikanten Verbesserungen im Ergebnis mehr festgestellt werden konnten.

Als Indikatoren signale wurde meist eine Teilmenge von $m = 14$ Merkmalen verwendet, die in einer parallelen Untersuchung mit dem *Maximum Relevance Ranking* ermittelt wurden [HBZ10].

HMMs

Um zeitdiskrete Daten zu erhalten, wurden die Daten entsprechend der Taktrate des Radarsteuergerätes verwendet; diese betrug ca. $\omega = 10\text{Hz}$ mit geringen Schwankungen. Da HMMs durch variable Verweildauern genau solche Schwankungen kompensieren können, wurde bewusst auf eine Nachverarbeitung verzichtet.

Die Modelle wurden mit randomisierten Transitionen und Startwahrscheinlichkeiten initialisiert; bei Gaussian Mixture Beobachtungen wurde die Initialisierung mittels *K-Means* durchgeführt. Die Kovarianzmatrizen Σ wurden auf eine Diagonalstruktur reduziert, so dass direkte Abhängigkeiten zwischen Merkmalen entfallen und deutlich weniger freie Parameter trainiert werden müssen - in Voruntersuchungen konnte kein bedeutender Vorteil der ansonsten vollbesetzten Σ gefunden werden.

Validierung

Zur Validierung der Maschinellen Lernverfahren wurden die gemessenen Daten zu gleichen Teilen in disjunkte Trainings- und Testdaten unterteilt. In einer Voruntersuchung konnte bestätigt werden, dass diese Aufteilung dank hinreichend großer Datenmenge repräsentativ war, da Kreuzvalidierung mit verschiedenen anderen Partitionen stets sehr ähnliche Ergebnisse liefert.

8.3 Bayes-Fehler

Mit dem Bayes-Fehler (Kapitel 6.4) wurde eine Abschätzung über die mögliche Klassifikationsgüte und den Informationsgehalt historischer Signalwerte durchgeführt. Es wurde ermittelt, ob und bis zu welchem Zeitpunkt historische Werte Information für die aktuelle Klasse liefern und damit zur Klassifikation beitragen können.

Bei den folgenden Auswertungen ist zu beachten, dass nur drei Merkmale mitsamt Historie verwendet wurden, so dass die resultierenden Fehlerraten in jedem Fall höher sein müssen als unter Verwendung aller Merkmale. Diese Einschränkung war nötig, weil die Kerndichteschätzung in der Praxis auf hochdimensionalen Daten schnell in Schwierigkeiten einer zu geringen Datenmenge lief.

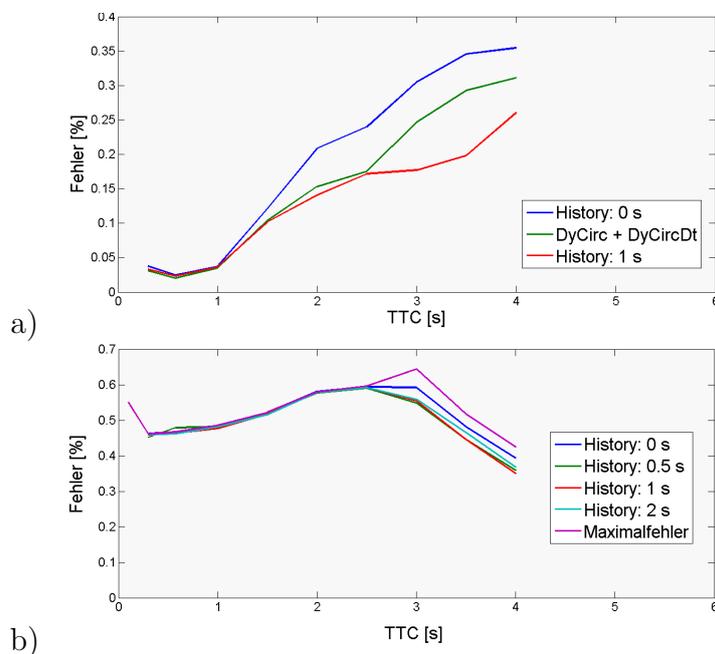


Abbildung 8.3: Bayes-Fehler bezüglich der geschätzten Kollisionszeit TTC für die Klassifikation der Deeskalations-Manöver mit den Merkmalen a) DyCirc = Kursversatz und b) dem Gradienten der Gaspedalstellung. Mit sinkender TTC, also Annäherung an das andere Objekt, wird die Schätzung der Fahrerabsicht zunehmend besser, was der Erwartung entspricht.

Resultate

Abb. 8.3 zeigt den Bayes-Fehler exemplarisch für zwei Merkmale, ausgewertet in Abhängigkeit von der geschätzten Kollisionszeit TTC (Kapitel 4.5.3). In a) wurde der Kursabstand f_{DyCirc} verwendet. Es wird deutlich, dass mit sinkender TTC bessere Klassifikationen möglich sind. In Kombination mit dem Gradienten $f_{DyCircDt}$ ergibt sich besonders für höhere TTCs eine weitere Verbesserung, die mit zusätzlich 1s Historie noch ausgeprägter wird.

In b) wurde das Merkmal $f_{GasPedalDt}$ untersucht. Es erzielt trotz Kombination mit historischen Abtastpunkten einen hohen optimalen Klassifikationsfehler von etwa 50%; die Anreicherung historischer Information bringt hier keine Vorteile.

Interpretation

Mit sinkender TTC sind bessere Schätzungen möglich. Dies wurde erwartet, weil eine niedrige TTC einem hohen Eskalationsgrad entspricht; steigt diese, verringern sich offensichtlich die Handlungsmöglichkeiten des Fahrers, und die Erkennung wird leichter.

Historische Werte können für bestimmte Merkmale zur Klassifikation beitragen. Dies motiviert grundsätzlich die Verwendung spezieller für sequentielle Daten entworfene Klassifikatoren, wie Hidden Markov Modelle. Auch Algorithmen der nicht sequentiellen Mustererkennung können von historischen Merkmalen profitieren, wofür z.B. das *Moving Window* Prinzip verwendet werden kann.

Weil Random Forest dafür bekannt ist, irrelevante Merkmale identifizieren zu können (Kapitel 6.2), kann er also mit historischen Werten befüllt werden. Die resultierende Merkmalsselektion kann ebenso wie der Bayes-Fehler Erkenntnisgewinn über die Bedeutung historische Daten geben - im Vergleich dazu jedoch direkt aus der praktischen Anwendung zur Klassifikation heraus.

8.4 Random Forest zur Fahrerintentionserkennung

Die Klassifikation der Fahrerintention wurde untersucht, um die Bedeutung historischer Daten zu bestätigen. Die vorherige Untersuchung des Bayes Fehlers hatte gezeigt, dass durch historische Signalwerte der Klassifikationsfehler sinken kann; dies wurde nun praxisnahe durch Training und Prädiktion ungesehener Daten nachvollzogen. Parallel wurde das Verhalten des Klassifikators auf verschiedenen Merkmalsmengen sowie die Bedeutung der Merkmalsberechnung betrachtet.

Nach dem manuellen Labelling und der Merkmalsextraktion (Kapitel 5.6) wurde ein Modell trainiert und die Klassen auf ungesesehenen Testdaten prädiziert. Dabei wurden ver-

schiedene Abtastpunkte h (Tabelle 8.1) verwendet, um die Indikatorsignale mittels Moving Window (Kapitel 6.3) durch historische Werte anzureichern.

Abtastpunkte	Bedeutung
h_1 (0.0s)	Keine historischen Informationen, nur aktuelle Messwerte
h_2 (1.0s, 0.5s, 0.1s, 0.0s)	Wenig Historie mit einem zusätzlichen Kurzzeit-Messpunkt des Alters 0.1s; auf diese Art können kurzzeitige Signaländerungen bzw. Gradienten gelernt werden.
h_3 (2.5s, 1.0s, 0.5s, 0.1s, 0.0s)	Mittlere Menge Historie, sinkende Abtastfrequenz bei älteren Werten
h_4 (2.5s, 1.0s, 0.5s, 0.0s)	Wie h_3 ohne den Kurzzeit-Messpunkt, gleichmäßig gesampled
h_5 (5.0s, 2.5s, 1.0s, 0.5s, 0.1s, 0.0s)	Maximale Historie, sinkende Abtastfrequenz bei älteren Werten.

Tabelle 8.1: Abtastzeitpunkte für die Random Forest Klassifikation der Highlevel-Klassen. Die Werte werden für Moving Window invertiert, damit sie Zeitpunkte referenzieren, die bezüglich des Prädiktionszeitpunktes in der Vergangenheit liegen.

Resultate für Highlevel-Manöver

Für die Klassifikation von Highlevel-Manövern waren die Ergebnisse stets optimal, wenn alle verfügbaren Signale genutzt wurden. Dies war bei der bekannten Robustheit von Random Forest zu erwarten und spricht sehr für dessen Anwendung gerade auf komplexen Fahrdaten.

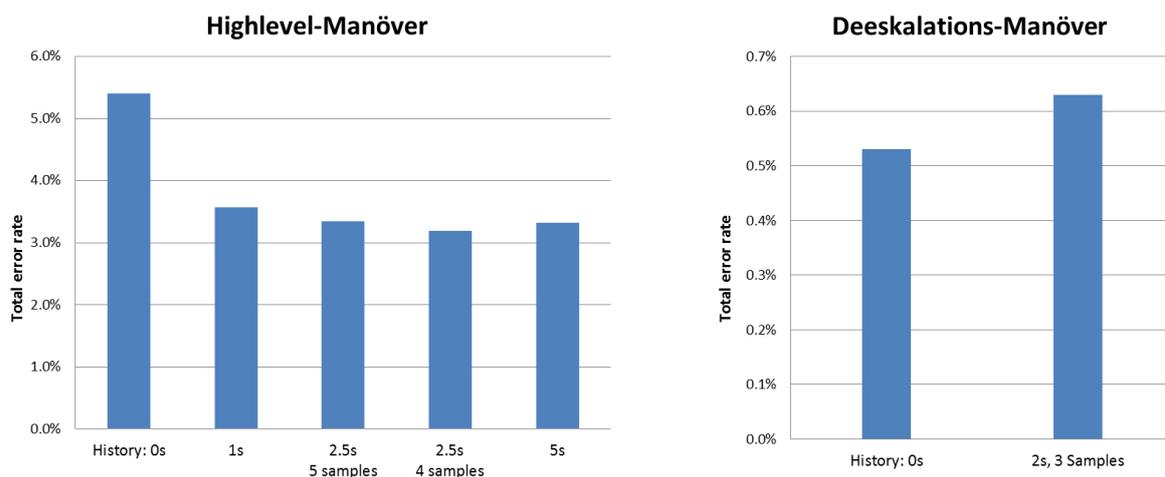


Abbildung 8.4: Resultat des Random Forest Algorithmus für die Prädiktion der Highlevel- und Deeskalationsklassen. Für beide Anwendungen wurden die historischen Abtastpunkte variiert und die Darstellung jeweils auf die aussagekräftigen Settings eingeschränkt. Man beachte die unterschiedliche Skalierung.

Die Fehlklassifikationsrate ohne Verwendung von Historie h_1 lag bei 5.4% (Abb. 8.4). Durch Anreicherung historischer Informationen konnte eine deutliche Verbesserung er-

reicht werden: Mit einem Optimum für die Abtastpunkte h_4 und 2.5s Historie sinkt der Fehler auf 3.2%. Die älteren Abtastpunkte h_5 , jüngeren Abtastpunkte h_1 sowie zu viele historische Abtastpunkte h_3 ergeben wiederum schlechtere Ergebnisse. Die Untersuchung wurde mit weiteren Kombinationen an Abtastpunkten und zunehmender Historie wiederholt, wobei die dargestellten Ergebnisse bestätigt wurden.

Resultate für Deeskalations-Manöver

Die Schätzung der Deeskalationsklassen ergab unterschiedliche Resultate. Ohne Verwendung von Historie wird die Fehlklassifikationsrate 0.53% erzielt (Abb. 8.4 rechts); der Wert ist per se stets sehr gering, da in den meisten Fällen gar keine Eskalation vorliegt.

Unter Verwendung von Historie wurden im Gegensatz zu den Highlevel-Manövern stets schlechtere Ergebnissen erzielt: Z.B. erhöhte sich mit den Abtastpunkten $h_6 = (2s, 1s, 0s)$ der Gesamtfehler auf 0.63%.

Variable Importance

Die *Variable Importance* [Bre01] ist ein empirischer Schätzer für die Relevanz eines Merkmals für einen Klassifikator; er wurde auch für Fahrerassistenzdaten bereits angewendet [TGS⁺08]. Die Werte der Signale werden künstlich verrauscht, indem sie über der gesamten Testdatenmenge permutiert werden. Dies entspricht also einer zufälligen Messung unter Berücksichtigung der ursprünglichen Verteilung vorkommender Werte. Durch Verrauschen eines Merkmals i ergibt sich die neue Klassifikation $\hat{\Psi}_i$. Die Differenz der Klassifikationsrate $\Delta e_i = p(\hat{\Psi} = \Psi) - p(\hat{\Psi}_i = \Psi)$ zwischen der Korrektheit auf nicht-verrauschten Daten und der im Allgemeinen niedrigeren Korrektheit auf den verrauschten Daten ergibt die Variable Importance für das Merkmal i .

Abb. 8.5 zeigt die Variable Importance aller verfügbaren Merkmale für die Klassifikation der Deeskalationsmanöver, ohne Berücksichtigung historischer Daten. Nebenbei wurde dadurch eine hohe Übereinstimmung mit dem vorab untersuchten Mutual Information Merkmalsranking erzielt [HBZ10]; es kann also angenommen werden, dass das trainierte Random Forest Modell die Merkmale optimal verarbeitet hat.

Die Ergebnisse lassen sich wie folgt interpretieren: Das Merkmal f_{TTC} wurde nicht dargestellt, weil es mit $\Delta e = 0.7$ einen extrem hohen Wert hat; Grund dafür ist, dass bei hohen TTCs per definitionem die Neutralklasse vorliegt. Die wichtigsten Merkmale sind danach die Relativgeschwindigkeit $f_{ObjectVxRel}$, der Kursversatz $f_{ObjectDyCirc}$ und der Lenkradwinkelgradient $f_{WheelAngleDt}$. Sie beschreiben offensichtlich die Situation und die Fahreraktion und dienen damit als wertvolle Indicatorsignale, was auch intuitiv nachvollziehbar ist.

Einige Signale wie die Kurskrümmung $f_{KapTrajFast}$ oder die Bremsaktivität $f_{Braking}$ erreichen fast keine Variable Importance: Dies ist ein Beweis dafür, dass die Merkmalsextrak-

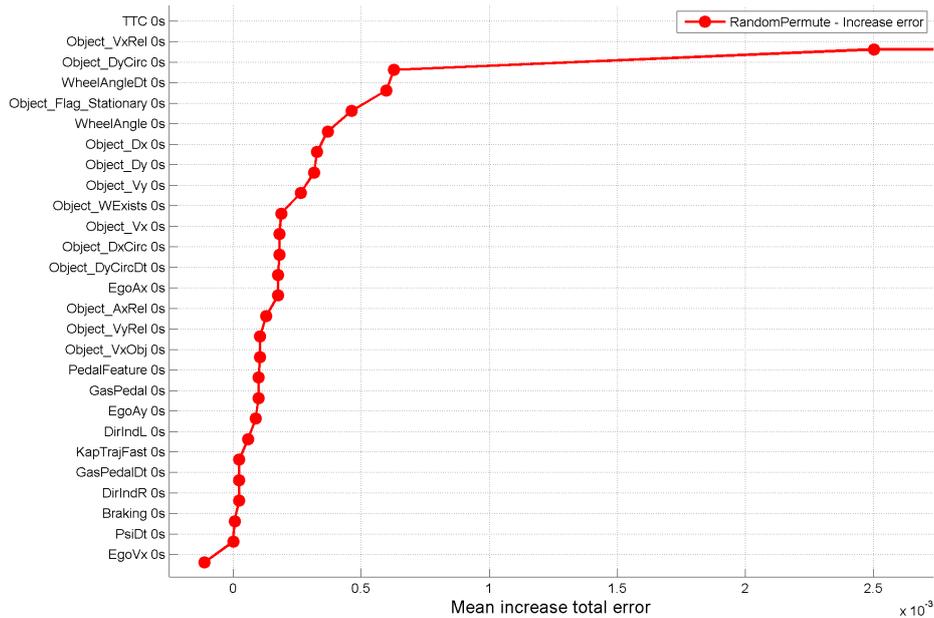


Abbildung 8.5: Variable Importance der verfügbaren Merkmale für die Klassifikation von Deeskalationsmanövern. Das Merkmal f_{TTC} hat per definitionem eine hohe Importance von 0.7 und wurde bewusst nicht dargestellt.

tion erfolgreich war, welche daraus die Features $f_{ObjectDyCirc}$ oder $f_{PedalFeature}$ bestimmen. Gegeben des Kursversatzes trägt die Kurskrümmung selbst also nicht mehr zur Klassifikation bei; ähnliche Aussagen können mit stochastischer Unabhängigkeitsprüfung getroffen werden. Ein Extrembeispiel für diese Beobachtung ist die Egofahrzeuggeschwindigkeit f_{EgoVx} : Sie geht bereits in die relative Geschwindigkeit der Objekte und damit die TTC mit ein, so dass sie mit negativer Δe sogar kontraproduktiv ist.

Die Blinkersignale $f_{DirIndL/R}$ erzielen nur eine geringe Bedeutung, weil die Deeskalationsmanöver in kritischen oder sportlichen Situationen definiert sind. Für die Erkennung von Highlevelmanövern hingegen erzielten die Blinker hohe Werte im Bereich $\Delta e = 0.1$, was den Unterschied der Klassendefinitionen verdeutlicht.

Diskussion

Die verbesserte Erkennung der Highlevelmanöver ist ein starker Motivator für historische Abtastpunkte; bei Deeskalationsmanövern führt genau dies jedoch zu schlechteren Ergebnissen. Diese Verschlechterung ist ein Indiz dafür, dass durch die Moving Window Technik zu viel redundante Information angereichert wurde: Jedes Muster taucht beim Verschieben des Moving Windows nacheinander in jedem Merkmal auf. Je länger das Fenster ist, desto häufiger muss also das exakt gleiche Muster gelernt werden. Um Overfitting zu vermeiden und eine gute Generalisierbarkeit zu erzielen, muss die Wahl der Abtastpunkte somit sehr vorsichtig geschehen.

Highlevel-Manöver werden durch Historie dennoch besser erkannt, was auf Merkmale mit Langzeitwirkung wie dem Blinker zurückgeführt werden kann. Deeskalationsmanöver sind hingegen erst in kritischen Situationen definiert. Sie weisen eine deutlich höhere Dynamik auf, die mit dem Moving Window Prinzip offensichtlich nicht abgebildet werden kann.

8.5 Größe der HMMs und Datenfrequenz

Als erste Untersuchung mit HMMs wurde zunächst eine optimale Abtastfrequenz sowie die Anzahl der Zustände Q bestimmt. In den meisten Arbeiten werden die Zustände so interpretiert, dass sie den sinnhaften Fortschritt der Handlung beschreiben. Aus den Teilschritten „Einleitung, Durchführung, Abschluss“ eines Manövers ergeben sich zum Beispiel 3 Zustände, durch Zwischenschritte oder Vervielfältigung werden häufig auch bis zu 10 Zustände verwendet [BED08]

Darüber hinaus werden bei kontinuierlichen HMMs die Beobachtungswahrscheinlichkeiten meistens durch Gaussian Mixtures modelliert, so dass die Wahrscheinlichkeitsverteilung der Beobachtung o im Zustand q die Form

$$p(o|q) = \sum_{i=1}^M w_i \mathbf{N}(\mu_i, \sigma_i) \quad (8.1)$$

besitzen. Es ergibt sich also als weiterer Variationspunkt die Anzahl M der Mixturen je Zustand. Sie drückt aus, wie stark diversifiziert ein Zustand sein kann: Während eine einzelne Gaussverteilung Daten eines Prozesses mit Erwartungswert und Varianz modelliert, können mit einer Summe von drei Gaussverteilungen auch drei verschiedene Prozesse modelliert werden.

Es wurden verschiedene Kombinationen der Abtastfrequenz, Anzahl Zustände und Gaussian Mixtures untersucht: Mit der entsprechenden Parametrierung wurde ein Ensemble-HMM auf Trainingsdaten trainiert und die Klassifikationsrate auf ungesehenen Testdaten ermittelt. Es wurde das Growing Window Prinzip angewandt, als Sequenzanfang war der Beginn der kritischen Szenen gewählt.

Resultate

Abb. 8.6 zeigt das Ergebnis der Voruntersuchung von Ensemble-HMMs mit verschiedenen Parametrierungen [Ste11]. Zunächst sieht man, dass die Klassifikationsrate asymptotisch mit wachsender Größe des HMMs und zunehmender Anzahl Gaussian Mixtures steigen. Interpretiert man eine Gaussverteilung als Erwartungswert mit Unsicherheit, so kann die Gesamtzahl QM der Gaussverteilungen im HMM als Anzahl Cluster aufgefasst werden, die notwendig sind, um die Daten zu quantisieren, ohne Information über die Fahrerintention zu verlieren.

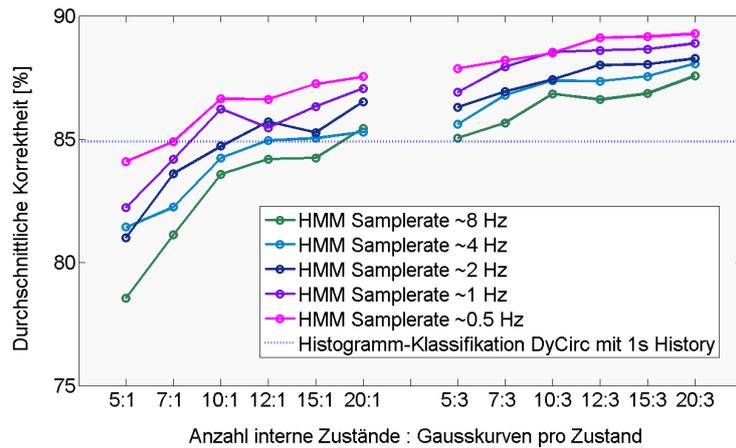


Abbildung 8.6: Voruntersuchung mit Ensemble-HMMs zur Prädiktion der Deeskalationsklassen [Ste11]. Mit zunehmender Anzahl Zustände und Gaussverteilungen steigt auch die Klassifikationsleistung. Auch mit sinkender Abtastrate ω steigt die Performanz kontinuierlich, obwohl zuvor die Datenfrequenz $\omega \approx 2\text{Hz}$ ermittelt wurde. Zum Vergleich wurde die Histogramm-Klassifikation eingezeichnet, mit welcher der Bayes-Fehler ermittelt wurde (Kapitel 6.4).

Offensichtlich besitzen die Daten also eine hohe Diversität. Bei gleicher Gesamtzahl an Gaussverteilungen erzielen Konfigurationen mit weniger Zuständen (und entsprechend mehr Mixturen) bessere Ergebnisse, was man an den Konfigurationen $Q = 5, M = 3$ und $Q = 15, M = 1$ erkennt. Dies deutet darauf hin, dass die Werte der Daten selbst, modelliert durch die Gaussian Mixtures, einen höheren Einfluss als der zeitliche Verlauf haben, welcher durch die Zustände und deren Verbindungen abgebildet wird: Die zusätzlichen Modellparameter der Zustandsübergänge im HMM sorgen also für schlechtere Ergebnisse.

Diese Tendenz passt zu der Beobachtung, dass mit steigender Samplerate die Klassifikationsrate ebenfalls sinkt. Entweder ist die wahre Frequenz der in den Daten enthaltenen Informationen also doch deutlich niedriger, also zuvor in Abschnitt 6.5 ermittelt ($\omega \approx 2\text{Hz}$), oder die Hidden Markov Modelle repräsentieren den komplexen zeitlichen Verlauf der Muster noch nicht ideal. Für letztere Interpretation spricht, dass bei höherer Samplerate definitiv mehr Informationen vorhanden sind als bei zu niedriger Rate. Erzielt das Framework dennoch schlechtere Ergebnisse, muss also eine grundlegende methodische Verbesserung erarbeitet werden.

Fazit

Mit sinkender Frequenz ω steigt die Klassifikationsrate deutlich über die empirisch ermittelte Shannon-Frequenz hinaus. Ähnlich wie bei den Random Forest Untersuchungen mit Historie wird der zeitliche Verlauf der Daten also nicht berücksichtigt bzw. führt sogar zu schlechteren Ergebnissen.

Unter Verwendung eines einfachen Bayes-Klassifikators mit Gaussian Mixtures, welcher ohne Historie aufgesetzt wurde, konnten 88.5% Korrektheit ermittelt werden [Ste11]. Dieser Bayes-Klassifikator kann äquivalent als HMM mit einem einzelnen Zustand und einem Zeitschritt formuliert werden - im Vergleich liefert er also fast genauso gute Ergebnisse wie HMMs mit mehreren Zuständen und Verwendung der historischen Daten.

Mit dieser Interpretation wurde vermutet, dass der Baum Welch Algorithmus noch stark verbessert werden muss, um relevante Zeitpunkte in den Daten zu finden und deren Erkennungsmerkmale durch spezielle Zustände zu modellieren. Die Strukturierungsmaßnahmen wurden dafür als Verbesserungsmöglichkeiten untersucht (Kapitel 8.9).

8.6 Growing Window

Sequenzen für Hidden Markov Modelle wurden für Training und Testphase wahlweise mit Growing Window (GW) oder Moving Window (MW) Prinzips erstellt, so dass insgesamt vier Kombinationen entstehen (Kapitel 7.1). Die Growing-Window Sequenzen beginnen dabei immer fest zu dem Zeitpunkt t_0 , bei welchem die Fahrsituation zum ersten Mal als kritisch eingestuft wird; die Moving-Window Sequenzen beginnen stets eine konstante Zeit vor dem aktuellen Zeitpunkt.

Resultate

Aus Abb. 8.7 wird ersichtlich, dass besonders bei hohen TTCs, also weit vom Objekt entfernt, die Growing Window Prinzipien die besten Klassifikationsraten liefern. Dies lässt sich dadurch interpretieren, dass nur die relevanten Teile der Sequenz berücksichtigt werden oder anders herum, dass die Sequenz tatsächlich keine Information trägt, bevor die kritisch wird. Dadurch bestätigt sich die Grundannahme des *Growing Window* Prinzips, dass alle Zeitpunkte ab Beginn der Szene relevant sind und die Zeitpunkte darüber hinaus keine relevante Information trägt.

Bei geringeren TTCs, also näher am Objekt, verschwinden die Vorteile der Growing Window Varianten: Der kritische Teil der Sequenz ist dann vergleichbar lang wie die Moving Window Sequenzen fester Länge. Somit werden mit beiden Varianten ähnliche Datenszenen zur Klassifikation verwendet, und die Erkennungsraten gleichen sich an.

Fazit

Mit dem Growing Window Prinzip werden offensichtlich genau die relevanten Teile einer Sequenz erfasst, die für die Klassifikation der Deeskalationsmanöver benötigt werden. Sowohl für das Training als auch die Prädiktion werden die Daten also ab dem Zeitpunkt verwendet, bei welchem die Szene zum ersten Mal kritisch wird (Kapitel 5.3).

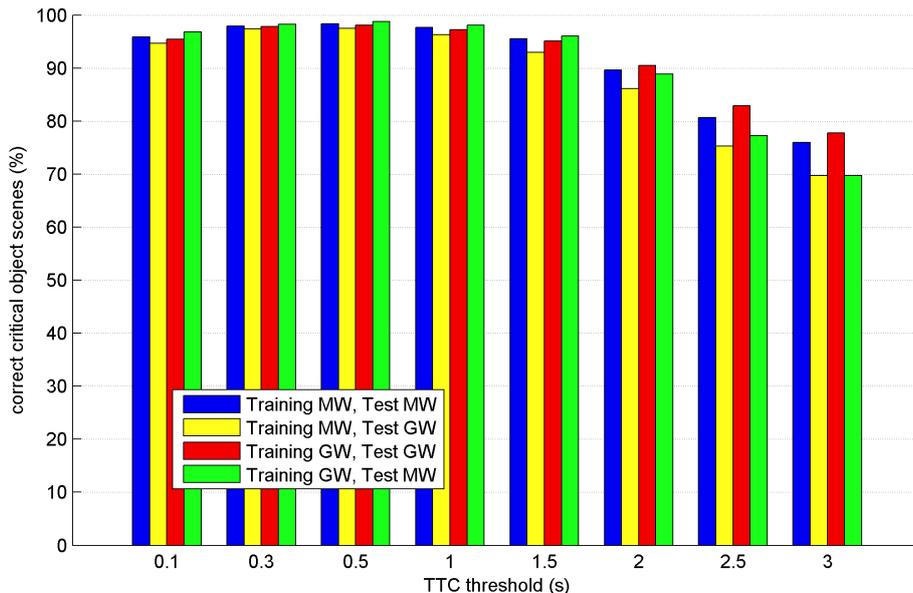


Abbildung 8.7: Klassifikationsraten mit Ensemble-HMMs. Sequenzen wurden für Training und Prädiktion wahlweise mit dem Growing Window (GW) oder Moving Window (MW) Prinzip erstellt [Dit09]

8.7 Klassifikations-HMMs

Nach den Voruntersuchungen wurden nun die neu eingeführten Klassifikations-HMMs (Kapitel 7.2) mit dem bekannten Ensemble Verfahren verglichen. Gleichzeitig wurde die Wahl der Sequenzen untersucht (Kapitel 7.1): In der ersten Variante wurden nur kritische Teilszenen betrachtet, während in der zweiten Variante auf diese Unterteilung verzichtet wurde und ganze Szenen als Sequenzen im HMM verwendet wurden.

Die Beobachtungswahrscheinlichkeiten wurden durch Gaussian Mixtures mit $M = 3$ Mixturekomponenten modelliert. Alle Modelle wurden randomisiert initialisiert, um den Fokus auf dem Vergleich der Verfahren zu belassen, und mit dem Baum Welch Algorithmus trainiert. Für reproduzierbare Ergebnisse wurden die Auswertungen unter Variation der Zustandsanzahl Q wiederholt.

Resultate

Die Ergebnisse sind in Abb. 8.8 dargestellt. Das Ensemble mit insgesamt $Q = 50$ Zuständen erreicht den geringsten Klassifikationsfehler von 24%, wenn nur kritische Teilszenen verwendet werden. Für jede der 5 Klassen besitzt es ein Sub-HMM mit jeweils 10 Zuständen. Das neu eingeführte Klassifikations-HMM erreicht bei derselben Zustandsanzahl ebenfalls seine beste Leistung, jedoch geringfügig schlechter als das Ensemble-Verfahren. Unter Rücksicht auf die Randomisierung und bei Betrachtung der variierenden Zustandsanzahl erreichen das Ensemble Verfahren und das neue Klassifikation-HMM in Summe ähnliche Leistungen.

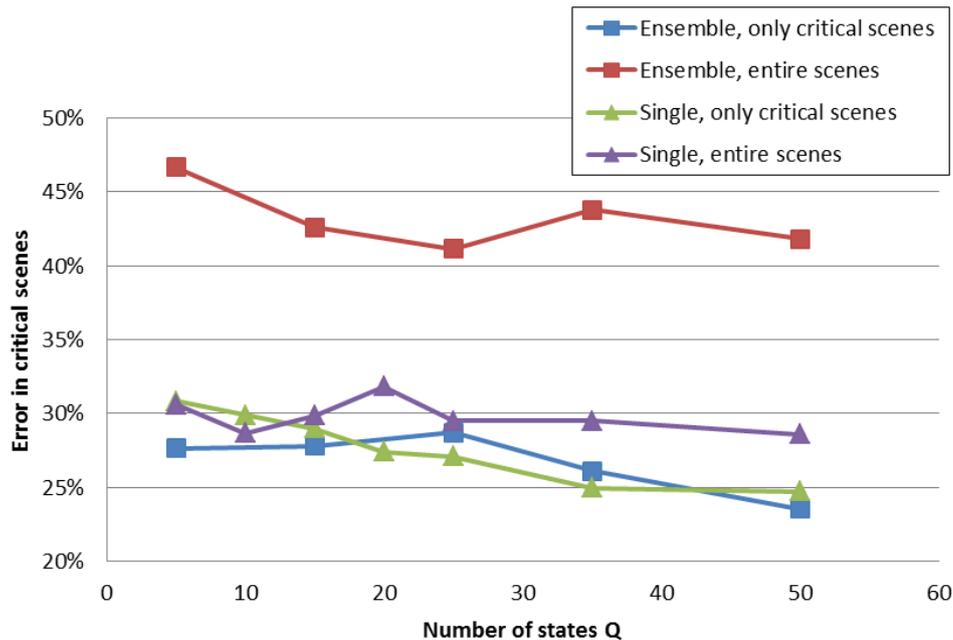


Abbildung 8.8: Klassifikationsraten mit einem HMM Ensemble und dem kombinierten Klassifikations-HMM (Single). Neben der Wahl der Trainingssequenzen wurde für eine bessere Vergleichbarkeit die Gesamtanzahl an Zuständen variiert, da diese offensichtlich einen hohen Einfluss auf das Ergebnis hat.

Bei Verwendung der gesamten Sequenzen erzielen die Ensemble-Verfahren wie erwartet extrem schlechte Ergebnisse, weil sie keinen Klassenwechsel zulassen. Die Klassifikations-HMMs sind hier nur 4% schlechter als auf den kritischen Teilszenen; Klassenwechsel werden also nun mit geringen Performancefehlern ermöglicht. Da die Verwendung der Gesamtsequenz grundsätzlich der einfachere und auch universellere Ansatz ist, ermöglicht durch Klassifikations-HMMs, ist dieser Nachteil gegen den Vorteil der besseren Verwendbarkeit abzuwägen.

Fazit

Die in dieser Arbeit vorgestellten Klassifikations-HMMs arbeiten vergleichbar gut wie Ensemble-HMMs, stellen dabei aber den allgemeineren Ansatz dar: Ihre Struktur ist einfacher, und die Daten müssen nicht mehr in Teilszenen eingeteilt werden.

Beide Verfahren wurden zwar randomisiert initialisiert; jedoch entsprechen die Ensemble-HMMs implizit einer hierarchischen HMM-Struktur mit einer Komponente je Klasse. Die geringen Vorteile dieser Untersuchung sind auf diese Struktur zurückzuführen. Sie motivieren dazu, die Klassifikations-HMMs ebenfalls durch eine Strukturierung zu verbessern, um letztendlich auch einen messbaren Vorteil zu erzielen (Kapitel 8.9).

8.8 Beobachtungen mit Random Forest

In dieser Untersuchung wurde nun Random Forest verwendet, um die Beobachtungswahrscheinlichkeiten des HMMs zu verbessern (Kapitel 7.3), und mit den üblicherweise verwendeten Gaussian Mixtures verglichen. Zunächst wurde dabei ein einzelnes, komplexes Random Forest Modell trainiert, welches gegeben der Beobachtung die Verteilung des aktiven Zustandes schätzt. Bei der zweiten Variante wurde für jeden Zustand ein eigenes Modell trainiert, welches gegeben der Beobachtung die Zustandsaktivität schätzt (Random Forest X).

Vergleich Random Forest mit Gaussian Mixtures

Schon auf Anhieb liefert die Ergebnisübersicht mit Random Forest ein zu Gaussian Mixtures völlig unterschiedliches Bild (Abb. 8.9). Mit einer Fehlerrate von 18.5% bei Verwendung von Teilsequenzen sind die Ergebnisse per se deutlich besser als in der Konfiguration mit Gaussian Mixtures: Letztere liefert bei derselben Zustandsanzahl 29% Fehlerrate.

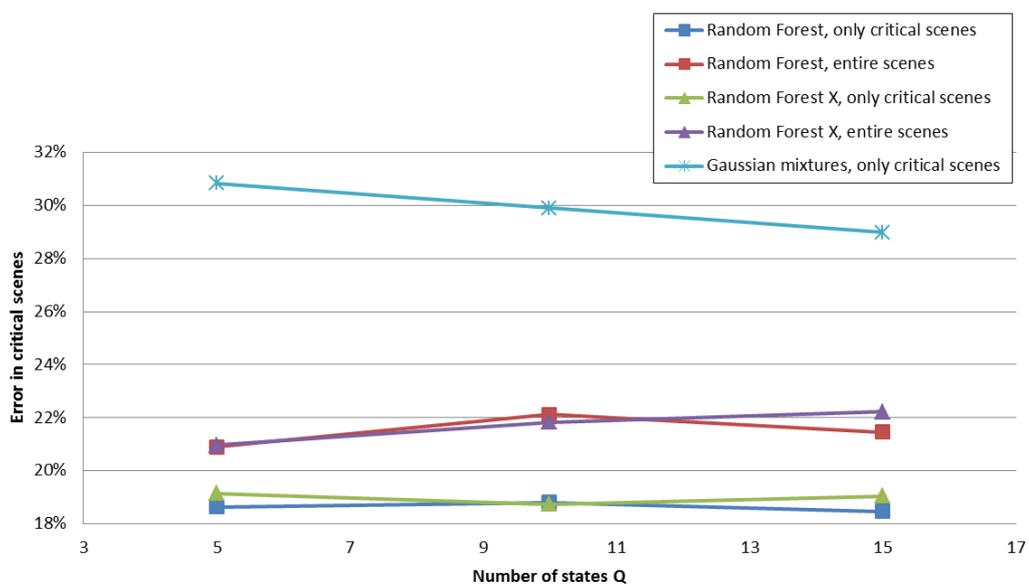


Abbildung 8.9: Klassifikationsraten mit einem einzelnen Random Forest Modell für alle Zustände (Random Forest) oder einem Modell je Zustand (Random Forest X). Zum Vergleich wurde die beste Konfiguration mit Gaussian Mixtures dargestellt, die deutlich schlechtere Ergebnisse liefert.

Obwohl die beiden Random Forest Varianten per Konstruktion sehr unterschiedlich sind, ergeben sich hier fast identische Resultate. Bei Verwendung der Teilszenen wird eine Klassifikationsrate nahezu unabhängig von der Zustandsanzahl erreicht; dieses Phänomen wird in Kapitel 8.9.3 detailliert ausgewertet und diskutiert. Auf den Gesamtszenen liefern beide Varianten wie schon bei auch bei Gaussian Mixtures deutlich mehr Fehlklassifikationen, die sich mit steigender Zustandsanzahl weiter verschlechtern.

Random Forest Beobachtungen im Detail

Im zweiten Versuch wurde der in dieser Arbeit vorgestellte Ansatz untersucht, gewichtete Trainingslabels durch Reproduktion der Trainingsbeispiele zu simulieren (Kapitel 7.6); ansonsten können Gewichte der Trainingsdaten durch Random Forest nicht berücksichtigt werden. Bei der Verwendung eines einzelnen Random Forest Modells wurde zusätzlich der Einsatz der Bayes-Formel verifiziert (Kapitel 7.3.1).

Abb. 8.10 zeigt die Ergebnisse der systematischen Untersuchung der erarbeiteten Möglichkeiten, die Beobachtungswahrscheinlichkeiten zu modellieren. Die beste Klassifikation mit einer Fehlerrate von 18,3% wird erhalten, indem ein Random Forest Modell je Zustand trainiert wird (Random Forest X) und die gewichteten Trainingsdaten zuvor reproduziert werden (Weighted).

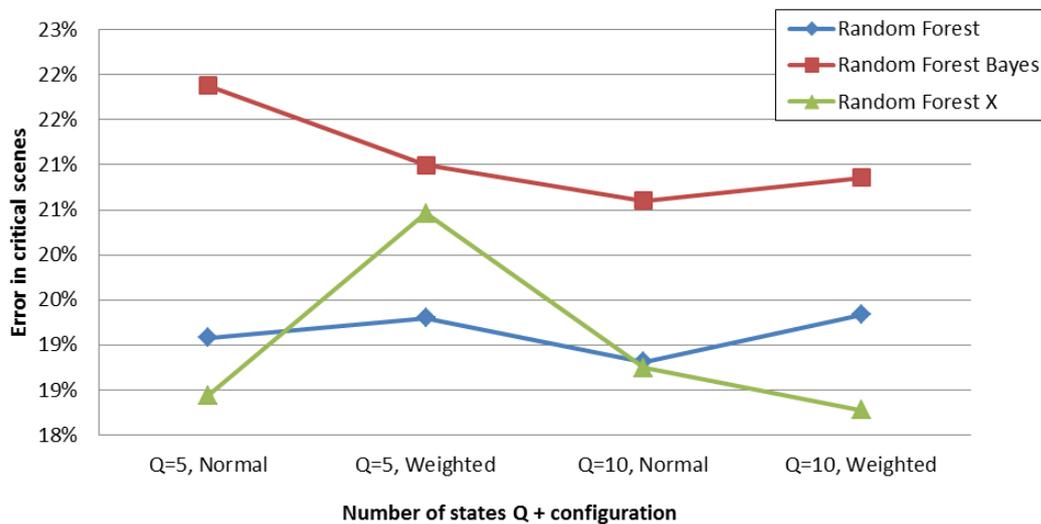


Abbildung 8.10: Variationen der Anzahl Zustände und der detaillierten Modellierung der Beobachtungswahrscheinlichkeiten und deren Auswirkung auf die Klassifikationsrate

Die Verwendung der Bayes-Formel, welche die von Random Forest geschätzten Zustandswahrscheinlichkeiten in Beobachtungswahrscheinlichkeiten umrechnet, resultiert stets in schlechteren Klassifikationsergebnissen. Gewichtete Trainingsdaten (Weighted) verursachen bei Random Forest stets ein schlechteres Ergebnis, bei Random Forest X hängt das Ergebnis von der Zustandsanzahl ab.

Analyse der gelernten Verteilungen

Zum detaillierten Vergleich der Beobachtungswahrscheinlichkeiten $p(o|q)$ wurden Klassifikations-HMM mit 5 Zuständen untersucht, also mit genau einem Zustand je Klasse. Für jeden Zeitpunkt in den Testdaten wurde die Wahrscheinlichkeit bestimmt. Darauf wurde eine einfache Kerndichteschätzung mittels eines Histogramms mit 20 Bins durch-

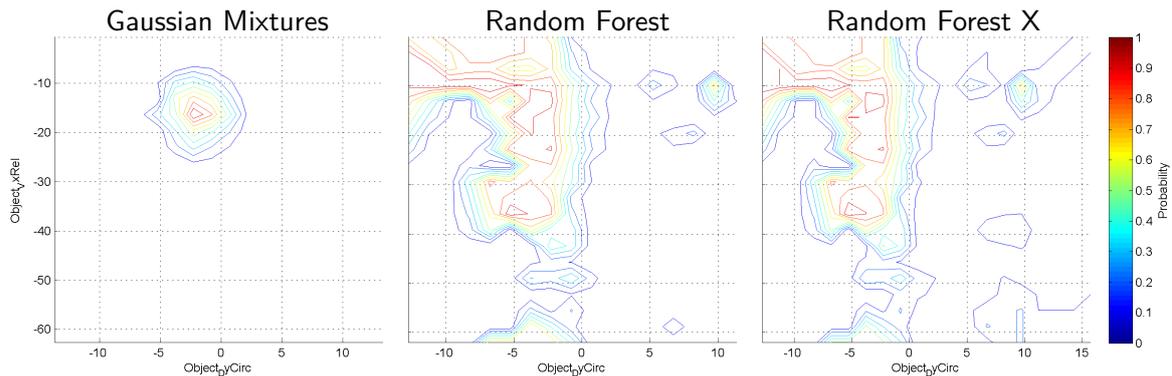


Abbildung 8.11: Beobachtungswahrscheinlichkeiten $p(o|q)$ bezüglich der Merkmale Kursversatz und Relativgeschwindigkeit bei Verwendung unterschiedlicher Modellierungen (Zeilen). Für Gaussian Mixtures wurden $M = 3$ Komponenten trainiert

geführt, was schließlich eine empirische Darstellung der Beobachtungswahrscheinlichkeiten ergibt.

Abb. 8.11 zeigt exemplarisch die resultierenden Unterschiede anhand des Zustandes für die Klasse Ausweichen Links und zweier Merkmale. So modellieren die Gaussian Mixtures für die Lateralposition des Objektes nur einen starken Peak im Bereich $f_{DyC} \approx -2m$; dies sind genau die Situationen, in denen das Objekt ohnehin schon rechts des Fahrzeugkurses liegt. Scheinbar entgegen der Erwartung liegen alle 3 Gaussian Mixtures aufeinander, was aber dadurch erklärt werden kann, dass die Diversifizierung in den nicht abgebildeten Dimensionen liegt.

Die Random Forest Beobachtungen lassen hier viel mehr Kombinationen zu; etwa kann der Zustand auch aktiv werden, wenn das Objekt noch links vom Fahrzeugkurs liegt ($f_{DyC} > 0$). In solchen Situationen steuert der Fahrer später noch auf die andere Seite, was genau die schwierigen, nicht trivialen Fälle sind. Gleichzeitig wird eine größere, gleichmäßige Streuung der Relativgeschwindigkeit f_{VxRel} modelliert.

Zudem sieht man, dass die Random Forest X Beobachtungen noch geringfügig komplexer sind als bei Random Forest.

Vergleich der Verteilungen

Da Random Forest mächtigere Modelle als Gaussian Mixtures erzeugen kann, fallen auch die $p(o|q)$ komplexer aus. Die Verteilungen sind breiter und liefern eine deutlich höhere Klassifikationsleistung (Kapitel 8.8). Random Forest berücksichtigt also vor allem Zusammenhänge zwischen den einzelnen Merkmalen besser.

Die generative Eigenschaft der Gaussian Mixtures wird in dieser Betrachtung sehr deutlich: Diese passen sich den Werten der Trainingsdaten an und modellieren nur geringe Varianzen. Random Forest hingegen arbeitet nicht auf den genauen Werten selbst, son-

den wählt darauf die optimalen Klassifikationsgrenzen. Die Gestalt der Wahrscheinlichkeitsfunktion ist damit nicht direkt von der Verteilung der Trainingsdaten beeinflusst; im Ergebnis sieht man dies an einer viel breiteren Verteilung.

Random Forest X kann die komplexesten Verteilungen erzeugen, weil die $p(o|q)$ je Zustand separat trainiert werden. Dies ist vorteilhaft, wenn eine Beobachtung o in verschiedenen Klassen vorkommen kann: Während Random Forest o tendenziell dem wahrscheinlichsten Zustand zuordnet, können mit Random Forest X mehrere Zustände o akzeptieren. Im Falle eines Ausweichens nach links oder rechts sind während der Annäherung z.B. die Indikatoren signale sehr ähnlich, obwohl die Fahrerintention sich unterscheidet (Kapitel 3.1). Mehrere Zustände im HMM müssen somit aktiv sein, bevor ein signifikantes Signal letztlich eine eindeutige Schätzung zulässt.

Zusammenfassung

Random Forest kann die komplexen Beobachtungen der Fahrdaten deutlich besser modellieren kann als Gaussian Mixtures. Dies ist zum Teil dadurch verursacht, dass die verwendeten Signale diskret oder sogar binär sind, was in der Praxis zu einem Degenerieren der Gaussverteilungen führte (Kapitel 7.3). Zusätzlich weisen die Signale systematische Abhängigkeiten z.B. von der Egogeschwindigkeit auf, die offensichtlich durch Random Forest besser abgebildet werden konnten.

Die Nachteile durch Verwendung der Bayes Umkehrformel können wie folgt erklärt werden: Da zum Training die korrigierten Posteriors verwendet werden, spiegeln sich die Klassenwahrscheinlichkeiten $p(c)$ bereits in der Aktivität der Zustände wieder. Diese Aktivität ist das Trainingssignal für Random Forest; ein zweites Hinzufügen der $p(c)$ ist damit nicht mehr gerechtfertigt.

Abschließend muss bemerkt werden, dass gegenüber der Klassifikation der Deeskalationsmanöver mit Random Forest ohne HMMs noch keine Verbesserung vorliegt. Da zudem mit steigender Zustandsanzahl als auch bei Verwendung der Gesamtszenen die Performanz nicht wie erwartet ansteigt, liegt offensichtlich noch eine systematische Schwäche vor. Dies motiviert eine genauere Betrachtung der Ergebnisse sowie manuelle Initialisierung in den folgenden Auswertungen.

8.9 Struktur des HMMs

Durch die vorherige Modellierung der Beobachtungswahrscheinlichkeiten von HMMs mit Random Forest konnten große Verbesserungen der Klassifikation erzielt werden. Weil diese jedoch den Random Forest Algorithmus allein noch nicht übertrafen, wurden Maßnahmen zur Verbesserung der HMM-Struktur untersucht. Abschließend wurden die Zustände

einzelnen detailliert betrachtet, um Rückschlüsse auf deren Bedeutung für die sequentiellen Daten zu ziehen.

8.9.1 Vermeidung isolierter Zustände

Zur Vermeidung isolierter Zustände wurden die in Kapitel 7.4.1 vorgestellten Maßnahmen in den folgenden Kombinationen untersucht:

- Minimale Transitionswahrscheinlichkeit $c_a = (0, 0.1, 0.5, 1)$
- Minimale Beobachtungswahrscheinlichkeit $c_o = (0, 0.1, 0.5)$
- Anzahl Zustände $Q = (5, 10)$

Die Anzahl der Zustände Q wurde gering gehalten, um die Effekte der Parameter zu betonen. Mit den Parametern c_a oder $c_o = 0$ liegt kein Unterschied zum üblichen Verfahren vor; der Wert 1 gestaltet die entsprechende Transition bzw. Beobachtung offensichtlich als trivial und macht besonders für letztere keinen Sinn.

Die Modellierung der Beobachtungswahrscheinlichkeiten wurde durch ein einzelnes Random Forest Modell gestaltet (Kapitel 7.3.1). Mit einem Random Forest Modell je Zustand (Kapitel 7.3.2) entstanden vergleichbare Ergebnisse.

Resultate

Die Resultate in Abb. 8.12 zeigen, dass eine Verwendung des Parameters c_o (MinProb) die Ergebnisse grundsätzlich verschlechtert. Eine Anpassung der c_a (Minimum global alpha) hingegen erzielt Verbesserungen: Mit 5 Zuständen sinkt der Prädiktionsfehler für den Extremwert $c_a = 1$ von 19% auf 16.2%. Bei 10 Zuständen wird ein Optimum von 16% Fehlerrate für $c_a = 0.5$ erreicht.

Diskussion

Bereits die geringe Anpassung der Beobachtungswahrscheinlichkeiten von $c_o = 0.1$ überschreibt die Random Forest Schätzungen, was als Indiz für deren hohe Wichtigkeit interpretiert werden kann. Die Maßnahme selbst bringt außer dieser Erfahrung offensichtlich keinen Nutzen.

Das HMM mit 5 Zuständen erzielt die besten Ergebnisse, wenn alle Transitionswahrscheinlichkeiten $p(q \rightarrow q') = 1$ sind. In diesem Fall verliert das HMM seine sequentielle

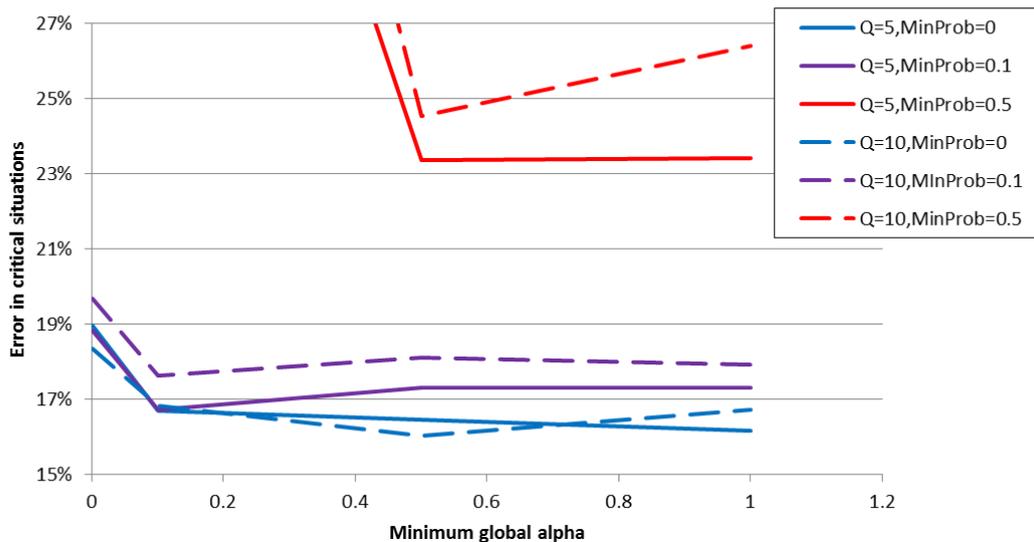


Abbildung 8.12: Anpassung der Transitions- und Beobachtungswahrscheinlichkeiten (Minimum global alpha bzw. MinProb), um isolierte Zustände zu vermeiden. Beide Parameter wurde je einmal mit 5 und 10 Zuständen in verschiedenen Kombination ausgewertet.

Bedeutung vollständig und klassifiziert äquivalent zu Random Forest, alleinstehend angewandt. Offensichtlich reicht ein einzelner Zustand je Klasse also nicht aus: Unter Berücksichtigung der Markov-Annahme kann die Zustandsverteilung des Zeitpunktes $t - 1$ für die Schätzung zum Zeitpunkt t nur beisteuern, welche Klasse zuletzt geschätzt wurde, sonst aber keine weiteren Informationen.

Das HMM mit 10 Zuständen zeigt ein Positivbeispiel dieses Ergebnisses. Es besitzt je Klasse 2 Zustände; die Zustandsverteilung vom Zeitpunkt $t - 1$ gibt also als zusätzliche Information an den Zeitpunkt t weiter, welche Klasse geschätzt wurde und in welcher der 2 Ausprägung. Die mittlere Anpassung der Transitionen mit $c_a = 0.5$ erzielt ein Optimum, und insbesondere bessere Ergebnisse als bei $c_a = 1$. Es muss also Transitionen mit $a_{i,j} > 0.5$ geben, welche die Klassifikation echt verbessern; genau diese beweisen, dass Information über den zeitlichen Ablauf der Beobachtungen genutzt werden.

8.9.2 Variation der Zustandsverweildauern

Die Zustandsverweildauern wurden wie in Kapitel 7.4.2 beschrieben initialisiert, indem die Selbsttransitionen $p(q \rightarrow q)$ gemäß einem vorgegebenen Erwartungswert d gesetzt wurden. Um den Fokus der Auswertung auf diesem Variationspunkt zu lassen, wurden die übrigen Transitionen und Startwahrscheinlichkeiten randomisiert initialisiert.

Während des Trainings wurde die Modellstruktur fixiert und nur die Beobachtungswahrscheinlichkeiten $p(o|q)$ trainiert. Hierfür wurde ein Random Forest Modell je Zustand verwendet. Der Versuch wurde mit variierender Zustandsanzahl wiederholt.

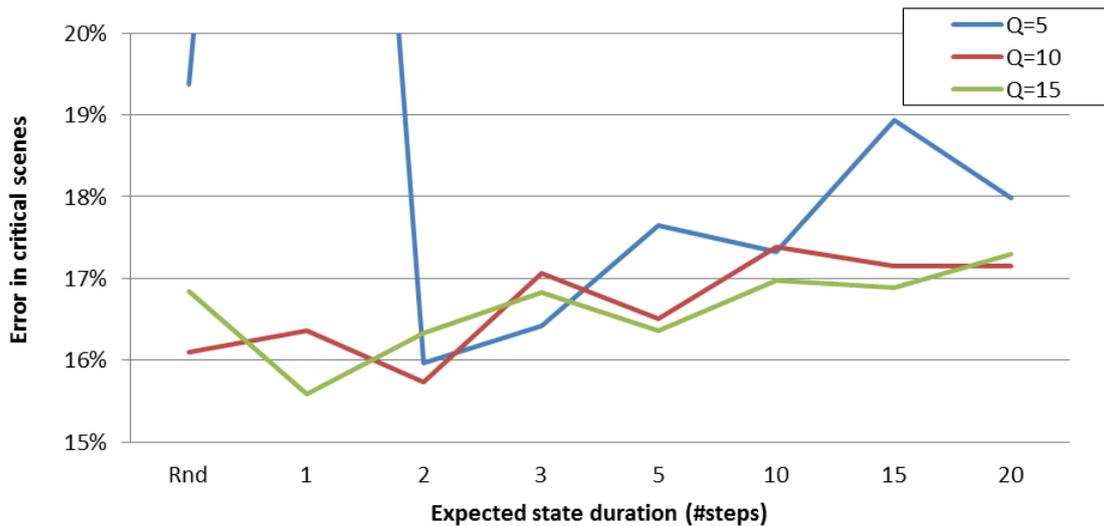


Abbildung 8.13: Variation der Zustandsverweildauern durch Anpassung der Selbsttransition. Bei der Messreihe *Rnd* fand keine Anpassung statt, somit war das gesamte Modell randomisiert. Diese Randomisierung in den übrigen Transitionen erklärt die Streuung der Resultate.

Resultate

Durch die randomisiert und anschließend fixierten Transitionen entsteht in der Auswertung eine Streuung, die jedoch das Ergebnis nicht verschleiert: Mit zunehmenden Zustandsverweildauern steigt der Klassifikationsfehler um etwa 2% an. Für $Q = 5$ besitzt das Modell je Klasse genau einen Zustand, was den besonders hohen Klassifikationsfehler von 34% bei der Verweildauer $D = 1$ erklärt: In diesem Fall ist $\forall q : p(q \rightarrow q) = 0$, wodurch nach jedem Zeitschritt ein Zustand- und damit Klassenwechsel erzwungen wird.

Für $Q \geq 10$ ergibt sich für $D = 1, 2$ eine geringfügige Steigerung der Größenordnung 0.5%, die jedoch durch die Randomisierung bedingt sein kann. Das Experiment müsste für eine signifikante Aussage also mehrfach wiederholt werden. Die Anpassung der Zustandsverweildauern verspricht jedoch keinen nennenswerten Nutzen und wurde damit nicht weiter untersucht.

8.9.3 Untersuchung der trainierten Zustände

Zur Untersuchung des resultierenden HMMs wurden die trainierten Zustandsverweildauern und Beobachtungswahrscheinlichkeiten $p(o|q)$ betrachtet. Weil besonders letztere aufgrund der hohen Dimensionalität nur schwer verglichen werden können, wurde für jeden Zustand nur der Erwartungswert der Beobachtungen $\mathbb{E}(p(o|q))$ betrachtet. Für zwei Zustände q, r beschreibt dann die Mahalanobis-Distanz dieser Erwartungswerte die Ähnlichkeit der Zustände $d(q, r)$. Die paarweise Darstellung dieser Distanzen visualisiert, wie spezialisiert ein Zustand ist.

Resultate

In dieser Analyse wurden HMMs mit Gaussian Mixtures Beobachtungen gegenüber Random Forest X Beobachtungen verglichen. Mit Gaussian Mixtures erreichen die Zustände eine mittlere Verweildauer von etwa $\mathbb{E}(D) = 7$ Zeitschritten, was der Dauer $0.7s - 1s$ entspricht (Abb. 8.14 links). Dieser Wert ist plausibel, da mit dem *Shannon-Theorem* für das Fahrerverhalten eine Abtastezeit von etwa $0.5s$ geschätzt wurde (Kapitel 6.5). Die Mahalanobis-Distanz weist fast überall einen Wert > 0 auf (Abb. 8.14 rechts); die Zustände sind also zu hohem Grad auf jeweils bestimmte Situationen spezialisiert.

Ein völlig anderes Bild zeigt sich für die Random Forest Beobachtungen (Abb. 8.15): Die Zustände erreichen eine geringe Verweildauer von $\mathbb{E}(D) = 1.2$ Zeitschritten, dafür weist die Ähnlichkeitsmatrix weit weniger Unterschiede auf: Nur die Zustände 1...10 unterscheiden sich, diese sind der Neutralklasse zugeordnet. Die Zustände 11...20, 21...30 gehören jeweils zu derselben Klasse, und deren Beobachtungswahrscheinlichkeiten gleichen einander völlig.

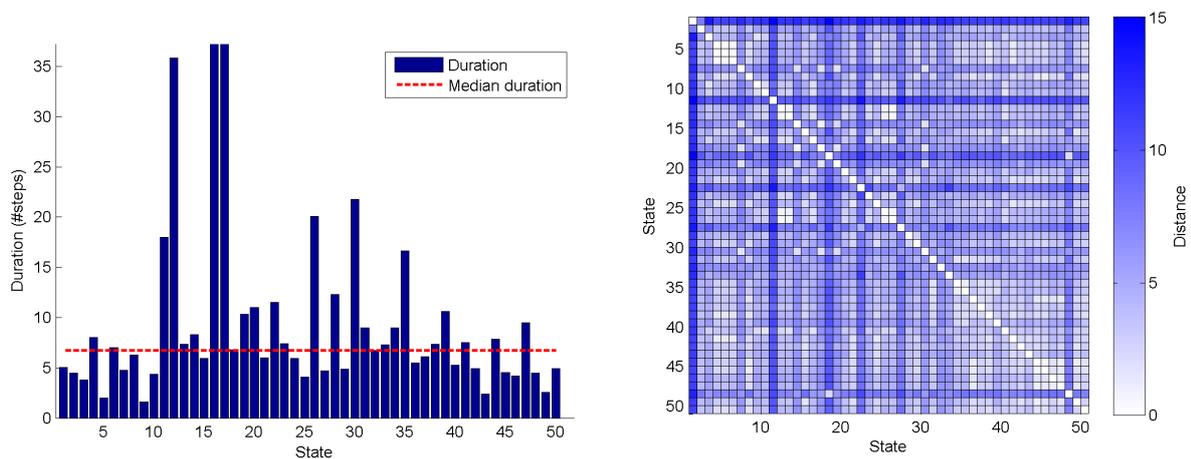


Abbildung 8.14: Analyse des Klassifikations-HMMs mit $Q = 50$ Zuständen und Gauss Beobachtungen, jeweils $M = 3$ Mixturen je Zustand. Links: Erwartete Zustandsverweildauern. Rechts: Ähnlichkeitsmatrix der Beobachtungswahrscheinlichkeiten je Zustand, dargestellt als die Mahalanobis-Distanz der Erwartungswerte.

Diskussion

Obwohl die Random Forest Modelle deutlich bessere Klassifikationsraten erzielen, scheint die HMM Struktur nicht wirklich ausgenutzt zu werden: Die Zustände der nicht neutralen Klassen weisen jeweils dieselben Beobachtungswahrscheinlichkeiten auf, und die Verweildauern sind sehr gering. Das HMM nutzt also die sequentielle Struktur der Daten nicht aus.

Diese Hypothese konnte bestätigt werden, indem die Transitionen in einem separaten Versuch randomisiert blieben: So wurden nur die Beobachtungswahrscheinlichkeiten, nicht aber die Modellstruktur trainiert, und es ergaben sich bessere Resultate. Als Fazit sollte

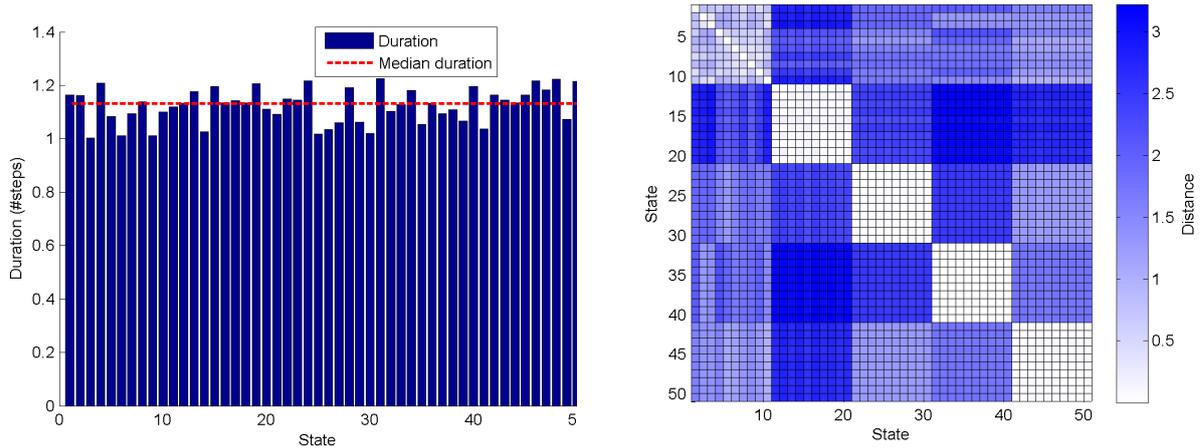


Abbildung 8.15: Analyse des Klassifikations-HMMs mit $Q = 50$ Zuständen und Random Forest X Beobachtungen. Links: Erwartete Zustandsverweildauern. Rechts: Ähnlichkeitsmatrix der Beobachtungswahrscheinlichkeiten je Zustand, dargestellt als die Mahalanobis-Distanz der Erwartungswerte.

also der Baum Welch Algorithmus so erweitert werden, dass die Zustände eine höhere Diversität erhalten und die trainierten Transitionen eine sinnvolle Struktur und Verweildauer ermöglichen.

8.10 Kursprädiktion mit Random Forest

Nach der ausführlichen Betrachtung der diskreten Fahrerintentionsschätzung wird nun die Kursprädiktion ausgewertet. Sie arbeitet auf der kontinuierlichen Größe des zukünftigen Egofahrzeugkurses und unterscheidet sich grundsätzlich von der Prädiktion der Deeskalationsmanöver, weshalb zunächst eine unabhängige Auswertung erfolgt.

Zunächst wurden für den Prädiktionshorizont $\Delta t = 2s$ die Parameter (a, κ) für das zirkuläre Kursmodell ermittelt. Sie wurden im Anschluss mit den untenstehenden Varianten transformiert (Kapitel 5.5), in 21 Klassen diskretisiert (Kapitel 7.7.1) und mit Random Forest prädiziert:

1. Direkte Prädiktion der Parameter (a, κ) ohne Transformation
2. Normalisierung der Kurskrümmung $(a, a_{y,Rel})$
3. Differenzkodierung und Normierung der Kurskrümmung $(\Delta a, \Delta a_{y,Rel})$
4. Transformation in Zielkoordinaten (x, y)
5. Zirkuläre Kursprädiktion unter der Annahme konstanter Längsbeschleunigung und Kurskrümmung, ohne Maschinelles Lernen

6. Untersuchung des Diskretisierungsfehlers bei Strategie 2: Die wahren Modellparameter ($a, a_{y,Rel}$) wurden diskretisiert, was einer bestmöglichen Parameterschätzung durch einen Klassifikator entspräche.

Für die Auswertung wurde die resultierende A-Posteriori-Verteilung der Kursparameter zu einer deterministischen Prädiktion vereinfacht, indem jeweils die Parametrierung mit der höchsten Wahrscheinlichkeit verwendet wurde.

Resultate

Die gemittelten Fehler der Kursprädiktion sind in Abb. 8.16 dargestellt. Die Differenzkodierung der normierten Kursparameter ($\Delta a, \Delta a_{y,Rel}$) erzielt für alle Kriterien die geringsten Fehler: Der mittlere Kursabstand liegt bei 11cm, was eine deutliche Verbesserung gegenüber der zirkulären Kursprädiktion mit 18cm darstellt. Eine ähnliche Verbesserung liegt bei Betrachtung des RMSE (Root of mean squared error) mit 29cm versus 49cm vor; besonders schlechte Prädiktionen, die vornehmlich in dynamischen Fahrsituationen auftraten, werden also ebenfalls vermindert.

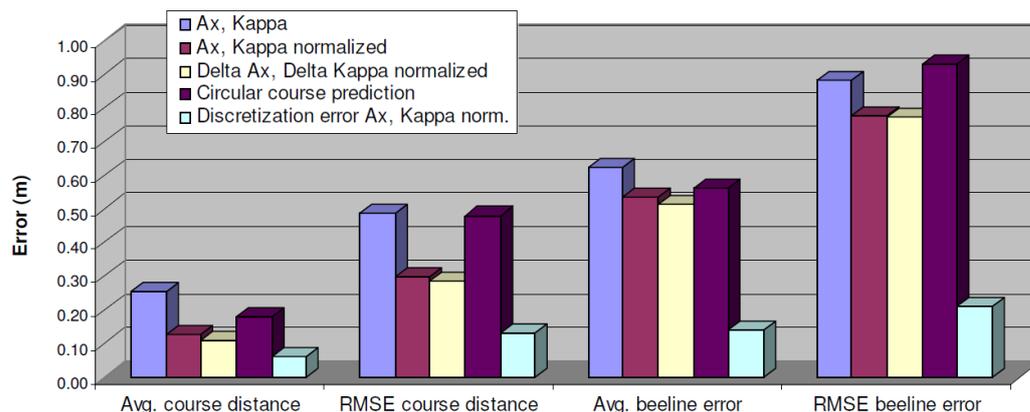


Abbildung 8.16: Ergebnisse der Kursprädiktion über einen Zeitraum von $\Delta t = 2s$. Ermittelt wurde die mittlere Kursdistanz sowie der Luftlinienfehler (beeline) für die im Text beschriebenen Strategien. Strategie 4 (Transformation in Zielkoordinaten) ist wegen zu hoher Fehlerraten nicht abgebildet.

Der Benefit relativiert sich etwas, wenn der Luftlinienfehler miteinander verglichen wird: Dieser beträgt im Mittel 51cm versus 56cm, woraus geschlossen werden kann, dass die Verbesserung vor allem in der Prädiktion der Lateralbewegung liegt. Dennoch werden auch große Schätzfehler der Längsbewegung vermindert, was sich in der Auswertung des RMSE des Luftlinienfehler zeigt; hier liegt mit 77cm versus 93cm eine deutliche Verbesserung vor.

Die Transformation in Zielkoordinaten kann mit einem mittleren Luftlinienfehler von 2.48m als Ausreißer angesehen werden; das Ergebnis ist deshalb nicht abgebildet, und die Transformation wurde nicht weiter verfolgt. Die Prädiktion der Kursparameter (a, κ) ohne Transformation erzielt sogar größere Fehler als die einfache zirkuläre Prädiktion,

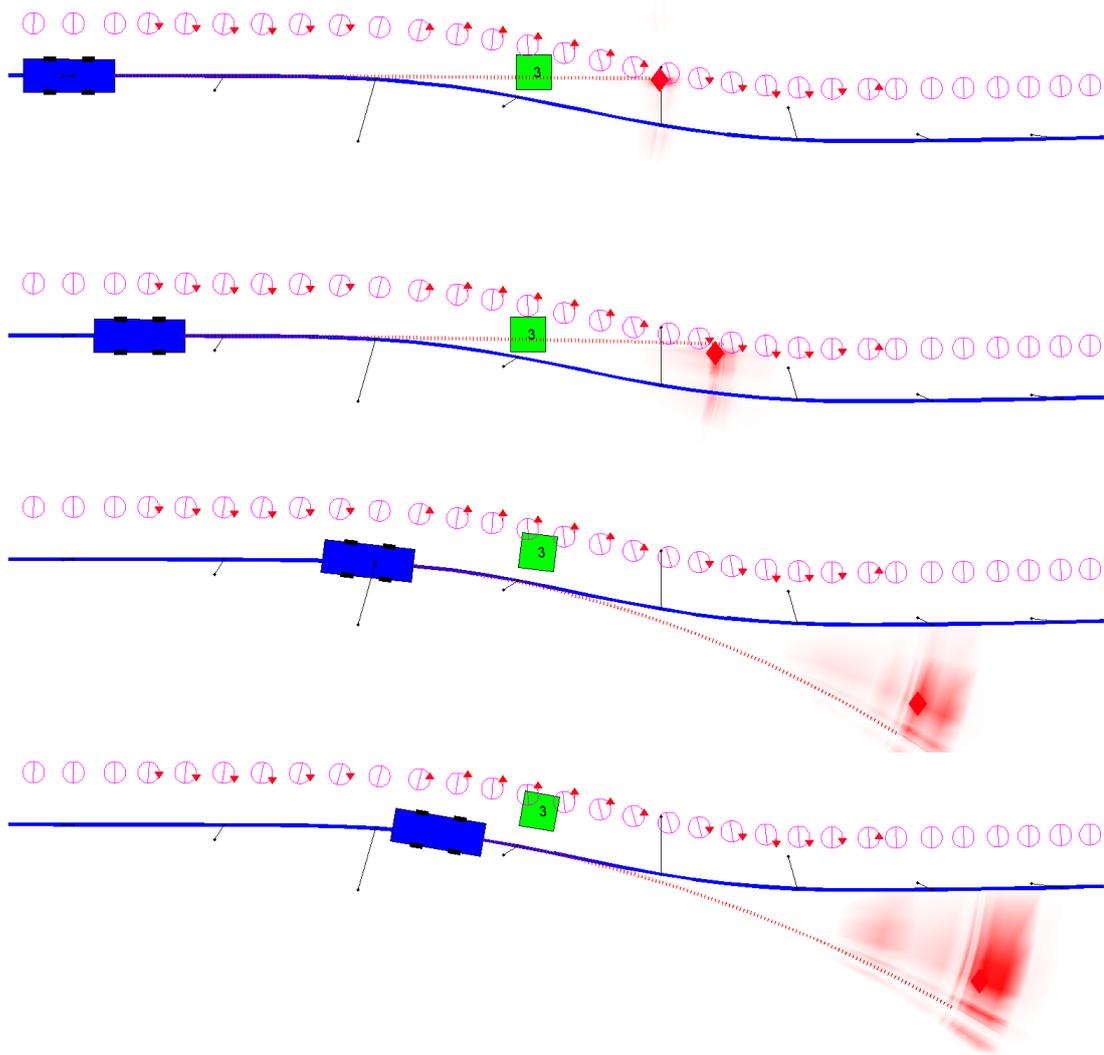


Abbildung 8.17: Probabilistische Kursprädiktion mit Random Forest. In dieser Szene weicht der Egofahrer dem Objekt (grün) sehr spät (ungefähr $TTC = 1s$) rechts aus. Die Kursprädiktion ist entsprechend der geschätzten Wahrscheinlichkeit mit einem intensiveren Rot hinterlegt. Zu Beginn der Szene wird noch kein Ausweichen erkannt, mit Fortschritt der Zeit häufen sich jedoch die Indikatoren (z.B. Drehen am Lenkrad, violette Kreise) und die Kursprädiktion tendiert in die korrekte Richtung.

was vor allem mit dem systematischen Diskretisierungsfehler zusammenhängt: Bei hohen Geschwindigkeiten nimmt die Kurskrümmung κ sehr kleine Werte an, die durch die Diskretisierung meistens den Wert 0 erhalten und somit die Kursprädiktion zu einer Geradeausfahrt verfälschen. Durch Normalisierung der Kurskrümmung ($a, a_{y,Rel}$) entstehen bereits deutlich bessere Resultate, welche durch die Differenzkodierung nur noch geringfügig verbessert werden konnten.

Für die beste Strategie - die Normalisierung der Kurskrümmung ($a, a_{y,Rel}$) - wurde der Diskretisierungsfehler ermittelt; er ist für etwa 50% des lateralen Fehlers verantwortlich. Trotz logarithmischer Diskretisierung besteht hier also weiteres Verbesserungspotenzial.

Fazit

Durch den Random Forest Algorithmus wurde eine deutliche Verbesserung gegenüber zirkulärer Kursprädiktion erreicht; dies ist darauf zurückzuführen, dass eine genauere Schätzung der Kursmodellparameter erfolgt, welche durch Berücksichtigung zahlreicher Indicatorsignale der Situation angepasst ist. Die geschickte Transformation und Diskretisierung der Parameter macht einen wesentlichen Teil des Erfolgs aus.

Eine besondere Stärke der Kursprädiktion mit Maschinellern Lernen ist, dass der Kurs nun probabilistisch statt deterministisch modelliert wird (Abb. 8.17). Darauf aufbauende Methoden können die Wahrscheinlichkeit als Konfidenzmaß verwenden, um zum Beispiel Kollisionswarnungen nur in eindeutig interpretierbaren Szenen zu aktivieren.

8.11 Gesamtvergleich der Methoden

Abschließend werden die drei untersuchten Methoden in Hinblick auf Fahrerintentionserkennung und Kollisionsdetektion detailliert miteinander verglichen. Es wurde jeweils die beste Parametrierung verwendet:

1. Random Forest ohne Historie auf allen verfügbaren Merkmalen
2. Kursprädiktion mit Differenzkodierung und Normierung der Kurskrümmung, anhand der Parameter $(\Delta a, \Delta a_{y,Rel})$
3. Klassifikations-HMMs mit $Q=15$ Zuständen, ein Random Forest Modell je Zustand (Random Forest X)

Die Evaluierungsmetriken sind detailliert in Kapitel 4.5 beschrieben. Die Auswertung wurde durchgängig auf kritische Zeitpunkte beschränkt, um die starke Überzahl der Neutral-Klasse zu kompensieren (Abb. 8.2).

8.11.1 Fahrerintentionserkennung

Tabelle 8.2 zeigt die Klassifikationsgüte der Deeskalationsmanöver. Random Forest (RF) und HMMs erzielen eine fast identische Fehlerrate von etwa 13%. Die Deeskalationsmanöver, von der Kursprädiktion abgeleitet, weichen zu 32.5% von der Ground Truth ab, und erzielen damit einen mehr als doppelt so hohen Fehler.

Strategie	Korrektheit (%)	Fehler (%)
1: Random Forest	86.7	13.3
2: Kursschätzung	67.5	32.5
3: HMM + RF	86.8	13.2

Tabelle 8.2: Gesamtkorrektheit der geschätzten Deeskalationsmanöver über alle kritischen Zeitpunkte in den Daten. Verfahren 1 und 3 wurden von vornherein nur auf diesen Klassen trainiert und erzielen eine höhere Güte als die Kursschätzung, die generisch und kontinuierlich arbeitet.

Performanz je Klasse

Eine Aufschlüsselung der Klassifikationsrate nach Manövern differenziert die Resultate (Abb. 8.18): RF und HMMs erzielen auf fast allen Klassen eine hohe Detektionsrate (*Sensitivity*) sowie Robustheit (*Precision*) von $> 67\%$. Lediglich Kollisionen werden mit 46% Erkennungsrate schlechter verarbeitet.

Die Kursprädiktion erkennt die Kollisionsklasse mit einer höheren Sensitivität von 85%, jedoch nur 30% Robustheit. Dies bedeutet, dass in den Testdaten unter 10 Zeitpunkten, zu denen eine Kollision geschätzt wurde, 7 Fehlalarme waren. Gleichzeitig ist sie für Bremsvorgänge mit einer Erkennungsrate von 6% scheinbar ungeeignet - es könnte daran liegen, dass die Längs- und Querschätzung unabhängig voneinander geschätzt werden und besonders die Längsschätzung noch Verbesserungspotenzial bietet.

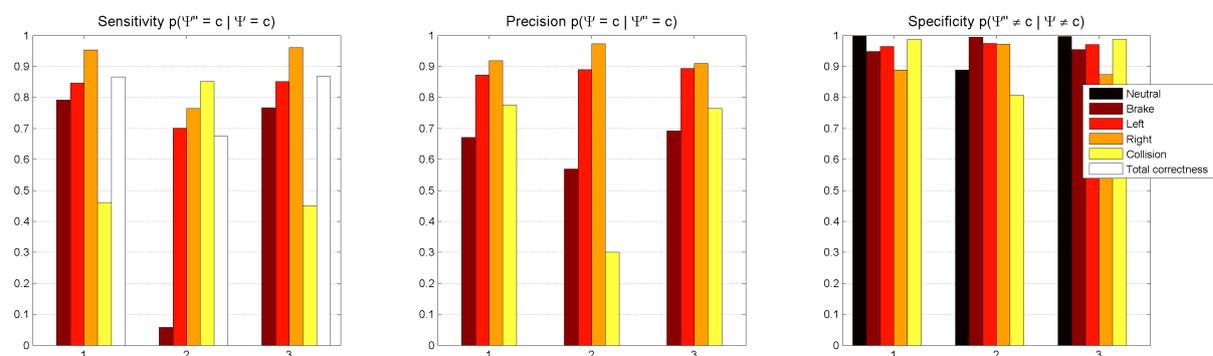


Abbildung 8.18: Gesamtfehler der Fahrerintentionserkennung, bedingt über die Klasse. Die Kursprädiktion (2) erzielt die geringste Korrektheit, besonders die Klasse *Brake* wird kaum erkannt. Kollisionen werden dafür zuverlässig gefunden, jedoch mit geringer Präzision und vielen Fehlalarmen.

Performanz bezüglich Kritikalität

Um situative Kennzahlen zu erhalten, wird die erweiterte bedingte Sensitivität (Kapitel 4.5.3) verwendet: Die Verteilung der Fehler wird bezüglich des Kritikalitätsmaßes TTC dargestellt, um zu ermitteln, zu welchen Zeitpunkten während einer kritischen Szene sich die Klassifikation verbessert oder verschlechtert. Weil niedrige TTCs extrem selten sind, muss berücksichtigt werden, dass die Statistik in diesem Bereich verfälscht sein kann.

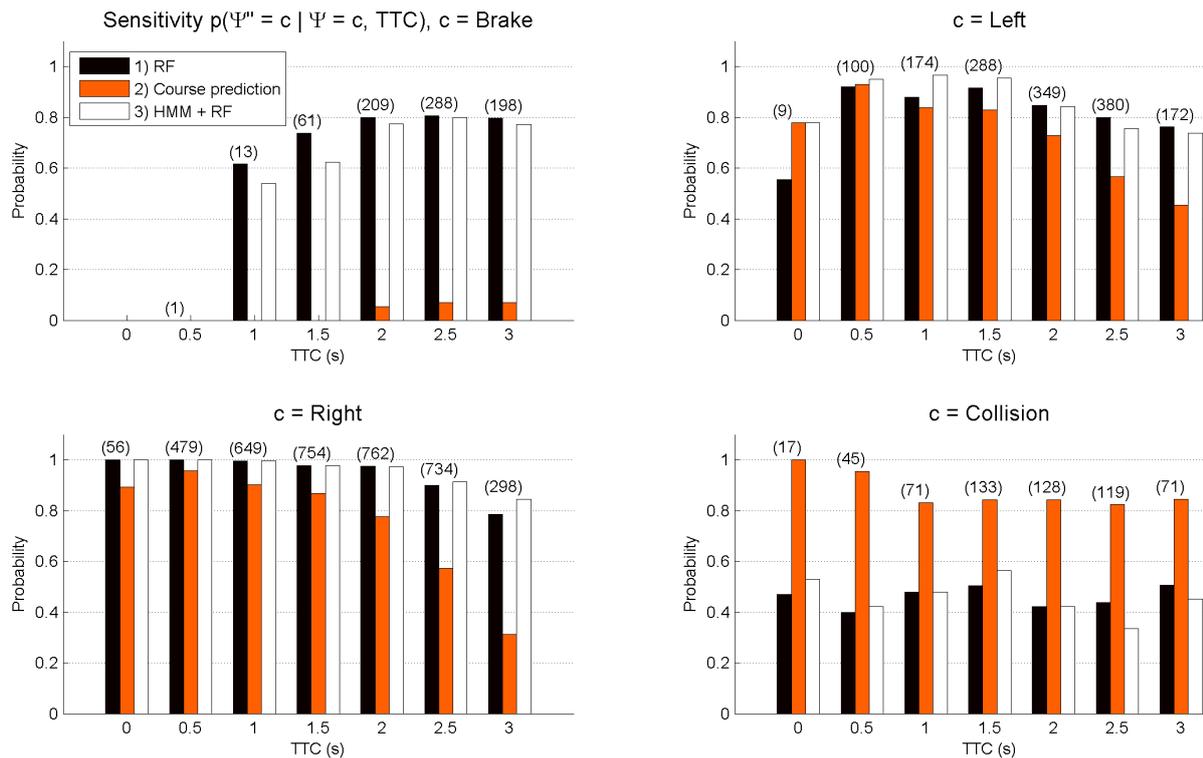


Abbildung 8.19: Situative Klassifikationsrate, bedingt über die wahre Klasse und die geschätzte Kollisionszeit TTC. Mit niedrigerer TTC werden die Szenen zunehmend kritischer. Die Zahlen kennzeichnen die Anzahl der Zeitpunkte, die in diese Kategorie mit eingingen.

Abb. 8.19 zeigt, dass mit sinkender TTC die Klassifikationsrate für Ausweichmanöver steigt. Intuitiv ist dies verständlich, weil mit der Annäherung die Handlungsmöglichkeiten des Fahrers sinken; auch die Voruntersuchung des Bayes-Fehlers ließ diese Verbesserung erwarten. Für die Anhalte- und Kollisionsklasse ergeben sich jedoch keine Veränderungen, was durch die stark verminderte Zahl ausgewerteter Szenen und damit niedrigere statistische Aussagekraft erklärt werden kann.

Das Defizit der Kursprädiktion gegenüber RF und HMMs steigt bezüglich Ausweichvorgängen mit der TTC an - zu frühen Zeitpunkten erkennen die diskreten Schätzer eine Ausweichabsicht des Fahrers also besser. Ein Grund dafür ist, dass mit dem verwendeten Kursmodell nur kreisförmige Kurse mit konstantem Radius abgebildet werden können; damit kann ein zukünftiges Ausweichmanöver mit diesem Modell nur schwer abgebildet werden (Kapitel 4.4).

Während HMMs mit Random Forest (3) gegenüber RF allein (1) in der Gesamtauswertung kaum Unterschiede aufweisen, stellt sich in der bedingten Auswertung eine leichte Tendenz heraus: HMMs können Ausweichvorgänge tendenziell besser erkennen, während RF mehr Bremsvorgänge korrekt findet.

8.11.2 Korrektheit der Warnfunktion

Mit Bezug zur Anwendung wurde die Schätzung der Deeskalationsklassen nun auf die Kollisionserkennung beschränkt und zusammen mit der Warnfunktion ausgewertet. Während einer eskalierenden Situation ist nur das erste Auftauchen einer Warnung relevant, weil daraufhin der Fahrer benachrichtigt wird. Die Anzahl der Zeitpunkte mit korrekter Warnung ist also nicht von Belang, sondern die Anzahl korrekt behandelter Szenen.

Die Ergebnisse dieser anwendungsspezifischen Untersuchung sind konform zu den vorherigen generischen Ergebnissen: Von insgesamt 54 Kollisionsszenen werden mit der Kursprädiktion zwar alle gefunden, jedoch erscheinen ebenso viele Fehlalarmszenen (Abb. 8.20). RF und HMMs finden 41 bzw. 43 Kollisionsszenen und weisen damit immer noch ein hohes Nutzenpotenzial für die Schadensvermeidung auf.

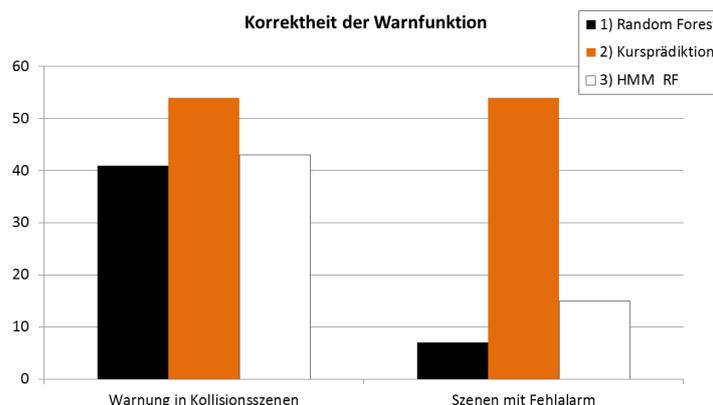


Abbildung 8.20: Auswertung der Warnfunktion szenenweise. Insgesamt wurden 163 kritische Szenen berücksichtigt, darunter 54 Kollisionsszenen. Die Kursprädiktion wird zwar in allen Kollisionsszenen aktiv, erzielt jedoch genauso viele Fehlalarme - bei Random Forest und HMMs ist das Verhältnis deutlich besser.

Bei Betrachtung der Fehlalarmszenen fällt nun erstmal ein Unterschied zwischen den sonst stets ähnlichen Resultaten auf: Mit 8 gegenüber 15 Fehlalarmszenen ist RF am robustesten gegen Fehlalarme; diesem großen Vorteil gegenüber HMMs stehen nur zwei zusätzlich verpasste Kollisionsszenen gegenüber. Eine manuelle Auswertung ergab zudem, dass etwa die Hälfte der Fehlalarme in subjektiv kritischen Situationen auftraten mit späten Lenkvorläufen oder Übertreten der Komfortgrenzen (Kapitel 3.2); damit sind sie zu einem bestimmten Grad tolerierbar und relativieren die Fehlalarme etwas.

Die Fehlalarmrate ist ein wichtiger Faktor für die Behandlung des Warndilemmas und das Akzeptanzverhalten des Fahrers (Kapitel 3.4). Letztendlich muss anhand des Gesamtsystems entschieden werden, welche Strategie das beste Kosten-Nutzen-Verhältnis liefert.

Diese Wahl ist unter anderem von der Zielgruppe der Autofahrer und deren Fahrstil abhängig.

8.11.3 Nutzen des Warnsystems zur Unfallvermeidung

Abschließend wurde ermittelt, welchen Nutzen die betrachteten Warnstrategien im realen Unfallgeschehen haben können. Hierzu wurde simuliert, dass der Fahrer in Kollisionsszenen nach der Warnung eine Notbremsung durchführt (Kapitel 3.2); die Anzahl vermiedener bzw. gelinderter Unfälle gibt damit Auskunft über den finalen Wirkungsgrad des Gesamtsystems.

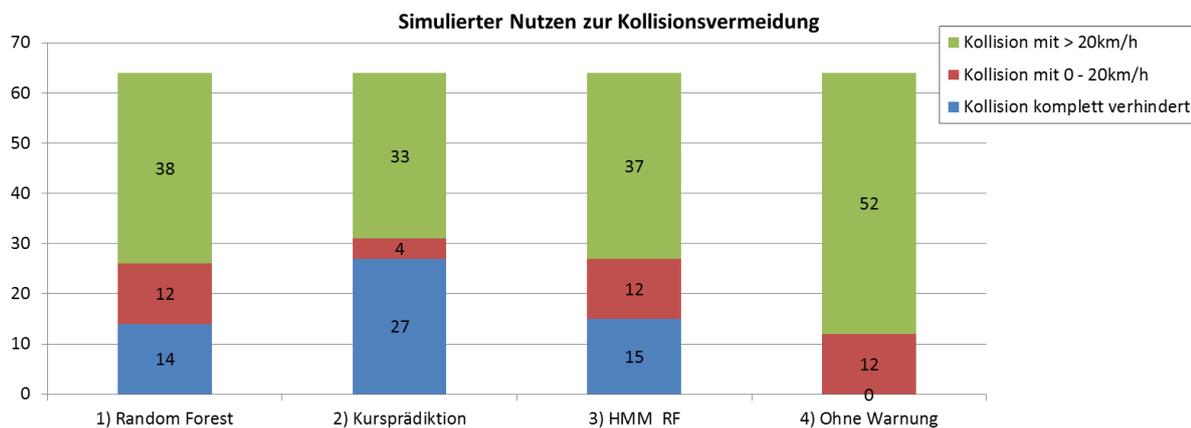


Abbildung 8.21: Simulierter Nutzen der verschiedenen Warnstrategien zur Kollisionsvermeidung. Für diese Auswertung wurden insgesamt 64 Kollisionsszenen betrachtet. Die verbleibenden Kollisionsszenen wurden abhängig von der Relativgeschwindigkeit in zwei Kategorien eingeteilt, für leichten und schweren erwarteten Schaden.

Insgesamt lagen 64 Kollisionsszenen vor, in 12 Szenen betrug die Kollisionsgeschwindigkeit weniger als 20 km/h, was als unkritisch angenommen werden kann. In einigen Szenen wurde das Objekt erst nach dem Warnzeitpunkt detektiert, so dass eine vollständige Vermeidung der Kollision unmöglich ist. Für den Vergleich der Verfahren hat dieser Umstand aber kaum eine Bedeutung.

Im Resultat entsteht durch die Kursprädiktion der meiste Nutzen: 27 Kollisionen werden vollständig vermieden, und es verbleiben nur noch 33 statt 52 schlimmen Unfällen. Random Forest und HMMs verhindern zwar nur 14/15 Kollisionen vollständig, reduzieren die schlimmen Kollisionen aber ebenfalls effektiv auf 38/37 (Abb. 8.21). Während die Kursprädiktion also insgesamt fast doppelt so viel Unfälle vermeidet wie Random Forest und HMMs, reduziert sich der Vorteil in schweren Unfallszenen auf etwa 13% mehr Nutzen.

8.11.4 Ergebnisse im Vergleich

Für die Erkennung der Deeskalationsklassen konnte mit RF und HMMs ein Gesamtfehler von 13% erzielt werden, was die grundsätzliche Anwendung dieser Verfahren unter Beweis stellt. Die Kursprädiktion erzielt mit 32% einen deutlich höheren Fehler - er wurde auf das zugrundeliegende Kursmodell zurückgeführt, welches noch nicht alle Deeskalationsmanöver abbilden kann.

Durch die anwendungsnahe Evaluierung der Warnfunktion erhalten die Unterschiede eine greifbare Bedeutung: Auf insgesamt 163 kritischen Szenen entsteht mit der Kursprädiktion eine korrekte Warnung in allen 54 Kollisionsszenen, jedoch mit genauso vielen Fehlalarmen. RF und HMMs warnen etwas seltener, erzielen dafür aber mit nur 8 bzw. 15 Fehlalarmszenen ein besseres Verhältnis zwischen Nutzen und Übernutzen.

Beachtlich ist der Vorteil von RF gegenüber HMMs; im Vergleich zur Erkennung der Deeskalationsmanöver wurde kein solcher Unterschied gefunden (Abschnitt 8.11.1). Die Fehlerkennungen der HMMs sind also auf mehrere Szenen verteilt, während RF eher einzelne Szenen komplett richtig oder falsch klassifiziert. Daraus lässt sich ableiten, dass das Training der HMM-Struktur noch nicht optimal ist und einzelne Ausreißer zulässt.

In der abschließenden Nutzenbetrachtung wurden mit der Kursprädiktion 27 von 64 Kollisionsszenen vollständig vermieden. RF und HMMs vermeiden zwar nur 14 bzw. 15 Kollisionen vollständig, werden aber grundsätzlich auch aktiv, da schlimme Unfälle nur zu 13% weniger reduziert wurden als mit der Kursprädiktion. Mit diesen Strategien wird die Kollision also grundsätzlich erkannt, jedoch zu entsprechend späteren Zeitpunkten als bei der Kursprädiktion.

Zusammenfassend erzielt RF den besten Kompromiss zwischen Nutzen und Übernutzen, gefolgt von den HMMs mit der doppelten Anzahl an Fehlalarmszenen. Die Kursprädiktion weist sehr viele Fehlalarmszenen auf, die den leicht erhöhten Nutzen deutlich abmindern.

8.12 Zusammenfassung

Anhand des Bayes-Fehlers konnte die Bedeutung historischer Merkmale für die Schätzung der Fahrerintention bewiesen werden. Um dieses Ergebnis in der praktischen Anwendung zu verifizieren, wurde der Random Forest Algorithmus mit historischen Daten untersucht. Die Resultate waren gemischt: Die Verwendung historischer Informationen mit dem *Moving Window* Prinzip ergab auf Highlevel-Manövern eine Verbesserung, auf Deeskalationsmanövern jedoch eine geringe Verschlechterung der Klassifikation. Sie wurde auf die variierende Ausführungs geschwindigkeit der Manöver zurückgeführt; eine entsprechende Anpassung der Abtastpunkte blieb jedoch ohne Erfolg. Eine mögliche Interpretation ist, dass die Ergänzung der Daten mit historischen Werten zuviel Redundanz verursacht. Letzt-

endlich erzielt Random Forest zwar gute Ergebnisse, ist aber nicht für sequentielle Daten konzipiert und stößt damit an seine Grenzen.

Um eine Verbesserung der Klassifikation zu erzielen und gleichzeitig die sequentielle Datenstruktur auszunutzen, wurden *Hidden Markov Modelle* angewandt und methodisch weiterentwickelt. Mit randomisierten Ensemble-HMMs wurde die Abtastrate der Größenordnung $\omega = 2\text{Hz}$ bestätigt. Gleichzeitig konnten Verbesserungen nur durch mehr Gaussian Mixtures erreicht werden, nicht jedoch durch mehr Zustände - im Extremfall eines einzelnen Zustands entstanden sogar die besten Resultate. Es wurde gefolgert, dass die Modellierung der Beobachtungswahrscheinlichkeiten, der HMM-Struktur sowie der zeitlichen Fortschritte mit sorgfältigem Expertenwissen durchgeführt werden muss; die Resultate des Baum Welch Trainings allein sind für diesen Anwendungsfall nicht benutzbar.

Die vorgestellten Klassifikations-HMMs mit Random Forest Beobachtungen ergaben deutlich verbesserte Ergebnisse, die durch eine Anpassung der HMM-Struktur weiter ausgebaut werden konnten. Eine detaillierte Analyse der Zustände zeigte, dass durch Random Forest komplexere Beobachtungswahrscheinlichkeiten trainiert wurden als durch Gaussian Mixtures, was die guten Ergebnisse erklärt. Der Vergleich mehrerer Zustände zeigte jedoch, dass häufig dieselben Beobachtungswahrscheinlichkeiten gelernt wurden. Gleichzeitig lag die erwartete Verweildauer in einem Zustand der trainierten Modelle bei $D = 1.2$, das HMM wechselt also ständig den Zustand. Während des Baum Welch Trainings fand somit keine Spezialisierung einzelner Zustände statt; das kompliziertere Training der Random Forest Beobachtungen ist also weiterhin ausbaufähig.

Mit der Kursprädiktion wurde in dieser Arbeit ein generischer Ansatz vorgestellt: Auch dynamische Situationen können prädiziert werden, ohne auf einen einzelnen Anwendungsfall eingeschränkt zu sein. Dank geschickter Transformation der Parameter konnte hier ein mittlerer Schätzfehler von 11cm Kursabweichung auf $\Delta t = 2s$ Prädiktionszeit erzielt werden - gegenüber dem Standardverfahren der zirkulären Prädiktion ist dies eine Verbesserung um fast 40%.

Im finalen, anwendungsbezogenen Vergleich konnte der Random Forest Algorithmus den besten Kompromiss zwischen Vermeidung von Kollisionen und geringer Fehlalarmrate erzielen. HMMs konnten die Fahrerintention global ausgewertet zwar gleich gut schätzen, warnten aber in 15 gegenüber 8 Szenen unberechtigt. Die Kursprädiktion erzielte bereits im Vergleich der Deeskalationsmanöver mehr als die doppelte Fehlerrate, was genauso in der Anzahl von Fehlalarmszenen wiederkehrte.

9 Zusammenfassung und Fazit

9.1 Zusammenfassung

In dieser Arbeit wurde eine Brücke zwischen Fahrerassistenzsystemen und der Anwendung grundsätzlicher Prinzipien des Maschinellen Lernens erstellt. Anhand des Warndilemmas in uneindeutigen Situationen wurde abgeleitet, dass zur Kollisionsvermeidung eine frühe Kenntnis der Fahrerintention von essentieller Wichtigkeit ist. Für deren Erkennung wurden verschiedene Verfahren untersucht und methodisch weiterentwickelt.

Anwendung in der Fahrerassistenz

Die Fahrerintention wurde durch diskrete Deeskalationsklassen modelliert, welche den Relationswechsel zwischen dem Egofahrzeug und Bezugsobjekt beschreiben. Während diskrete Klassen durch Maschinelles Lernen leicht prädiziert werden können, ist die Verwendung einer solchen Schätzung jedoch stark auf Kollisionswarnsysteme eingeschränkt; zudem wurde vermutet, dass eine getrennte Schätzung der Ego- und Objektzukunft Abhängigkeiten minimieren und damit das Ergebnis verbessern könnte.

Mit dieser Hypothese wurde die Prädiktion des zukünftigen Fahrzeugkurses untersucht, welche universeller anwendbar ist und zudem in Deeskalationsmanöver überführt werden kann. Durch ein geschickt gewähltes Prädiktionsmodell ergab sich eine grundsätzlich gute Modellierbarkeit des Kurses, und mit Maschinellern Lernen für die Parameterfindung ein 30% geringerer Kursfehler gegenüber kreisförmiger Kursprädiktion. Bei der Überführung in Deeskalationsmanöver entstanden jedoch zu viele fehlerhaft gefundene Kollisionen, und das auf der Kursprädiktion basierende Warnsystem verursachte viele Fehlalarme.

Die Untersuchungen der Fahrerintentionserkennung ließen trotz verschiedener Algorithmen und Modellierungen konforme Schlussfolgerungen über die Daten zu. Die grundlegende Definition der Evaluierungsmetriken in dieser Arbeit verhalten zu dieser Vergleichbarkeit; die Metriken sind skalierbar zwischen generisch und anwendungsspezifisch und können damit auch in verwandten Domänen der Mustererkennung oder Fahrerassistenz verwendet werden.

Verschiedene Experimente ergaben eine Abtastrate der Größenordnung $\omega = 2\text{Hz}$, um Informationen über das menschliche Steuerverhalten abzubilden. Gleichzeitig konnten durch

Maschinelles Lernen die wichtigsten Merkmale für den Anwendungsfall identifiziert werden. Ein solches Ranking ist in der Anwendung von besonders hohem Nutzen, es erlaubt z.B. die explizite Modellierung heuristischer Klassifikatoren.

Methodische Erweiterungen

Im methodischen Fokus stand die Erweiterung von Hidden Markov Modellen für komplexe Sensorsignale; sie wurden aufgrund vorheriger Untersuchungen mit dem Random Forest Algorithmus kombiniert, welcher für die Berechnung der Beobachtungswahrscheinlichkeiten eingesetzt wurde.

Weil Random Forest konstruktionsbedingt ein diskriminatives Verfahren ist, HMMs aber generativ arbeiten, wurde das Baum Welch Training entsprechend ausgebaut. Der ursprüngliche Klassifikationsfehler von 29% mit Gaussian Mixture Beobachtungen konnte damit auf beachtliche 18.5% gesenkt werden.

Auch die Struktur der HMMs wurde erweitert: Die bekannten Ensemble-HMMs wurden zu einem Klassifikations-HMM kombiniert, was eine vereinfachte Anwendung für die Online-Klassifikation langer Datensequenzen ermöglicht. Es konnte bewiesen werden, dass die neue Struktur mindestens gleichmächtig zu den Ensemble-HMMs ist; darüberhinaus bietet sie aber eine flexible Modellierbarkeit der Klassenübergänge, wodurch eine weitere Verbesserung auf 16% erzielt wurde (13% in der finalen Auswertung mit allen verfügbaren Merkmalen).

Grenzen der Untersuchungen

Letztendlich konnten durch die HMM Erweiterungen mit Random Forest zwar deutlich bessere Ergebnisse als mit Gaussian Mixture gesammelt werden, jedoch ergaben sich in der Praxis keine Vorteile gegenüber Random Forest allein.

Eine Interpretation dazu ist, dass HMMs keine globale zeitliche Streckung verarbeiten können: Manöver werden in unterschiedlicher Geschwindigkeit und Intensität gefahren (Kapitel 2.3) - diese Einflüsse sind aber über einen Steuervorgang ansatzweise konstant, womit kein Markovprozess erster Ordnung mehr vorliegt. Hierfür müssten komplexere sequentielle Modelle untersucht werden. Lineare dynamische Systeme wären eine naheliegende Lösung, die zur Klassifikation kontinuierliche und diskrete Zustände benötigen. Die Komplexität des Modelltrainings steigt dadurch jedoch stark an [Mur02].

Die vorgestellten HMMs mit Random Forest Beobachtungen werden noch nicht optimal trainiert, was eine detaillierte Untersuchung der Zustände zeigt: Durch den diskriminativen Charakter von Random Forest entstand während des Trainings häufig der Fall, dass zu einem Zeitpunkt mehrere Zustände dieselben Aktivitäten besitzen. Dadurch konnte keine Spezialisierung der Zustände stattfinden, und das HMM konnte nicht die erwarteten

Vorteile aus der zeitlichen Struktur ziehen. Um während des Trainings also die Sequenzen spezifisch auf bestimmte Zustände abzubilden, müssten noch passende Erweiterungen des Baum Welch Algorithmus erforscht werden.

9.2 Fazit

Für die Klassifikation sequentieller Daten wurde durch Klassifikations-HMMs mit Random Forest Beobachtungen ein wichtiger Beitrag geleistet: Er ermöglicht eine gute Modellierbarkeit von Klassenwechseln, Online-Verarbeitung langer Sequenzen und stabile Analyse der Indikatorsignale. Das Training muss jedoch weiter optimiert werden, um zeitliche Informationen in den Daten besser zu berücksichtigen.

Die probabilistische Modellierung des Frameworks als Bayessches Netzwerk zeigt, wie stochastische Unabhängigkeiten ausgenutzt werden können. Die relevanten Vorgänge im Verkehrsgeschehen werden dabei jeweils für sich geschätzt und sind nicht mehr abhängig von mehreren Objekten; hierdurch ergeben sich Vorteile für die Schätzung und eine transparente Evaluierung. Die probabilistischen Schätzer berechnen auch Konfidenzen; Fahrerassistenzsysteme können abhängig von diesen intensiver oder zurückhaltender einzugreifen und so die Akzeptanz des Benutzers zu verbessern.

Bei der Kursprädiktion konnten durch das parametrische Kursmodell und Maschinelles Lernen vielversprechende Fortschritte gegenüber Standardverfahren erzielt werden. In dynamischen Situationen zeigte sich ein großes Ausbaupotenzial, weil durch die Kursapproximation mit einem zirkulären Modell zu hohe Abweichungen entstanden. An dieser Stelle soll ein komplexeres Kursmodell motiviert werden, welches entsprechende Vorgänge präziser abbilden und vorhersagen kann.

9.3 Ausblick

Zukünftige Fahrerassistenzsysteme sollten die Situationsanalyse von der Objekterkennung bis hin zur Fahrerintentionserkennung möglichst universell durchführen (Abb. 1.3). Eine ideale Modellierung deckt die Zusammenhänge zwischen allen relevanten Größen ab, wie zum Beispiel sich beeinflussende Verkehrsteilnehmer. Von größerer Bedeutung ist aber die Ausnutzung der Unabhängigkeiten, denn ein einzelnes Modell muss mit wachsender Anzahl an Verkehrsteilnehmern und Objekten in der Szene früher oder später an seine Grenzen stoßen. Statt eines einzelnen HMMs oder Random Forest Modells, welches auf allen Merkmalen parallel arbeitet, sollten also mit geschickter Dekomposition kleinere Modelle erstellt werden, die jeweils nur die tatsächlich relevanten Informationen austauschen. Mit der Kursprädiktion wurde bereits ein Schritt in diese Richtung unternommen.

Für die Modellierung der einzelnen Komponenten sollte der Fokus vor allem auf die Abbildung der zeitlichen Reihenfolge gelegt werden. Diese repräsentiert nicht nur Abläufe einzelner Manöver, sondern auch kausale Zusammenhänge und menschliche Schlussfolgerungen, wie die Reaktion auf vorausfahrenden Verkehr oder eine komplizierte Kreuzungssituation mit weiteren Verkehrsteilnehmern. Für den zeitlichen Rückblick wurde mit HMMs ein entsprechender Beitrag geleistet; vorausschauend werden teils spieltheoretische Ansätze wie *Backtracking* für Prädiktionen verwendet. Nur ein gemeinsames, integriertes Framework kann die Komplexität und zahlreichen Möglichkeiten des Verkehrsgeschehens beherrschen. Bei korrekter Modellierung und einem hinreichenden Abstraktionsgrad kann es gleichfalls als Fahrerunterstützung und für autonomes Fahren eingesetzt werden - eine große Gemeinsamkeit zweier bisher unterschiedlicher Domänen, die zukünftig eine große Rolle spielen kann.

Auf funktionaler Seite ist der genaue Modus der Fahrerunterstützung und damit die Interaktion mit der menschlichen Komponente noch stark ausbaufähig. Unsicherheiten werden in derzeitigen Systemen kaum berücksichtigt; je nach Resultat der Schätzung geschieht ein Eingriff oder nicht. Studien zeigten großes Potenzial und hohe Akzeptanz für graduelle Eingriffe. Für zukünftige Systeme wird deshalb empfohlen, die Eingriffsstärke entsprechend der Sicherheit der Situationsanalyse zu regulieren, was mit einem probabilistischen Ansatz wie dem vorgestellten Framework leicht möglich ist.

Literaturverzeichnis

- [ASB09] ALTHOFF, M. ; STURSBURG, O. ; BUSS, M.: Model-Based Probabilistic Collision Detection in Autonomous Driving. In: *IEEE Transactions on Intelligent Transportation Systems*, 2009, S. 299–310
- [ASP10] ABDEL HADY, M. F. ; SCHWENKER, F. ; PALM, G.: Semi-supervised learning for tree-structured ensembles of RBF networks with Co-Training. In: *Neural Networks* 23 (2010), Nr. 4, S. 497–509
- [Bat94] BATTITI, R.: Using mutual information for selecting features in supervised neural net learning. In: *IEEE Transactions on Neural Networks* 5 (1994), S. 537–550
- [BB96] BENGIO, S. ; BENGIO, Y.: An EM algorithm for asynchronous input/output hidden Markov models. In: *International Conference On Neural Information Processing* 78 (1996), S. 328–334
- [BED08] BERNDT, H. ; EMMERT, J. ; DIETMAYER, K.: Continuous Driver Intention Recognition with Hidden Markov Models. In: *11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China*, 2008, S. 1189–1194
- [BFHP10] BÖRGER, J. ; FASSBENDER, B. ; HÄRING, J. ; PALM, G.: A new Approach of Situation Prediction for Manifold Driver Assistance Systems. In: *26. VDI/VW-Gemeinschaftstagung Fahrerassistenz und integrierte Sicherheit*, 2010
- [BFOS84] BREIMAN, Leo ; FRIEDMAN, Jerome H. ; OLSHEN, Richard A. ; STONE, Charles J.: *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984
- [BHP10] BÖRGER, J. ; HÄRING, J. ; PALM, G.: Improved Vehicle Course Prediction by Machine Learning for Collision Warning Systems. In: *19th Aachen Colloquium „Automobile and Engine Technology“*, 2010
- [Bil06] BILMES, J. A.: What HMMs Can Do. In: *IEICE - Transactions on Information and Systems* E89-D (2006), March, Nr. 3, S. 869–891. – ISSN 0916–8532

- [Bis06] BISHOP, C. M.: *Pattern recognition and machine learning*. Bd. 4. Springer New York, 2006
- [BL08] BLASCHKO, M. B. ; LAMPERT, C. H.: Learning to Localize Objects with Structured Output Regression. In: *10th European Conference on Computer Vision*, 2008, S. 2–15
- [BOP97] BRAND, M. ; OLIVER, N. ; PENTLAND, A.: Coupled hidden Markov models for complex action recognition. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997)*
- [Bre96] BREIMAN, L.: Out-of-bag Estimation. In: *Technical Report, Statistics Department, University of California* (1996)
- [Bre01] BREIMAN, L.: Random forests. In: *Machine Learning* 45 (2001), S. 5–32
- [BS06] BREFFELD, U. ; SCHEFFER, T.: Semi-Supervised Learning for Structured Output Variables. In: *23. International Conference on Machine Learning*, 2006
- [BSF08] BLASCHKE, C. ; SCHMITT, J. ; FÄRBER, B.: Überholmanöver-Prädiktion über CAN-Bus-Daten. In: *ATZ* 11 (2008)
- [BSH09] BOYRAZ, P. ; SATHYANARAYANA, A. ; HANSEN, J.H.L.: Driver Behavior Modeling Using Hybrid Dynamic Systems For 'Driver-Aware' Active Vehicle Safety. In: *Enhanced Safety of Vehicles* (2009)
- [BTK05] BULD, S. ; TIETZE, H. ; KRÜGER, H.-P.: Auswirkungen von Teilautomation auf das Fahren. In: *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. (2005)
- [BWB09] BATZ, T. ; WATSON, K. ; BEYERER, J.: Recognition of dangerous situations within a cooperative group of vehicles. In: *IEEE Intelligent Vehicles Symposium*, 2009, S. 907–912
- [Cav07] CAVENEY, D.: Stochastic Path Prediction using the Unscented Transform with Numerical Integration. In: *IEEE Intelligent Transportation Systems*, 2007
- [CKK⁺07] CHOI, S.J. ; KIM, J.H. ; KWAK, D.G. ; ANGKITITRAKUL, P. ; HANSEN, J.H.L.: Analysis and Classification of Driver Behavior using In-Vehicle CAN-Bus Information. In: *Biennial Workshop on DSP for In-Vehicle and Mobile Systems, Istanbul, Turkey* (2007)
- [CKY08] CARUANA, R. ; KARAMPATZIAKIS, N. ; YESSENALINA, A.: An empirical evaluation of supervised learning in high dimensions. In: *25th international*

conference on Machine learning, 2008, S. 96 – 103

- [CLB04] CHEN, C. ; LIAW, A. ; BREIMAN, L.: Using Random Forest to Learn Imbalanced Data. In: *Discovery* (2004), Nr. 1999, S. 1–12
- [CPD03] CONSOLE, L. ; PICARDI, C. ; DUPRE, D.T.: Temporal Decision Trees: Model-based Diagnosis of Dynamic Systems On-Board. In: *Journal of Artificial Intelligence Research* 19 (2003), S. 469–512
- [DDM⁺09] DICKMANN, Juergen ; DIEWALD, Fabian ; MÄHLISCH, Mirko ; KLAPPSTEIN, Jens ; ZUTHER, Sebastian ; PIETZSCH, Silvia ; HAHN, Stefan ; MUNZ, Michael: Environmental Perception for Future Integrated Safety Systems. In: *International Technical Conference on the Enhanced Safety of Vehicles*, 2009
- [DFF07] DEML, B. ; FREYER, J. ; FÄRBER, B.: Ein Beitrag zur Prädiktion des Fahrstils. In: *Fahrer im 21. Jahrhundert, Human Machine Interface, VDI-Bericht* 2015 (2007), S. 47–59
- [DGW07] DU, Y. ; GOLDAU, A. ; WOHLBERG, S.: Algorithmen zur Fahrspurprädiktion für Fahrerassistenzsysteme am Beispiel Adaptive Cruise Control. In: *ATZ* 12 (2007)
- [DHS01] DUDA, R. O. ; HART, P. E. ; STORK, D. G.: *Pattern Classification*. Wiley & Sons, 2001
- [Die02] DIETTERICH, T. G.: Machine learning for sequential data: A review. In: *Structural, Syntactic and Statistical Pattern Recognition*, Springer-Verlag, 2002, S. 15–30
- [Dit09] DITTMANN, T.: *Lernalgorithmen zur Vorhersage und Klassifikation von Ausweichmanövern*, Fakultät für Ingenieurwissenschaften und Informatik der Universität Ulm, Diplomarbeit, 2009
- [DV09] DEMIRDJIAN, D. ; VARRI, C.: Recognizing events with temporal random forests. In: *International conference on Multimodal interfaces*, 2009, S. 293 – 296
- [EF71] EKMAN, P. ; FRIESEN, W. V.: Constants across cultures in the face and emotion. In: *Journal of Personality and Social Psychology* 17 (1971), S. 124–129
- [EFT⁺10] ERSAL, Tulga ; FULLER, Helen J. A. ; TSIMHONI, Omer ; STEIN, Jeffrey L. ; FATHY, Hosarn K.: Model-based analysis and classification of driver distraction under secondary tasks. In: *IEEE Transactions on Intelligent Transportation Systems* (2010), September, S. 692–701

- [Enc12] ENCYCLOPEDIA OPENTOPIA - FREE ONLINE ENCYCLOPEDIA: (2012). http://encycl.opentopia.com/term/Decision_tree
- [Eur01] EUROPEAN COMMISSION: White Paper, European Transportation Policy for 2010: Time to Decide. (2001)
- [Eur10] EUROPEAN COMMISSION: Towards a European Road Safety Area: policy orientations on road safety 2011-2020. (2010)
- [FAB08] FUCHS, K. ; ABENDROTH, B. ; BRUDER, R.: Aktive Gefahrenbremsungen - Wie reagiert der Fahrer? In: *24. VDI VW-Gemeinschaftstagung - Integrierte Sicherheit und Fahrerassistenzsysteme*, VDI Verlag, 2008
- [FM98] FRAILE, R. ; MAYBANK, S. J.: Vehicle Trajectory Approximation and Classification. In: *British Machine Vision Conference*, 1998
- [Grü05] GRÜNDL, M.: *Fehler und Fehlverhalten als Ursache von Verkehrsunfällen und Konsequenzen für das Unfallvermeidungspotenzial und die Gestaltung von Fahrerassistenzsystemen*, Universität Regensburg, Diss., September 2005
- [GSS11] GLODEK, M. ; SCHERER, Stefan S. ; SCHWENKER, F.: Conditioned Hidden Markov Model Fusion for Multimodal Classification. In: *12th Annual Conference of the International Speech Communication Association, Florence*, 2011
- [GZL⁺09] GEORGI, A. ; ZIMMERMANN, M. ; LICH, T. ; BLANK, L. ; KICKLER, N. ; MARCHTHALER, R.: New Approach of Accident Benefit Analysis for Rear End Collision Avoidance and Mitigation Systems. In: *International Technical Conference on the Enhanced Safety of Vehicles*, 2009
- [Har08] HARPER, C. B.: Forward Collision Warning System NCAP Confirmation Test / U.S. Department of Transportation, National Highway Traffic Safety Administration. 2008. – Forschungsbericht
- [HBZ10] HÜLSEN, M. ; BÖRGER, J. ; ZÖLLNER, M.: Identification of relevant features in complex road traffic situations. In: *26. VDI/VW-Gemeinschaftstagung Fahrerassistenz und integrierte Sicherheit*, 2010
- [HKM⁺94] HUANG, T. ; KOLLER, D. ; MALIK, J. ; OGASAWARA, G. ; RAO, B. ; RUSSELL, S. ; WEBER, J.: Automatic symbolic traffic scene analysis using belief networks. In: *12th national conference on Artificial intelligence* Bd. 2, American Association for Artificial Intelligence, 1994, S. 966–972
- [HTF01] HASTIE, T. ; TIBSHIRANI, R. ; FRIEDMAN, J.H.: *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2001 (Springer series in statistics)

- [HW09] HÄRING, J. ; WILHELM, U.: Situation-Interpretation As A Key Enabler For Cost-Effective And Low-Risk Driver Assistance Systems With High Collision Mitigation Capabilities. In: *21st International Technical Conference on the Enhanced Safety of Vehicles*, 2009
- [HWB09] HÄRING, J. ; WILHELM, U. ; BRANZ, W.: Entwicklungsstrategie für Kollisionswarnsysteme im Niedrigpreis-Segment. In: *ATZ - Automobiltechnische Zeitschrift* Bd. 3, 2009
- [HXX⁺04] HU, W. ; XIAO, X. ; XIE, D. ; TAN, T. ; MAYBANK, S.: Traffic accident prediction using 3-D model-based vehicle tracking. In: *IEEE Transactions on Vehicular Technology*, 53 (2004), Nr. 3, S. 677 – 694
- [JJG02] JANSSON, J. ; JOHANSSON, Jonas J. ; GUSTAFSSON, F.: Decision Making for Collision Avoidance Systems. In: *SAE 2002 World Congress and Exhibition, Detroit* (2002)
- [KA04] KUMAGAI, T. ; AKAMATRU, M.: Modeling and Prediction of Driving Behavior. In: *2nd International Symposium on Measurement, Analysis and Modelling of Human Functions. 1st Mediterranean Conference on Measurement* (2004)
- [Kap98] KAPADIA, S.: *Discriminative Training of Hidden Markov Models*, University of Cambridge, Diss., 1998
- [KHB09] KUEHN, M. ; HUMMEL, T. ; BENDE, J.: Benefit estimation of advanced driver assistance systems for cars derived from real-life accidents. In: *21st International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV), Stuttgart, Germany*, 2009
- [KLF05] KIEFER, R. J. ; LEBLANC, D. J. ; FLANNAGAN, C. A.: Developing an inverse time-to-collision crash alert timing approach based on drivers' last-second braking and steering judgments. In: *Accident Analysis & Prevention* 37 (2005), Nr. 2, S. 295 – 303
- [LHH04] LEE, J. D. ; HOFFMAN, J. D. ; HAYES, E.: Collision warning design to mitigate driver distraction. In: *SIGCHI conference on Human factors in computing systems*, 2004
- [LM04] LI, Z. ; MILGRAM, P: Manipulating optical looming to influence perception of Time-to-collision and its application in automobile driving. In: *48th Annual Meeting on Human Factors and Ergonomics Society*, 2004
- [MDPB09] MEYER-DELIUS, D. ; PLAGEMANN, C. ; BURGARD, W.: Probabilistic Situation Recognition for Vehicular Traffic Scenarios. In: *IEEE International Conference on Robotics and Automation, Kobe, Japan* (2009)

- [Mit90] MITSCHKE, M.: *Dynamik der Kraftfahrzeuge. Band C: Fahrverhalten*. Springer Verlag, 1990
- [Mit04] MITROVIC, D.: *Learning Driving Patterns to Support Navigation*, University of Canterbury, Diss., 2004
- [Mit05] MITROVIC, D.: Reliable Method for Driving Events Recognition. In: *IEEE Transactions on Intelligent Transportation Systems* 6 (2005), Nr. 2
- [MPPM07] MENG, H. ; PITTERMANN, J. ; PITTERMANN, A. ; MINKER, W.: Combined Speech-Emotion Recognition for Spoken Human-Computer Interfaces. In: *IEEE International Conference on Signal Processing and Communications*, 2007, S. 1179–1182
- [MS05] MANDALIA, Hiren M. ; SALVUCCI, Dario D.: Using support vector machines for lane change detection. In: *Human Factors and Ergonomics Society 49th Annual Meeting, Orlando*, 2005
- [Mur02] MURPHY, K. P.: *Dynamic Bayesian Networks: Representation, Inference and Learning*, University of California, Berkeley, Diss., 2002
- [MW69] MCRUER, D. ; WEIR, D.H.: Theory of Manual Vehicular Control. In: *Man-Machine Systems, IEEE Transactions on* 10 (1969), Nr. 4, S. 257 –291
- [MWH07] MAGES, M. ; WINNER, H. ; HOPSTOCK, M.: Aktive Sicherheit durch Kreuzungsassistentz. Erkennung von Gefahrensituationen, Auswahl von Warnstrategien und Ableitung von Sensoranforderungen. In: *VDI Tagung: Der Fahrer im 21. Jahrhundert: Human Machine Interface*, 2007
- [NPVB05] NGUYEN, N. T. ; PHUNG, D. Q. ; VENKATESH, S. ; BUI, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Bd. 2*, 2005
- [OP00] OLIVER, N. ; PENTLAND, A.: Graphical Models for Driver Behavior Recognition in a SmartCar. In: *IEEE International Conference on Intelligent Vehicles*, 2000, S. 7–12
- [ORBD08] OH, S. M. ; REHG, J. M. ; BALCH, T. ; DELLAERT, F.: Learning and Inferring Motion Patterns using Parametric Segmental Switching Linear Dynamic Systems. In: *International Journal of Computer Vision* 77 (2008), Nr. 1-3, S. 103 – 124
- [Par62] PARZEN, E.: On Estimation of a Probability Density Function and Mode. In: *The Annals of Mathematical Statistics* 33 (1962), Nr. 3, S. 1065–1076

- [PK08] PFISTER, Beat ; KAUFMANN, Tobias: *Sprachverarbeitung*. Springer Verlag, 2008
- [PL99] PENTLAND, A. ; LIU, A.: Modeling and prediction of human behavior. In: *Neural Computation* 11 (1999), Nr. 1, S. 229–242
- [PTAA07] POLYCHRONOPOULOS, A. ; TSOGAS, M. ; AMDITIS, Angelos A. ; ANDREONE, L.: Sensor Fusion for Predicting Vehicles' Path for Collision Avoidance Systems. In: *IEEE Transactions on Intelligent Transportation Systems* 8 (2007), Nr. 3, S. 549–562
- [Rab89] RABINER, Lawrence R.: A tutorial on hidden markov models and selected applications in speech recognition. In: *Proceedings of the IEEE*, 1989, S. 257–286
- [RN02] RUSSEL, S. ; NORVIG, P.: *Artificial Intelligence: A modern approach (2nd edition)*. Prentice Hall, 2002
- [Rob03] ROBERT BOSCH GMBH: *Bosch ACC Adaptive Cruise Control*. Gelbe Reihe, 2003
- [Rus94] RUSSELL, J. A.: Is there universal recognition of emotion from facial expression? A review of the cross-cultural studies. In: *Psychological Bulletin* 115 (1994), S. 102–141
- [SF08] SCHULZ, A. ; FRÖMING, R.: Analyse des Fahrerverhaltens zur Darstellung adaptiver Eingriffsstrategien von Assistenzsystemen. In: *ATZ* 12 (2008)
- [SH82] SCHIMMELPFENNIG, K.-H. ; HEBING, N.: Geschwindigkeiten bei kreisförmiger Kurvenfahrt/Stabilitäts- und Sicherheitsgrenze. In: *Der Verkehrsunfall* 20 (1982), S. 97 – 99
- [SKB09] SCHMIDT, G. J. ; KHANAFER, A. ; BALZER, D.: Successive Categorization of Perceived Urgency in Dynamic Driving Situations. In: *SAE International* (2009)
- [SKSM05] STANLEY, K. ; KOHL, N. ; SHERONY, R. ; MIIKKULAINEN, R.: Neuroevolution of an Automobile Crash Warning System. In: *Genetic and Evolutionary Computation Conference*, 2005
- [SOB06] SCHMIDT, C. ; OECHSLE, F. ; BRANZ, W.: Research on trajectory planning in emergency situations with multiple objects. In: *Intelligent Transportation Systems Conference*, 2006
- [SPBS98] SPORRER, A. ; PRELL, G. ; BUCK, J. ; SCHAIBLE, S.: Realsimulation von Spurwechselforgängen im Straßenverkehr. In: *Verkehrsunfall und Fahrzeug-*

technik 36 (1998), S. 69 – 76

- [SSI08] SCHORN, M. ; STAHLIN, U. ; ISERMANN, Rolf: Anticollision system PRORETA with automatic braking and steering. In: *Vehicle System Dynamics* 46 (2008), Nr. 1, S. 683 – 694
- [SSP09] SCHERER, S. ; SCHWENKER, F. ; PALM, G.: Classifier Fusion for Emotion Recognition from Speech. In: *Advanced Intelligent Environments*. Springer US, 2009
- [Ste11] STEINKE, A.: *Analyse von Fahrverhalten mit Hidden Markov Models*, Mathematisches Institut der Albert-Ludwig-Universität Freiburg, Diplomarbeit, 2011
- [TGS⁺08] TORKKOLA, K. ; GARDNER, M. ; SCHREINER, C. ; ZHANG, K. ; LEIVIAN, B. ; ZHANG, H. ; SUMMERS, J.: Understanding Driving Activity Using Ensemble Methods. In: *Computational Intelligence in Automotive Applications*. 2008, S. 39–58
- [TVL04] TORKKOLA, K. ; VENKATESAN, S. ; LIU, H.: Sensor Selection for Maneuver Classification. In: *7th International IEEE Conference on Intelligent Transportation Systems*, 2004
- [TVL06] TORKKOLA, K. ; VENKATESAN, S. ; LIU, H.: Sensor Sequence Modeling for Driving. In: *IEEE International Joint Conference on Neural Networks, part of the IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada*, 2006, S. 4734–4739
- [Uns00] UNSER, M.: Sampling-50 years after Shannon. In: *Proceedings of the IEEE* 88 (2000), Nr. 4, S. 569–587
- [vH95] VAN DER HORST, R. ; HOGEMA, J.: Time-To-Collision and Collision Avoidance Systems. In: *6th ICTCT workshop Salzburg*, 1995
- [VPS07] VEERARAGHAVAN, H. ; PAPANIKOLOPOULOS, N. ; SCHRATER, P. R.: Learning Dynamic Event Descriptions in Image Sequences. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007
- [Wil06] WILHELM, Ulf: Optimierung des Akzeptanz-Nutzenverhältnisses am Beispiel der „Predictive Collision Warning“. In: *VDI Gemeinschaftstagung Integrierte Sicherheit und Fahrerassistenzsysteme* (2006)
- [WN99] WIACEK, C. ; NAJM, W.: Driver/Vehicle Characteristics in Rear-End Pre-crash Scenarios Based on the General Estimates System (GES). In: *SAE International Congress and Exposition*, 1999

Danksagung

Diese Doktorarbeit entstand in Zusammenarbeit mit dem Institut für Neuroinformatik der Universität Ulm, dem Heidelberg Collaboratory for Image Processing sowie dem Bereich Fahrerassistenzsysteme des Robert Bosch Standorts Leonberg. Die Kooperation mit vielen Beteiligten und deren jeweiligen Vertiefungsgebieten brachte vielfältige Aspekte ein.

Als erstes möchte ich mich herzlich bei meinem Doktorvater Prof. Günther Palm bedanken, durch dessen fachliche Unterstützung es erst möglich war, das Thema Fahrerassistenz in die Domäne der Sprachverarbeitung zu übertragen und mit bewährten sowie neuen Methoden zu bearbeiten. Durch die intensive Betreuung wurden meine Ideen stets wieder auf den zielführenden und systematischen Pfad zurückgebracht. Darüberhinaus möchte ich mich bei Prof. Wolfgang Minker für die Übernahme des Korreferats und Prof. Fred Hamprecht für die Kooperation mit der Universität Heidelberg bedanken.

Ein weiteres Dankeschön gilt meinen Kollegen der Robert Bosch Standorte Leonberg und Schwieberdingen, die mir mit fachlichem Wissen, Ideenfindung und Versuchen sehr geholfen haben. Hier möchte ich besonders meinen Betreuer Dr. Jürgen Häring erwähnen; er hat den wesentlichen Beitrag dazu geleistet, von vielen möglichen Ideen die relevanten herauszufiltern und erfolgreich zu verfolgen.

Viele Untersuchungen, Kontakte zu weiteren Forschungsgruppen, neue Ideen und Sichtweisen verdanke ich meinen Diplomanden und studentischen Mitarbeitern Tobias Dittman, Anne Steinke und Nino Häberlen. Ihre Ergebnisse sind an vielen Stellen mit eingeflossen.

Zu guter Letzt gebührt ein besonderer Dank meinen Eltern, Geschwistern und meinen Freunden Björn, Britta, Dominik, Kathrin, Michael, Sarah und Torsten - dank eurer fachlichen Unterstützung und mentalen Energieschubs in wichtigen Momenten konnte mir diese Arbeit gelingen und hat gleichzeitig viel Freude bereitet.

Joachim Börger

Lebenslauf

- 2007 - 2012 Doktorarbeit bei der Robert Bosch GmbH Leonberg
Bereich Radar- und Videobasierte Fahrerassistenzsysteme
Universitäten Ulm und Heidelberg
- 2006 Informatik-Diplom, RWTH Aachen
- 2006 Diplomarbeit bei der Virtual Reality Group, Lehrstuhl für Hochleistungsrechnen der RWTH Aachen: Interaktive Bewegungssynthese für realistische Menschenmodelle in virtuellen Umgebungen
- 2003 - 2004 Auslandsstudium an der INSA Lyon im Rahmen des Socrates-Programms
- 1999 – 2006 Studium der Informatik mit Anwendungsfach Psychologie an der RWTH Aachen
- 1999 Abitur in Mathematik, Chemie, Geschichte und Deutsch
- 1990 – 1999 Kaiser-Karls-Gymnasium in Aachen
- 26.09.1980 Geboren in Aachen