

Probabilistisch-logische Inferenz relationaler Situationsbeschreibungen aus Verkehrsbildfolgen

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)
genehmigte

Dissertation

von

DIPL.-ING. IRINA HENSEL

aus Sofia

Hauptreferent:

Prof. Dr.-Ing. C. Stiller

Korreferent:

Prof. Dr.-Ing. J. M. Zöllner

Tag der mündlichen Prüfung: 11. 10. 2013

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Mess- und Regelungstechnik des Karlsruher Instituts für Technologie unter der Leitung von Prof. Dr.-Ing. Christoph Stiller. Ihm danke ich für die wissenschaftliche Freiheit, die in Kombination mit dem angenehmen Arbeitsklima und der fortlaufenden Unterstützung die Dissertation in dieser Form ermöglicht hat.

Herrn Prof. Dr.-Ing. J. Marius Zöllner gilt mein Dank für die Übernahme des Korreferats und dem Interesse an meiner Arbeit.

Bedanken möchte ich mich bei allen Kolleginnen und Kollegen am MRT für die gute Zusammenarbeit und für zahlreiche spannende Diskussionen in den Kaffeerunden und auf den Sommerseminaren. Mein besonderer Dank gilt meinen Kollegen Alexander Bachmann und Philip Lenz für die Bereitstellung von Ergebnissequenzen aus ihrer Forschungstätigkeit. Weiterhin möchte ich mich bei meiner Kollegin Britta Hummel für die Unterstützung zu meiner Anfangszeit am MRT, das fachliche Interesse an meiner Arbeit und das mühevollen Korrekturlesen der Dissertation bedanken. Nicht zuletzt gebührt dem Sekretariat umfangreicher Dank für die Hilfe in allen Verwaltungsbelangen sowie den Mitarbeitern der Werkstätten und Herrn Werner Paal für die Hilfe in allen praktischen Belangen.

Mein herzlicher Dank gilt meinen Eltern sowie Schwiegereltern für die tatkräftige Unterstützung in all den Jahren. Der größte Dank gilt meinem Ehemann Stefan Hensel für seine Hilfe sowohl bei fachlichen als auch bei nicht fachlichen Fragestellungen und für seine Geduld.

Karlsruhe, im Oktober 2013

Irina Hensel

Kurzfassung

Im modernen Automobilbereich gewinnen Fahrerassistenzsysteme mit maschineller Umfeldwahrnehmung zunehmend an Bedeutung. Dabei ist die Tendenz zu einem stetig steigenden Automatisierungsgrad der Fahrerassistenz zu beobachten. Infolgedessen wachsen auch die Anforderungen hinsichtlich der Komplexität der zu bewältigenden Verkehrsszenarien. Fahrzeuge, die in der Lage sind Unfälle zu vermeiden, müssen nicht nur ihre Umwelt sensorisch erfassen, sondern vielmehr die aktuelle Situation analysieren, interpretieren und im Hinblick auf das eigene Handeln bewerten können. Eine wissensbasierte Beschreibung der Verkehrssituation, mit allen darin vorkommenden Objekten, deren Attributen und Relationen untereinander, ist dabei für die Situationsinterpretation und ein darauf aufbauendes sicheres und zielgerichtetes Handeln unabdingbar.

Die vorliegende Arbeit präsentiert ein auf Markov-Logik-Netzen basierendes Modell, mit welchem eine mit Wahrscheinlichkeiten versehene Situationsbeschreibung einer variablen Verkehrsszene probabilistisch-logisch inferiert wird. Dabei werden Klasse und Paarrelationen aller interessierenden segmentierten Objekthypothesen der Szene in einem Inferenzschritt geschätzt. Der probabilistisch-logische Formalismus Markov-Logik-Netze ermöglicht eine ausdrucksstarke relationale Repräsentation von komplexen Diskursbereichen, in welcher Unsicherheiten modelliert werden können und probabilistisches Schließen durchgeführt werden kann. Der entwickelte Ansatz wird anhand exemplarisch verwendeter Videosequenzen evaluiert.

Schlagnworte: Fahrerassistenzsysteme – Markov-Logik-Netze – Situationsrepräsentation – Kognitives maschinelles Sehen

Abstract

In modern automotive systems, driver assistance based on machine perception is increasingly gaining importance. This coincides with a steady rise in automation concerning the assistance function. As a result, the demand on behalf of the complexity of the traffic scenarios to be handled grows. Vehicles capable of avoiding accidents, do not only have to be able to perceive their environment with sensors, but to analyze and interpret the current situation in terms of their own actions. A knowledge-based description of the traffic situation, including all occurring objects, their attributes and interrelations is therefore essential for scene interpretation and successive actions that are safe and reasonable.

This thesis presents a model based on Markov Logic Networks. It is used for probabilistic-logical based inference of a situation description enhanced with probabilities given variable traffic scenes. For each segmented object hypothesis, class and relations to other object are estimated within a single inference step. The probabilistic-logical formalism Markov Logic Networks allows for an expressive relational representation of complex domains. In this representation it is possible to model uncertainties and conduct probabilistic reasoning, which is exemplarily evaluated on processed video sequences.

Keywords: Driver assistance system – Markov-Logic-Networks – Situation representation – Cognitive vision

Inhaltsverzeichnis

Symbolverzeichnis	IX
1 Einleitung	1
1.1 Einordnung der Arbeit	2
1.1.1 Begriffsdefinition	2
1.1.2 Kenntnisstand	3
1.1.2.1 Kognitives maschinelles Sehen	3
1.1.2.2 Probabilistisch-logisches Lernen	7
1.2 Ziele der Arbeit	10
1.3 Gliederung der Arbeit	11
2 Theoretische Grundlagen	13
2.1 Prädikatenlogik erster Stufe	13
2.1.1 Syntax	13
2.1.2 Semantik	14
2.1.3 Normalformen	17
2.1.4 Herbrand-Interpretation	17
2.1.5 Inferenz	19
2.2 Probabilistische Graphische Modelle	20
2.2.1 Bedingte Unabhängigkeit	21
2.2.2 Markov-Netze	22
2.2.3 Inferenz	25
2.2.4 Lernen	26
2.3 Markov-Logik-Netz	26
2.3.1 Markov-Logik	26
2.3.2 Inferenz	30

2.3.3	Parameterlernen	31
2.3.3.1	Generative Parameterlernverfahren	32
2.3.3.2	Diskriminative Parameterlernverfahren	33
3	Modellierung	37
3.1	Ontologie	37
3.1.1	Konzepte	39
3.1.2	Relationen	41
3.1.3	Formalisierung der Ontologie	43
3.2	Modell: Objektklassifikation (OK)	45
3.2.1	Axiome	45
3.2.2	Regeln	46
3.3	Modell: Relationale Objektklassifikation (ROK)	46
3.3.1	Axiome	47
3.3.2	Regeln	47
3.4	Modell: Inferierte Objektrelationen (IR)	48
3.4.1	Axiome	48
3.4.2	Regeln	50
3.5	Modell: Direkte Objektrelationen (DR)	50
4	Qualitative Abstraktion der Objektattribute	53
4.1	Qualitative Abstraktion der einfachen Objektattribute	53
4.2	Qualitative Abstraktion der relativen Objektattribute	55
5	Statistische Gütekriterien	59
5.1	Mehrklassen-Genauigkeit	60
5.2	Mehrklassen-AUC	61
5.3	Bedingte Log-Likelihood (CLL)	61
6	Lernen der Modellparameter	63
6.1	Trainingsdaten	63
6.2	Vergleichsuntersuchung der Lernverfahren	65

7	Inferenzergebnisse	69
7.1	Inferenz anhand simulierter Daten	70
7.1.1	Manuell annotierte Testdaten	70
7.1.2	Evidenzdaten	72
7.1.3	Methodik	73
7.1.4	Ergebnisse	74
7.2	Inferenz anhand experimentell gewonnener Daten	78
7.2.1	Experimentell gewonnene Testdaten	78
7.2.2	Evidenzdaten	81
7.2.3	Methodik	81
7.2.4	Ergebnisse	83
8	Zusammenfassung und Ausblick	101
	Literaturverzeichnis	104

Symbolverzeichnis

Abkürzungen

AUC	<u>A</u> rea <u>U</u> nder <u>C</u> urve
CLL	<u>C</u> onditional <u>L</u> og- <u>L</u> ikelihood
DN	<u>D</u> iaagnoal <u>N</u> ewton
DR	Modell „ <u>D</u> irekte <u>O</u> bjekt <u>r</u> elationen“
IR	Modell „ <u>I</u> nerierte <u>O</u> bjekt <u>r</u> elationen“
IROK	MLN „ <u>I</u> nerierte <u>O</u> bjekt <u>r</u> elationen mit <u>O</u> bjekt <u>k</u> lassifikation“
KNF	<u>K</u> onjunktive <u>N</u> ormal <u>f</u> orm
MAP	<u>M</u> aximum <u>a</u> <u>P</u> osteriori
MCMC	<u>M</u> arkov <u>C</u> hain <u>M</u> onte <u>C</u> arlo
MC-SAT	<u>M</u> arkov <u>C</u> hain <u>SAT</u> -Solver
MLN	<u>M</u> arkov <u>L</u> ogik <u>N</u> etz
MPE	<u>M</u> ost <u>P</u> robable <u>E</u> xplanation
OK	Modell „ <u>O</u> bjekt <u>k</u> lassifikation“
PL1	<u>P</u> rädikaten <u>l</u> ogik der ersten Stufe
PSCG	<u>P</u> reconditioned <u>S</u> caled <u>C</u> onjugate <u>G</u> radient
ROC	<u>R</u> eceiver <u>O</u> perator <u>C</u> haracteristic
ROK	Modell „ <u>R</u> elationale <u>O</u> bjekt <u>k</u> lassifikation“
VP	<u>V</u> oted <u>P</u> erceptron
WalkSAT	<u>W</u> alk <u>SAT</u> -Solver

1 Einleitung

Im modernen Automobilbereich gewinnen Fahrerassistenzsysteme mit maschineller Umfeldwahrnehmung zunehmend an Bedeutung. Der Begriff fasst Systeme zusammen, die neben dem Zugewinn an Komfort und Effizienz auch einen wesentlichen Beitrag zur aktiven Sicherheit, also der Minderung von Unfallfolgen oder sogar der vollständigen Unfallvermeidung, beitragen. Beispiele solcher Systeme sind die automatische Abstandsregelung, Spurhalte-, Spurwechsel-, Park- oder auch Notbremsassistent. Die Funktionalitäten der einzelnen Assistenzsysteme erfordern ein unterschiedlich tiefes Verständnis der aktuellen Verkehrssituation und bedingen einen unterschiedlich starken Eingriff in die Fahrzeugführung. Dabei ist die Tendenz zu einem stetig steigenden Automatisierungsgrad der Fahrerassistenz zu beobachten. Infolgedessen wachsen auch die Anforderungen hinsichtlich der Komplexität der zu bewältigenden Verkehrsszenarien, was den Einsatz neuer Methoden zur Wissensrepräsentation und -inferenz motiviert.

Fahrzeuge, die in der Lage sind Unfälle zu vermeiden, müssen im Allgemeinen die Fähigkeiten *Kognitiver Automobile* besitzen, welche im gleichnamigen Sonderforschungsbereich (SFB) Transregio der Deutschen Forschungsgemeinschaft [Sti07], dem auch die vorliegende Arbeit ihren Ursprung verdankt, erforscht wurden. Ein kognitives Automobil ist ein autonom handelnder Agent, der nicht nur seine Umwelt sensorisch erfassen, sondern vielmehr die aktuelle Situation analysieren, interpretieren und im Hinblick auf das eigene Handeln bewerten kann. Dazu muss er über Wissen seiner eigenen Fähigkeiten und seiner Umwelt verfügen. Eine wissensbasierte Beschreibung der Verkehrssituation, mit allen darin vorkommenden Objekten, deren Attributen und Relationen untereinander, ist dabei für seine Situationsinterpretation und ein darauf aufbauendes sicheres und zielgerichtetes Handeln unabdingbar.

Eine solche ganzheitliche relationale Situationsrepräsentation wird nicht nur für autonom fahrende Fahrzeuge benötigt. Die Notwendigkeit dieser wird mit der zunehmenden Komplexität der für die Fahrerassistenzsysteme relevanten Verkehrsszenarien immer deutlicher [Hül11]. Einerseits, stößt die heutzutage bei den Fahrerassistenzsystemen vorherrschende Architektur der unabhängigen Einzellösungen zur Umfelderfassung und -interpretation an ihre Grenzen. Die im Fahrzeug parallel laufenden verschiedenen Fahrerassistenzsysteme berechnen unabhängig voneinander, oft redundant, die Informationen, die sie zur Ausführung ihrer jeweiligen Funktionalität brauchen, meistens ohne das gewonnene Wissen auszut-

auschen. Bei komplexen, sicherheitsrelevanten Assistenzsystemen reicht eine solche dezentrale Situationserkennung nicht aus, um konsistente Entscheidungen der einzelnen Systeme zu garantieren [Sch10]. Andererseits, wird die verbreitete Herangehensweise zur Einschätzung eines potentiell gefährlichen Kollisionsobjektes dahingehend erweitert, dass mehrere Objekte simultan in die Betrachtung miteinbezogen werden. Eine Auswertung der Unfallstatistik ergibt, dass die Analyse der Multiobjektszenarien in vielen Fällen zusätzliche Informationen liefert, so dass eine frühzeitige Einschätzung des Gefahrenpotentials der Situation ermöglicht wird [Hae09].

Die Wissensmodellierung einer relationalen Situationsbeschreibung ist in der Kognitions-kette als Schnittstelle zwischen der sensorischen Umfelderkennung und der Situationsinterpretation und Verhaltensgenerierung zu betrachten [Sti08]. Sie überführt die quantitativen Sensordaten der Wahrnehmung auf die begriffliche Ebene, ermöglicht die Inferenz von Objektrelationen und ist eine formale Grundlage, auf die die Situationsinterpretation aufsetzt. Eine Rückkopplung von der abstrakten Symbol- auf die Signalebene dient zur Verbesserung der sensornahen Signalverarbeitung [Bac09].

In der vorliegenden Arbeit wird ein Ansatz zur Inferenz einer mit Wahrscheinlichkeiten versehenen, relationalen Verkehrssituationsbeschreibung aus sensorisch erfassten Umgebungsdaten entwickelt und anhand exemplarisch verwendeter Videosequenzen validiert. Hierfür wird ein probabilistisch-logischer Formalismus auf die Anwendbarkeit als Wissensrepräsentationssprache untersucht. Somit ist diese Arbeit im Bereich des kognitiven maschinellen Sehens [Nag04] angesiedelt.

1.1 Einordnung der Arbeit

Dieses Kapitel beginnt mit einer Definition der für die Arbeit relevanten Begriffe und stellt anschließend eine Literaturübersicht aus den Bereichen „Kognitives maschinelles Sehen“ und „Probabilistisch-logisches Lernen“ vor.

1.1.1 Begriffsdefinition

Begriffe, wie Szene oder Situation, sind im Bereich der Fahrerassistenz nicht eindeutig festgelegt. In der Literatur existieren diesbezüglich viele verschiedene Definitionen, wie der Übersicht aus [Sch10] entnommen werden kann. Um Missverständnisse zu vermeiden, werden die einzelnen Begriffe zunächst definiert.

In der vorliegenden Arbeit wird die Definition aus [Vac09] übernommen, welche an [Haa98] angelehnt ist. Demnach ist eine *Szene* der lokale, sensoruell wahrnehmbare Ausschnitt der Welt um das Eigenfahrzeug, inklusive einer Beschreibung aller erkannten Objekte durch einfache sensornahe Attribute, wie beispielsweise Objekthöhe oder -position. Eine Erweiterung dieser Repräsentation um Informationen, die nicht direkt sensoruell bestimmt werden können, definiert eine *Situation*. Damit gemeint sind sowohl geschätzte interne Objektzustände als auch inferierte Objektrelationen. Die anschließende *Situationsanalyse* setzt das Eigenverhalten in Beziehung mit der erkannten Situation. Dabei wird die Weiterentwicklung der Situation betrachtet, um die Auswirkungen eines möglichen Eigenverhaltens auf die aktuelle Situation einzuschätzen.

1.1.2 Kenntnisstand

Die vorliegende Arbeit befasst sich mit der Entwicklung einer relationalen Situationsrepräsentation mit Hilfe eines probabilistisch-logischen Formalismus. Somit ist sie im Bereich des kognitiven maschinellen Sehens verankert. Die für diesen Bereich relevanten Veröffentlichungen werden im Folgenden vorgestellt und diskutiert.

1.1.2.1 Kognitives maschinelles Sehen

Kognitives maschinelles Sehen (engl. *cognitive vision*) bezeichnet die explizite Nutzung von Wissen und Inferenzalgorithmen zusammen mit abstrahierten, aus der Umfeldsensierung gelieferten Daten [Ver06]. Solche Verfahren nutzen a priori Wissen über bestehende räumliche und zeitliche Objektrelationen, welches unterschiedlich modelliert wird, wie z. B. probabilistisch [How00], frame- [Hot08] oder logikbasiert [Neu08], mittels Situationsgraphenbäumen [Are04] oder Szenarien [Geo06]. Typische Anwendungen solcher Methoden schließen Überwachungssysteme, Verkehrssituationsanalyse oder inhaltsbasierte Bildersuche ein.

Für alle drei erwähnten Forschungsbereiche sind zuverlässige Objekterkennung und -klassifikation von großer Bedeutung. Bei komplexen Szenarien, wie im innerstädtischen Bereich, stellen die Vielzahl der möglichen Objekte, die Variabilität in der Erscheinung innerhalb einer Objektklasse sowie Teilverdeckungen einzelner Objekte allerdings die bildbasierte Objekterkennung und -klassifikation vor enorme Herausforderungen.

Typischerweise besteht der Prozess der Objektdetektion in Bildern aus zwei Schritten: Hypothesengenerierung und Hypothesenverifikation [Lai07]. Bei der

Hypothesengenerierung werden Hypothesen darüber aufgestellt, wo sich Objekte im Bild befinden können. Anschließend werden die generierten Hypothesen durch geeignete Tests verifiziert. Die Verfahren zur Hypothesengenerierung und -verifikation lassen sich grob in merkmals-, model-, stereo- und trainingsbasiert einteilen. Ein guter Überblick verschiedener Methoden ist in [Sun06] am Beispiel der Fahrzeugdetektion zu finden. Das Fazit dieses Artikels ist, dass gängige Algorithmen zur Objekterkennung und -klassifikation keine zufriedenstellende Ergebnisse in komplexen Umgebungen liefern. Sun et. al. empfehlen weiterhin Domänenwissen sowie zeitliche und räumliche Randbedingungen in den Detektionsprozess zu integrieren. In den letzten Jahren wird verstärkt in diese Richtung geforscht. Viele Veröffentlichungen berichten von verbesserten Objekterkennungs- und Klassifikationsraten durch Integration von Kontextinformation als zusätzliches Merkmal, ergänzend zu traditionellen Bilddeskriptoren [Son11]. Der Begriff *Kontext* bedeutet im Bereich des maschinellen Sehens sowohl Angaben zu Umweltbedingungen, geographische Lage oder Zeit, als auch bildbasierte Informationen, wie lokale Pixelcharakteristika, globale Szenenmerkmale, geometrische oder semantische Angaben zu Objektkonfigurationen [Div09].

Ein hierarchisches System basierend auf Conditional Random Fields ist in [Kum05] präsentiert, welches sowohl lokalen Bildkontext als auch relative räumliche Konfigurationen von gelabelten Bildregionen für die Objektdetektion berücksichtigt. [Car04] schlägt ein Markov Random Field, auch Markov-Netz genannt, vor, das räumliche Nachbarschaftsbeziehungen von Objekten in Bildern modelliert, um die Klassifikation von segmentierten Bildregionen in semantischen Kategorien, wie z. B. „sky“ oder „water“, zu verbessern. In [Li07] wurde ein Bayessches Netz mit dem gleichen Ziel entwickelt. Weitere Verfahren modellieren Relationen zwischen den Teilen der zu klassifizierenden Objekte [Hub04], [Ber05].

Im Gegensatz zu den bisher referenzierten Arbeiten verwenden die folgenden explizite Wissensmodellierung, um Kontextinformationen in die Objektdetektion zu integrieren. In [Tsa07] beispielsweise wird die relative räumliche Konfiguration von Bildregionen regelbasiert modelliert. [EM08] präsentieren einen ontologiebasierten Ansatz zur Klassifikation von Insekten aus Bildern. In [Ben08] werden explizite räumlich-zeitliche Konsistenzbedingungen mittels logischer Inferenz zur Verbesserung der Objekterkennung und -verfolgung in Videodaten berücksichtigt.

Nachfolgend werden Arbeiten vorgestellt, die sich ebenfalls mit wissensbasierter Inferenz aus Bilddaten beschäftigen, welche allerdings über das Thema Objektdetektion hinaus in Richtung Szenenverstehen und Extraktion von Objektverhalten gehen. In [Dan93] ist einer der ersten Ansätze beschrieben, der bildbasiert mit aus Luftaufnahmen gewonnener Information über die Bewegung der Objekte auf

einer innenstädtischen Kreuzung mit wissensbasierten Verfahren kombiniert. Das Wissen wird durch ein semantisches Netz zusammen mit regelbasierter Auswertung der Prädikate modelliert. Die Ausgabe des Systems liefert eine Interpretation der Verkehrsregeln aufgrund der Konfiguration der Autos in der Kreuzung. In [How00] wird ein modelbasiertes Trackingverfahren genutzt, um die begriffliche Repräsentation von Geschehen an einem Kreisverkehr (following, overtaking-passing) abzuleiten. In [Mie04b] und [Geh05] ist eine qualitative Repräsentation einer Verkehrsszene vorgestellt. Es werden sogenannte Verkehrssituationsmuster, wie *collision_course* oder *overtake_manoeuvre*, als logische Formel definiert und mit Hilfe eines Prolog-ähnlichen Reasoners inferiert. Diese Arbeiten bauen auf der qualitativen räumlich-zeitlichen Bewegungsbeschreibung aus [Mie04a] auf, die keine Unsicherheiten berücksichtigen kann. [Coh06] stellen ein System vor, welches Verkehrsgeschehen aus qualitativ räumlichen Beziehungen zwischen nahen Fahrzeugen, die entlang bekannter Pfade fahren, lernen kann.

[Neu03] verwendet in Beschreibungslogik [Baa03] formalisierte Aggregate, um geometrische Objekte und deren Beziehungen zu beschreiben. Ziel ist in [Hum09] die wissensgetriebene Interpretation von innerstädtischen Straßenkreuzungen aus fahrzeugeigenen Sensordaten wie Kamera, Laserscanner, Navigationskarten und Ortungssensoren. Es wird argumentiert, dass in ein solches System ein beträchtliches Maß an explizit formuliertem Vorwissen einfließen muss. Dieses Vorwissen wird in Form einer Straßennetz-Ontologie ebenfalls in der Sprache der Beschreibungslogik formalisiert. Formallogische Inferenz in dieser Wissensbasis ermöglicht die Deduktion fehlender Sensordaten sowie die Generierung all derjenigen Kreuzungshypothesen, welche konsistent mit Sensordaten und Vorwissen sind. Der gleiche Formalismus wird in [Vac09] für eine ontologiebasierte Wissensmodellierung des Umfeldes aus Videodaten verwendet. Die in der Ontologie eingeführten Verhalten von Objekten, wie anhalten, fahren, bremsen, Fahrzeug folgen oder freies Fahren, werden in einer Vergleichsuntersuchung auf zwei Arten – mit einer Support Vector Machine und mit einem Bayesschen Netz, aus Sensordaten klassifiziert. Dieses Wissen fließt in einen Szenengraphen ein, der die Objekte der Szene mit ihren Attributen und Verhalten enthält. Die anschließende Interpretation greift auf diesen zurück und verwendet fallbasiertes Schließen, um das geeignete Verhalten eines autonomen Fahrzeuges zu inferieren.

[Pro04] verwenden ontologiebasierte Inferenz für Hindernisse auf der Straße, um die Routenplanung von autonomen Fahrzeugen zu verbessern. Eine Ontologie der Objekte der Umgebung gemeinsam mit Regeln für Abschätzung der Schäden, die ein Zusammenstoß mit unterschiedlichen Objekten in unterschiedlichen Situationen verursachen würden, werden an ein automatisches Inferenzsystem übergeben. Dessen Inferenzergebnis soll dem Routenplaner bei der Entscheidung helfen, ob ein Ausweichmanöver zu planen ist. [Fuc08] hat ein ontologiebasiertes Fahrerassistentensystem entwickelt, das die Routenplanung von autonomen Fahrzeugen verbessert. Eine Ontologie der Objekte der Umgebung gemeinsam mit Regeln für Abschätzung der Schäden, die ein Zusammenstoß mit unterschiedlichen Objekten in unterschiedlichen Situationen verursachen würden, werden an ein automatisches Inferenzsystem übergeben. Dessen Inferenzergebnis soll dem Routenplaner bei der Entscheidung helfen, ob ein Ausweichmanöver zu planen ist. [Fuc08] hat ein ontologiebasiertes Fahrerassistentensystem entwickelt, das die Routenplanung von autonomen Fahrzeugen verbessert.

sistenzsystem zur Inferenz der Sicherheit eines Überholmanövers in der aktuellen Verkehrssituation entwickelt und im Rahmen einer Simulation validiert. Das in [Bre11] vorgeschlagene Verfahren überführt Klassifikationsergebnisse für die Objekte einer Verkehrsszene zusammen mit einer Menge Objektattributen auf die begriffliche Ebene und bildet diese auf Konzepte und Prädikate der *OpenCyc*-Ontologie ab. Anschließend werden räumlich-zeitliche Regeln zur Objektklassifikation und Szenenverstehen ausgewertet und eine Beschreibung der Verkehrssituation in *CycL* oder in einer quasi-natürlichen Sprache ausgegeben. Dabei werden relative Objektattribute, wie relative Position, relativer Abstand oder relative Orientierung, und auf ein Objekt bezogene Geschehen, wie *performedBy* oder *running* inferiert.

Die sensorisch beobachtete Verkehrsszene aus einem Fahrzeug heraus wird in [Dic98] hierarchisch in einer Dynamischen Objektdatenbasis (DOB) im Rahmen des 4D-Ansatzes dargestellt. Dabei wird die räumliche Beziehung der Objekte rein geometrisch kodiert. Objektzustand und Relationen zwischen Objekten der Szene werden in [Pel03] weiterentwickelt und mit Hilfe von Situationsaspekten modelliert. Ein Situationsaspekt stellt eine linguistische Variable im Sinne der Fuzzy-Theorie dar. Die explizit repräsentierte Situationsbeschreibung wird anschließend einer autonomen Verhaltensentscheidung zugänglich gemacht. Die Arbeit von Schneider [Sch10] beschreibt eine unscharfe Modellierung der Objektmerkmale zur Erkennung von Fahrsituationen (z. B. Annäherungsfahrt) und Fahrmanövern (z. B. Spurwechsel) mittels Fuzzy-Zugehörigkeitsfunktionen. Für jede Situation oder jedes Manöver wird je ein Bayessches Netz erstellt, mit welchem eine Wahrscheinlichkeit inferiert wird. Aus der Informationsqualität der Sensoren ein Qualitätsmaß der Situationsentscheidung berechnet und damit die Sensorungenauigkeiten im Modell mitberücksichtigt. Der Ansatz wird am Beispiel eines Notbremsassistenten validiert. Ein anderes System, präsentiert in [Ger08], importiert extrahierte geometrische Trajektorien in eine Repräsentation elementarer Verkehrsgeschehen, die auf Fuzzy metrisch-temporaler Logik basiert. Der gleiche Wissensrepräsentationsformalismus wird in [Fer08] als Basis für die Architektur eines Systems, welches natürlichsprachliche Beschreibungen der Geschehnisse an einem Zebrastreifen extrahiert, verwendet.

Die vorgestellten Ansätze basieren auf durchaus unterschiedlichen Methoden. Einige bearbeiten Videosequenzen, die mit stationär angebrachten Kameras über dem interessierenden Verkehrsbereich aufgenommen sind. Die verwendeten Modelle sind daher oft der konkreten Straßengeometrie angepasst und sind deshalb nicht übertragbar auf andere Szenarien. Ein Großteil der Arbeiten erkennen Geschehen, die das Verhalten eines einzelnen Objekts beschreiben, wie Beschleunigen oder Bremsen. Geschehen, die das relative Verhalten zwischen zwei Objekten

angeben, werden meistens für ein Referenzobjekt ausgewertet, nicht für alle Paare von erkannten Objekten in der beobachteten Szene. Weiterhin wurden viele der Ansätze „nur“ anhand simulierter Daten getestet und bleiben somit leider einen Beweis, auch mit realen Daten umgehen zu können, schuldig. Eine weitere Gemeinsamkeit besteht darin, dass alle Modelle, bis auf die Arbeiten, die sich der Fuzzy-Theorie oder graphischer Modelle bedienen, keine Unsicherheiten berücksichtigen können.

Probabilistische graphische Modelle, wie beispielsweise die bereits genannten Bayessche Netze, sind aufgrund ihrer Struktur sehr gut dazu geeignet, Wahrscheinlichkeitsverteilungen einer Variablenmenge darzustellen. Da sie aber trotz effizienter Algorithmen nur in der Lage sind, aussagenlogische Zusammenhänge über Objekte effizient zu repräsentieren, sind Generalisierungen über Objektgruppen mit ihnen nicht realisierbar [Rus03]. Eine daraus folgende Restriktion betrifft die Flexibilität der Netztopologie, so dass diese für unterschiedliche Instanzen des gleichen Problems jeweils manuell angepasst werden muss. Um eine umfassendere Darstellung zu ermöglichen und gleichzeitig von den gründlich untersuchten Lernverfahren und Inferenzalgorithmen der probabilistischen Netze zu profitieren, zielen Forschungsarbeiten im Gebiet des „Probabilistisch-logischen Lernen“ unter anderem darauf ab, die Unzulänglichkeiten der rein probabilistischen Formulierung durch die Kombination mit Prädikatenlogik der ersten Stufe zu überwinden.

1.1.2.2 Probabilistisch-logisches Lernen

Der Begriff *probabilistisch-logisches Lernen* (*probabilistic logic learning*, PLL) beschreibt ein jüngeres Forschungsgebiet der Künstlichen Intelligenz, welches relationale oder logische Repräsentationen, probabilistisches Schließen und Prinzipien des maschinellen Lernens integriert. In der Literatur werden auch die Begriffe *probabilistische induktive Logikprogrammierung* (*probabilistic inductive logic programming*), *statistisches relationales Lernen* (*statistical relational learning*) oder *multi-relationales Data-Mining* (*multi-relational data mining*) verwendet [Rae08]. Dabei werden *probabilistische* Repräsentations- und Schlussfolgerungsmechanismen, wie Bayessche Netze, Markov-Netze oder probabilistische kontextfreie Grammatiken verwendet. Diese werden mit formaler *Logik* - meist mit Prädikatenlogik und relationalen Repräsentationen verknüpft. Diese ermöglichen eine elegante Beschreibung komplexer Situationen samt aller darin vorkommenden Objekte und deren Relationen. Die Motivation für das *Lernen* liegt darin, dass es oft einfacher ist, für einen komplexen Diskursbereich Trainingsdaten zu generieren und ein Modell zu lernen, als ein Modell mit konventionellen Methoden des Wissensengineering zu erstellen. Somit hat PLL zum Ziel, probabilistische Modelle auf eine kompakte und intuitive Weise darzustellen, welche die relationale

Struktur des Diskursbereichs widerspiegelt und, idealerweise, effizientes Lernen und Schlussfolgern unterstützt [Get07a].

Im Bereich der statistischen Künstlichen Intelligenz werden Daten durch Punkte in einem hochdimensionalen Raum repräsentiert. Diese Abstraktion stellt sicherlich einen nützlichen Ansatz dar, solange die Aufgabe überschaubar und speziell bleibt. Für die Lösung von allgemeinen und komplexen Aufgaben reicht diese Abstraktion allerdings nicht aus, da sie die ausschlaggebende logische Struktur der zugrunde liegenden Daten verbirgt [Get07a]. Beispielsweise könnte man nicht nur das Vorhandensein einer Person in einem Bild erkennen wollen, sondern auch, dass diese Person ein Fußgänger ist, der sich auf einen Zebrastreifen zubewegt. Dazu werden Datenrepräsentationen benötigt, in denen Objekte durch deren Eigenschaften beschrieben werden und ein Teil von Relationen und Handlungen sein können. Solche relationalen Repräsentationsstrukturen besitzen die wichtige Fähigkeit, Abhängigkeiten zwischen Objekten zu modellieren. Verfügbare Information über ein Objekt hilft, Schlüsse über ein anderes, mit ihm in Beziehung stehendes Objekt, zu ziehen. Zum Beispiel, wäre es bei der Klassifikation von Verkehrsteilnehmern hilfreich, den Typ eines bereits klassifizierten Objekts zur Klassifikation von weiteren, zu diesem in Relation stehenden, Objekten hinzuzuziehen.

Dabei ist es erforderlich Unsicherheiten zu berücksichtigen, wenn man es mit experimentell gewonnenen Signalen, welche verrauscht und im Allgemeinen unvollständig sind (z.B. Verdeckungen im Bild), zu tun hat. Bei relationalen Modellen entstehen weitere Unsicherheiten bzgl. Objekttyp oder welche Objekte in welchen Relationen zueinander stehen. Daher ist eine geschickte Behandlung der Unsicherheit in relationalen Modellen für eine robuste Problemlösung unabdingbar [Get07a]. Eine geeignete Kombination von Methoden aus den Bereichen Logik, Lernen und Wahrscheinlichkeitstheorie kann diese Anforderungen erfüllen.

Paarweise Überschneidungen dieser drei Fachgebiete sind längst Thema intensiver Forschungsarbeiten. Zum einen wurden diverse Verfahren zum *probabilistischen (statistischen) Lernen*, wie gradientenbasierte Methoden oder EM-Algorithmus (Erwartungs-Maximierung), entwickelt und nahezu erschöpfend erforscht [Rus03]. Zum anderen beschäftigen sich die induktive logische Programmierung sowie das multi-relationale Data-Mining mit *logischem Lernen*, d.h. mit Lernen und Data-Mining innerhalb prädikatenlogischer oder relationaler Repräsentationen [Mug94], [Dze01]. Weiterhin wurde auch in Richtung *probabilistische Logik* geforscht oder wie integriert man Wahrscheinlichkeiten in Logikprogrammen. In [Poo93] wird probabilistische Logik aus der Perspektive der Wissensrepräsentation untersucht.

Erst in letzter Zeit sind Verfahren erschienen, die versuchen, alle drei PLL-Aspekte zu adressieren. Die meisten davon basieren auf Kombinationen von probabilistischen Graphen oder probabilistischen kontextfreien Grammatiken und logischen

Formeln oder Frames. Dazu gehören *Stochastic Logic Programs* (SLPs) [Mug02], *PRogramming In Statistical Modeling* (PRISM) [Sat95] und *Independent Choice Logic* (ICL) [Poo97], *Probabilistic Relational Models* (PRMs) [Fri99], *Bayesian Logic Programs* (BLPs) [Ker00], and *Markov Logic Networks* (MLNs) [Ric06a]. Einen Überblick über alle bisherigen PLL-Ansätze geben [Get07a] und [Rae08]. SLPs verbessern die Ausdrucksfähigkeit der stochastischen kontextfreien Grammatiken auf dem Level der Logikprogramme. Dabei kennzeichnen SLPs jede Klausel durch einen Wahrscheinlichkeitswert, wohingegen PRISM und ICL jedem instantiierten atomaren Satz eine Wahrscheinlichkeit zuordnen, aus welcher weiteres Wissen abgeleitet werden kann [Get07a]. Die probabilistische Semantik der restlichen oben genannten PLL-Systeme basiert auf graphischen Modellen. PRMs und BLPs nutzen gerichtete Graphen in Kombination mit Frames bzw. Logikprogrammen. Markov-Logik-Netze kombinieren dagegen ungerichtete Netzwerke mit Prädikatenlogik. Inferenzalgorithmen werden in solchen Modellen auf einem anhand aller Evidenzvariablen instantiierten Graphen angewendet. Bei der Inferenz wird nur der für die Anfragevariablen relevante Teil des instantiierten Netzes konstruiert [Rae08]. Auf diese Weise wird die aussagenlogische Ausdrucksfähigkeit der probabilistischen graphischen Modelle erweitert.

Nachfolgend wird auf bekannte Anwendungen des Formalismus Markov-Logik-Netze, der im Rahmen dieser Arbeit zur Wissensrepräsentation im Bereich des kognitiven maschinellen Sehens eingesetzt wird, eingegangen. Viele Beispiele für Anwendungen von Markov-Logik-Netznen kommen aus genetischen, web-orientierten oder sozialen Fragestellungen ([Lip08], [Rie05], [Rie08], [Sch08], [Sin08]). In [Sin06] wird ein MLN-Model zur automatischen Erkennung der sozialen Beziehungen von Personen, die in einer Menge von Bildern gemeinsam zu sehen sind, vorgestellt. Die kollektive Inferenz der Beziehungen über alle Bilder anhand der Evidenzdaten über Personenanzahl, -alter und -geschlecht erlaubt eine gegenseitige Beeinflussung der Komponenten und erzielt dadurch bessere Ergebnisse. Auch der Robotikbereich zeigt zunehmend Interessen an MLN-Anwendungen [Sin11]. Im Folgenden werden die wenigen MLN-Anwendungen referenziert, die auf bildbasierten Sensoren aufbauen. Eine erste Gruppe von Anwendungen stammt aus dem Bereich der Sicherheitstechnik und passiven Überwachung. In [Tra08] wird ein MLN vorgestellt, das die in einem Parkplatzvideo zu sehenden Personen an- und abfahrenden Fahrzeugen zuordnet. [Kem10] beschreibt ein MLN zur automatischen Erkennung und Annotation von Infrastrukturelementen, wie Straße, Bushaltestelle, Fußgängerüberweg und Bürgersteig, anhand der Trajektorien von Personen, PKWs und Busse in Videodaten von statisch installierten Kameras. Das Modell enthält Formeln, welche die zu identifizierenden Infrastrukturelemente definieren, sowie solche die allgemeines Wissen über die 3D-Szenengeometrie und auftretende Verdeckungen festhalten. In der Veröffentli-

chung von Chechetka et al. [Che10] wird ein MLN zur Personenidentifikation aus Innenraumüberwachungsvideos umrissen. Die zweite Gruppe von veröffentlichten MLN-Anwendungen basieren auf Videodaten, welche aus einem fahrenden Fahrzeug aufgenommen werden. In [Nie11] ist eine MLN-Anwendung vorgestellt, die die Relevanz von statischen Objekten, wie Verkehrsschilder oder -ampeln, in der Nähe des Ego-Fahrzeugs für die bestehende Verkehrssituation schätzt. [Sou11] beschreibt die Modellierung eines Spurassistent mit Markov-Logik.

1.2 Ziele der Arbeit

Ein Großteil aktueller Forschungsaktivitäten auf dem Gebiet der sensorbasierten Fahrerassistenz beschäftigen sich noch immer mit der Erfassung von Umgebungsinformationen. Der Frage, wie diese, insofern verfügbare, Informationen weiterverarbeitet und verwertet werden, wird bisher sehr viel weniger Aufmerksamkeit geschenkt. Die vorliegende Arbeit widmet sich dieser Frage. Es soll ein Markov-Logik-Netz entwickelt werden, mit welchem eine mit Wahrscheinlichkeiten versehene begriffliche Situationsbeschreibung einer veränderlichen Verkehrsszene aus sensoruell erfassten vorverarbeiteten Umfelddaten geschätzt werden kann. Diese Repräsentation soll als Schnittstelle zwischen der sensornahen Umfelderkennung und einer abstrakten Situationsinterpretation dienen. Das Modell soll daher auf segmentierte und über die Zeit verfolgte Objekthypothesen als Eingangsdaten aufbauen. Die inferierte Situationsbeschreibung soll die Wahrscheinlichkeiten der in einer Ontologie vordefinierten Objektklasse und -paarrelationen zu jeder Objekthypothese der Szene beinhalten.

Mit Hinblick auf die große Variabilität der vorhandenen umfeldwahrnehmenden Sensorplattformen im Bereich der Fahrerassistenzsysteme soll die entwickelte Situationsrepräsentation sensorunabhängig sein und Objektattribute verwenden, die aus verschiedenen Sensorsignalen gewonnen werden können. Diese Attribute sollen weiterhin möglichst einfach und sensorisch robust bestimmbar sein. Das Modell soll darüber hinaus in der Lage sein, Situationen mit Objekten unterschiedlicher Klassen und/oder Anzahl sowie auch aus unterschiedlichen Verkehrsraumumgebungen, wie Autobahn, Landstraße oder Stadt, zu beschreiben. Hierbei wird bei der Modellierung der Fokus auf innerstädtische Verkehrsszenen gelegt. Im Rahmen der Arbeit soll außerdem die Idee aus der Fachliteratur zur Berücksichtigung der relativen räumlichen Position von Bildregionen zur Verbesserung der Klassifikation aufgegriffen werden und auf dynamische Objektrelationen übertragen und untersucht werden. Schließlich ist die Leistungsfähigkeit des entwickelten Modells zur probabilistisch-logischen Inferenz von Verkehrssituationsbeschreibungen an simulierten und realen Daten zu evaluieren.

1.3 Gliederung der Arbeit

Kapitel 2 stellt die für die Arbeit relevanten theoretischen Grundlagen vor. Als erstes wird auf die Prädikatenlogik erster Stufe eingegangen. Anschließend werden probabilistische graphische Modelle umrissen. Der Schwerpunkt liegt dabei auf den ungerichteten Modellen, den sogenannten Markov-Netzen. Das Kapitel schließt mit der Einführung des Markov-Logik-Netzes (MLN) – ein probabilistisch-logischer Formalismus, welcher die beiden erstgenannten kombiniert.

Kapitel 3 erläutert zuerst den Begriff „Ontologie“, bevor es detailliert die im Rahmen der Arbeit entwickelte Ontologie einer Verkehrssituation und deren Formalisierung in Prädikatenlogik erster Stufe schildert. Anschließend werden mehrere auf der Ontologie basierende Markov-Logik-Modelle vorgestellt. Diese bestehen aus prädikatenlogischen Formeln, die sich in Axiome und Regeln unterteilen lassen. Die Axiome sind stets gültige Aussagen über den Diskursbereich. Regeln repräsentieren dagegen Wissen über den Diskursbereich, das lediglich typischerweise wahr ist. Jede Regel bekommt während einer anschließenden in Kapitel 6 beschriebenen Trainingsphase ein Gewicht zugeordnet, welches ein Maß für die Gültigkeit der durch diese Regel gemachte Aussage ist. Das Modell „Objektklassifikation“ (OK) modelliert die 3D-Objektklassen der Ontologie in Abhängigkeit von den einfachen Attributen. Das Modell „Relationale Objektklassifikation“ (ROK) ist eine Erweiterung des ersten Modells um die Konzepte der Objektrelationen, wie z. B. folgen oder entgegenkommen, und ihrer Beziehung zu den 3D-Objektklassen. Die beiden letzten Modelle „Inferierte Objektrelationen“ (IR) und „Direkte Objektrelationen“ (DR) zeigen zwei verschiedene Herangehensweisen hinsichtlich der Formulierung von Definitionen für die modellierten Objektrelationen auf. Im Modell IR werden die Zusammenhänge zwischen den Objektrelationen der Ontologie und der Objektattribute – relative Position, relativer Abstand und Differenz der Geschwindigkeitsrichtungen, mit Hilfe von Regeln formalisiert. Während der Trainingsphase dieser Regeln werden formale Definitionen der Objektrelationen und deren Eigenschaften gelernt. Das Modell DR gibt dagegen feste Definitionen für die Objektrelationen in Abhängigkeit von der relativen Position und der relativen Geschwindigkeitsrichtung beziehungsweise dem relativen Abstand vor.

Kapitel 4 schildert die qualitative Abstraktion der Objektattribute. Diese besteht aus geeigneter Diskretisierung der quantitativen Daten und anschließender Überführung auf die begriffliche Ebene. Die Objektattribute werden in zwei Gruppen

eingeteilt. Die *einfachen* Attribute beziehen sich auf ein Objekt, während die *relativen* Attribute sich auf zwei Objekte beziehen.

Kapitel 5 widmet sich den statistischen Gütekriterien, die zur quantitativen Auswertung der Inferenzergebnisse der entwickelten Modelle verwendet werden. Diese sind die Mehrklassen-Genauigkeit, die Fläche unter der ROC-Kurve für Mehrklassen-Probleme und die mittlere bedingte logarithmische Likelihood.

Kapitel 6 beschreibt die Trainingsdatenbank sowie das Parameterlernen der in Kapitel 3 umrissenen Markov-Logik-Modelle. Die zu den Regeln des Modells „Relative Objektklassifikation“ assoziierten Gewichte werden mit vier verschiedenen Parameterschätzalgorithmen berechnet. Die so gelernten Gewichte entsprechen dem Wahrheitsgehalt der jeweiligen Regel bezüglich den Trainingsdaten. Die Inferenzergebnisse der vier auf diese Weise bestimmten MLN werden anschließend untersucht. Basierend auf diesem Vergleich wird die Auswahl für eines der Parameterlernverfahren getroffen.

Kapitel 7 beinhaltet die Validierung der entwickelten Markov-Logik-Modelle zur Situationsbeschreibung anhand simulierter und experimentell gewonnener Videodaten.

Kapitel 8 fasst die wesentlichen Beiträge der Arbeit zusammen und gibt einen Ausblick auf mögliche weiterführende Forschungsarbeiten.

2 Theoretische Grundlagen

Dieses Kapitel gibt eine Einführung in die Grundlagen der Prädikatenlogik erster Stufe. Danach wird auf die Theorie der graphischen Modelle und insbesondere der ungerichteten Markov-Netze eingegangen. Schließlich wird das Markov-Logik-Netz eingeführt, das eine Kombination der ersten beiden Formalismen darstellt.

2.1 Prädikatenlogik erster Stufe

Prädikatenlogik der ersten Stufe (PL1) ist eine klassische Logik, die einen Rahmen liefert, um Aussagen über eine Domäne zu formalisieren und auf ihre Gültigkeit zu überprüfen [Sch07]. Die PL1 erweitert die Sprache der Aussagenlogik [Tha88]. Die Aussagenlogik geht davon aus, dass Fakten in der Welt gelten oder nicht gelten, also wahr oder falsch sind. Diese Sprache ist allerdings nicht genügend ausdrucksstark, um Wissen aus komplexen Umgebungen auf präzise Weise zu repräsentieren. Die PL1 geht davon aus, dass die Welt aus Objekten besteht, die in bestimmten Relationen zueinander stehen, die gelten oder nicht gelten. Die Sprache der PL1 ist ausdrucksstark genug, um einen Großteil unseres Allgemeinwissens zu repräsentieren. Die PL1 wird in diesem Kapitel formalisiert.

2.1.1 Syntax

Das Vokabular der PL1-Sprache enthält folgende Symbole: Individuenvariablen oder kurz Variablen, Funktionssymbole, Prädikatensymbole und Junktoren: Negations- (\neg), Konjunktions- (\wedge), Disjunktions- (\vee), Implikationssymbol (\Rightarrow), Symbol für beiderseitige Implikation (\Leftrightarrow), Allquantor (\forall) und Existenzquantor (\exists). Jedem Prädikaten- oder Funktionssymbol wird eine *Stelligkeit* zugeordnet, d. h. die Anzahl ihrer Argumente wird festgelegt. Ein nullstelliges Prädikatensymbol ist ein aussagenlogisches *Atom*. Ein nullstelliges Funktionssymbol heißt auch *Individuenkonstante* oder kurz *Konstante*. Variablen, Prädikaten- und Funktionssymbole werden durch Buchstaben oder Wörter repräsentiert.

Das Vokabular ermöglicht die Definition von *Termen*, *Atomen* und *Formeln* durch entsprechende Syntaxregeln.

- Ein *Term* ist ein logischer Ausdruck, der sich auf ein Objekt bezieht. Individuenkonstanten und Variablen sind Terme. Ein Term heißt *Grundterm*, wenn er keine Variablen enthält. Ein *komplexer Term* wendet Funktionssymbole auf Terme an, um ein Objekt zu benennen.
- Ein *Atom*, auch atomare Formel genannt, wendet Prädikatensymbole auf eine passende Menge von Termen an. Ist S ein Prädikatensymbol der Stelligkeit n und ist t_1, \dots, t_n eine aus n Termen bestehende Liste, so ist die Zeichenfolge $S(t_1, \dots, t_n)$ als atomare Formel oder Atom der Sprache definiert. Atome, die nur Grundterme enthalten werden *Grundatome* genannt.
- Eine *Formel* wird folgendermaßen induktiv definiert:
 - Jede atomare Formel ist eine Formel der Sprache.
 - Ist A eine Formel, so auch $\neg A$.
 - Sind A und B Formeln, so auch $(A \wedge B)$ sowie $(A \vee B)$; $(A \Rightarrow B)$ und $(A \Leftrightarrow B)$.
 - Ist A eine Formel und y eine Variable, so ist auch $\forall y A$ und $\exists y A$ jeweils eine Formel.

Die prädikatenlogische Sprache ist die Menge aller Formeln, die sich in der oben beschriebenen Weise aus dem Alphabet konstruieren lassen.

Variablen repräsentieren ein beliebiges Element der Domäne¹, ohne dieses explizit zu benennen. Innerhalb eines Ausdrucks bezeichnen Variablen gleichen Namens dasselbe Objekt. Man nennt eine Variable *gebunden* in einer Formel der Prädikatenlogik, wenn ihr in der Formel ein Quantor vorangeht. Ist eine Variable nicht gebunden, nennt man sie *frei*. Diese anschauliche Definition soll hier genügen, für eine präzisere Definition wird auf [Tha88] verwiesen. Für Terme gibt es keine Bindungen von Variablen, so dass jedes Auftreten einer Variable in einem Term als freies Auftreten angesehen wird. Eine Formel heißt *geschlossen*, wenn sie keine freien Variablen enthält [Sch07].

2.1.2 Semantik

Im Fall der Aussagenlogik ordnet eine semantische Interpretation jeder aussagenlogischen Konstante einen Wahrheitswert - wahr (\mathcal{W}) oder falsch (\mathcal{F}) - zu. Bei der Prädikatenlogik ist die Situation etwas komplizierter, da wir es mit Termen

¹Die Domäne, auch Diskursbereich oder Diskursuniversum genannt, stellt den darzustellenden Weltausschnitt dar [Gör03].

und Prädikatensymbolen zu tun haben. Eine *Interpretation*, auch Struktur genannt, der Prädikatenlogik muss somit einerseits den Termen bestimmte Objekte aus der Domäne zuordnen, andererseits muss sie den Prädikatensymbolen Relationen zuordnen, damit die Wahrheit oder Falschheit einer Formel bestimmt werden kann.

Eine *Interpretation* I ist ein Tripel (D, I_c, I_v) mit folgenden Eigenschaften:

- D ist eine nichtleere Menge, die sogenannte *Domäne der Interpretation*.
- I_c ist eine Abbildung, die:
 - jeder Konstante K ein Element $I_c(K) \in D$,
 - für $n \geq 1$: jedem n -stelligen Funktionssymbol f eine Funktion $I_c(f) : D^n \rightarrow D$,
 - jedem 0-stelligen Prädikatensymbol S einen Wahrheitswert $I_c(S) \in \{\mathcal{W}, \mathcal{F}\}$,
 - für $m \geq 1$: jedem m -stelligen Prädikatensymbol S eine m -stellige Relation $I_c(S) \subseteq D^m$ zuordnet.
- Die *Variablenbelegung* I_v ist eine Funktion, die jeder Variable y ein Element $I_v(y) \in D$ zuordnet.

Die zunächst sinnleeren Symbole der Sprache werden an konkrete Objekte gebunden - Namen bezeichnen, wie in einer natürlichen Sprache, Objekte oder Merkmale. Die Definitionen für die Junktoren in PL1 werden von der Aussagenlogik übernommen [Tha88].

Die Regeln für die Auswertung von Termen und Formeln mit Hilfe der Interpretation $I = (D, I_c, I_v)$ ordnen jedem Term t ein Element $I(t)$ in D und jeder Formel A einen Wahrheitswert $I(A)$ zu:

- Ist y eine freie Variable, so gilt $I(y) \Leftrightarrow I_v(y)$.
- Ist f ein n -stelliges Funktionssymbol und sind t_1, \dots, t_n Terme, so gilt $I(f(t_1, \dots, t_n)) \Leftrightarrow I_c(f)(I(t_1), \dots, I(t_n))$.
- Ist S ein m -stelliges Prädikatensymbol und sind t_1, \dots, t_m Terme, so gilt $I(S(t_1, \dots, t_m)) \Leftrightarrow I_c(S)(I(t_1), \dots, I(t_m))$.
- Sind q und t Terme, so hat $I(q = t)$ den Wert \mathcal{W} , wenn gilt $I(q) = I(t)$, und sonst den Wert \mathcal{F} .
- Sind A und B Formeln, so werden $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$ und $(A \Leftrightarrow B)$ wie in der Aussagenlogik interpretiert.

Diese Regeln genügen für die Interpretation von *Matrizen*, d. h. Formeln, die keine Quantoren enthalten. Sei z. B. $M(y_1, \dots, y_n)$ eine Formel mit Variablen y_1, \dots, y_n , dann bezeichnet $I(M)$ die funktionale Abbildung von jedem n -Tupel $(I(t_1), \dots, I(t_n))$ auf einen Wahrheitswert $I(M(t_1, \dots, t_n))$.

Bevor die Regeln für die Interpretation der Quantoren definiert werden, wird die folgende Notation eingeführt. Seien $I = (D_I, I_c, I_v)$ eine Interpretation, y eine Variable und d ein Objekt aus D_I , dann bezeichnet $I_{y/d}$ die Interpretation J , für die gilt $D_J = D_I$, $J_c = I_c$, $J_v(y) = d$ und $J_v(z) = I_v(z)$ für jede Variable $z \neq y$. Diesen Vorgang des Einsetzens eines Objektes anstelle einer Variable nennt man *Substitution*.

Sei A eine Formel und y eine Variable, dann gilt:

- $I(\forall y A)$ ist \mathcal{W} , wenn $I_{y/d}(A)$ ist \mathcal{W} für jedes Objekt $d \in D$,
- $I(\exists y A)$ ist \mathcal{W} , wenn $I_{y/d}(A)$ ist \mathcal{W} für mindestens ein Objekt $d \in D$.

Ist $I(A) = \mathcal{W}$, so *erfüllt* die Interpretation I die Formel A . Eine solche Interpretation wird *Modell* von A genannt [Tha88].

Prädikatenlogische Formeln lassen sich in drei Gruppen einteilen: *allgemeingültige* Formeln, die in allen Interpretationen wahr sind; *unerfüllbare* Formeln, die in allen Interpretationen falsch sind und *erfüllbare* Formeln, die in mindestens einer Interpretation wahr sind.

Anders als bei der Aussagenlogik ist es allerdings bei der PL1 nicht ohne weiteres möglich, die Allgemeingültigkeit einer Formel festzustellen [Tha88]. Die PL1 ist dennoch *vollständig*. Das heißt: es existieren Kalküle², mit denen man alle allgemeingültigen Formeln der Prädikatenlogik auflisten kann. Andererseits ist die PL1 *unentscheidbar*. Für eine gegebene prädikatenlogische Formel lässt sich nicht in jedem Fall in endlich vielen Schritten feststellen, ob diese allgemeingültig ist.

Nachdem das Konzept der Wahrheit definiert wurde, kann die *Folgerungsbeziehung* oder auch *logische Konsequenz* \models (engl.: entailment) eingeführt werden. Eine Formel A *folgt logisch* aus einer anderen Formel B genau dann, wenn für jede Interpretation I gilt: Wenn $I(B) = \mathcal{W}$, dann $I(A) = \mathcal{W}$ [Gör03]. Die Definition der logischen Konsequenz kann angewendet werden, um logische Inferenz auszuführen.

²Ein Kalkül beschreibt die Herleitbarkeit durch logische Axiome und Schlussregeln [Gör03].

2.1.3 Normalformen

Ein *Literal* ist eine Formel, die aus einem Atom oder einem negierten Atom besteht. Die Disjunktion einer endlichen Anzahl von Literalen heißt *Klausel*. Die leere Klausel ist die einzige unerfüllbare Klausel. *Konjunktive Normalform* (KNF) ist die Konjunktion einer endlichen Anzahl von Klauseln. Jede prädikatenlogische Formel kann in die KNF umgewandelt werden [Tha88]. Der Begriff einer Klausel ist wichtig in der Praxis. Die Beschreibung von Problemen und Algorithmen in Form von Klauseln hat zur Entstehung der Logischen Programmierung und insbesondere der logischen Programmiersprache Prolog geführt [Rus03].

Formeln in *Skolemform* sind geschlossene, universell quantifizierte Formeln: Nur Allquantoren kommen vor, und sie stehen sämtlich im Präfix zu Beginn der Formel. Eine Formel ist genau dann erfüllbar, wenn ihre äquivalente Klausel in Skolemform erfüllbar ist. Es existiert ein berechenbarer Test auf Erfüllbarkeit für solche Klauseln. Die *Klauselform* oder *Klauselnormalform* ist eine Skolemform, deren Matrix in KNF ist. Das Präfix wird normalerweise weggelassen, jede Variable wird implizit als universell quantifiziert angenommen. Klauselformen dienen zur maschinellen Beweisführung über logische Formeln. *Hornklauseln* stellen eine spezielle Klauselnormalform dar, bei der jede Klausel maximal ein positives Literal enthält. Bei einer Formel in Skolemform lässt sich wesentlich Genaueres über ihre Modelle aussagen als im allgemeinen Fall. Hierauf soll im Folgenden noch kurz eingegangen werden.

2.1.4 Herbrand-Interpretation

Herbrand-Interpretationen spielen eine Schlüsselrolle in der logischen Programmierung. Wie in Kapitel 2.1.2 bereits erwähnt, ist es bei der Prädikatenlogik erster Stufe nicht ohne weiteres möglich, die Allgemeingültigkeit, Erfüllbarkeit oder Unerfüllbarkeit einer Formel festzustellen. Dies ergibt sich aus der Unendlichkeit der möglichen Interpretationen prädikatenlogischer Formeln. Die Frage ist, ob es notwendig ist all diese Interpretationen zu berücksichtigen, um zu bestimmen, ob eine Formel konsistent ist oder nicht. Der Logiker Jacques Herbrand hat einen vereinfachten Konsistenztest entwickelt, dessen Grundidee im folgenden vorgestellt wird [Sch07]. Eine Klauselform einer Formel ist unerfüllbar genau dann, wenn diese in allen Interpretationen falsch ist. Da es aber unendlich viele mögliche Interpretationsdomänen gibt, wäre es interessant eine besondere Domäne zu haben, für die gälte, dass eine Klauselform einer Formel genau dann unerfüllbar ist, wenn diese falsch in allen Interpretationen *dieser Domäne* ist. So eine Domäne wird *Herbrand-Universum* genannt [Tha88].

Definition 2.1.1 Ein *Herbrand-Universum* D_A wird wie folgt induktiv definiert:

- Alle in A vorkommenden Konstanten sind in D_A . Falls A keine Konstante enthält, so ist $a \in D_A$.
- Für jedes in A vorkommende n -stellige Funktionssymbol f und Terme $t_1, \dots, t_n \in D_A$ ist der Term $f(t_1, \dots, t_n) \in D_A$.

Das Herbrand-Universum D_A einer geschlossenen Formel A in Skolemform ist demnach die Menge aller Grundatome, die aus den Bestandteilen von A gebildet werden können. Falls in A keine Konstante vorkommt, wird zunächst eine beliebige Konstante gewählt und dann die variablenfreien Terme gebildet.

Eine *Herbrand-Interpretation* oder auch *Herbrand-Algebra*, oder *mögliche Welt* genannt, wird im folgenden definiert [Tha88].

Definition 2.1.2 *Herbrand-Interpretation* einer Menge von geschlossenen Klauseln in Skolem-Normalform ist eine Interpretation, die folgende Bedingungen erfüllt:

- Die Domäne der Interpretation ist das Herbrand-Universum.
- $I(f)(t_1, \dots, t_n) = f(t_1, \dots, t_n)$ für alle n -stelligen Funktionssymbole f und beliebige Grundterme t_1, \dots, t_n . Im Falle $n = 0$ heißt das $I(K) = K$.

Für eine Formelmenge M ist ein *Herbrand-Modell* von M eine Herbrand-Interpretation, welche Modell von M ist. In einer Herbrand-Interpretation wird also jeder Grundterm als er selbst interpretiert. Das bedeutet $I(t) = t$ für Grundterme und gilt insbesondere für Konstanten K .

Definition 2.1.3 Sei $\forall y_1 \dots \forall y_n A$ mit quantorenfreiem A eine geschlossene Formel. Eine **Grundinstanz dieser Formel** ist eine Formel $\{y_1/t_1, \dots, y_n/t_n\}(A)$ mit t_1, \dots, t_n beliebigen Grundtermen.

Ist M eine Menge geschlossener, universell quantifizierter Formeln, so sei M_G die Menge aller Grundinstanzen von Formeln in M . Der Satz von Herbrand lautet [Sch07]:

Theorem 2.1.1 Sei M eine Menge geschlossener, universell quantifizierter Formeln, wobei keine der Formeln das Gleichheitssymbol enthält. Dann sind äquivalente Aussagen:

1. M hat ein Modell.

2. M hat ein Herbrand-Modell.
3. M_G hat ein Modell.
4. M_G hat ein Herbrand-Modell.

Folglich ist eine Menge von Klauseln unerfüllbar genau dann, wenn diese in allen Herbrand-Interpretationen unerfüllbar ist. Mit anderen Worten werden mit Hilfe der Herbrand-Interpretationen die Unerfüllbarkeit einer Menge von Klauseln auf die Unerfüllbarkeit der Menge aller Grundinstanzen der Klauseln im Herbrand-Universum zurückgeführt. Die Herbrand-Interpretationen können als aussagenlogische Interpretationen betrachtet werden und somit können Methoden des automatischen Schließens angewendet werden.

Zusammenfassend kann über die Wahrheit einer logischen Formel erst dann geschlussfolgert werden, nachdem eine Interpretation und eine Variablenbelegung definiert wurden [Sch07]. Die Interpretation spezifiziert die Zuordnung der Formelkomponenten zu Objekten, Funktionen und Relationen der Domäne. Die Variablenbelegung ordnet allen Variablen Konstanten zu.

2.1.5 Inferenz

Eine prädikatenlogische Wissensbasis besteht aus einer Menge von Formeln in der Sprache der Prädikatenlogik erster Stufe. Die Formeln einer Wissensbasis sind implizit konjunktiv verknüpft, so dass diese als eine große Formel betrachtet werden kann. Mittels Propositionalisierung wird eine PL1-Wissensbasis durch eine ihr äquivalente aussagenlogische Wissensbasis ersetzt. In endlichen Domänen wird dafür jede allquantisierte (existenzquantisierte) Formel durch die Konjunktion (Disjunktion) all ihrer Grundinstanzen ersetzt. Eine PL1-Wissensbasis ist erfüllbar, genau dann wenn die äquivalente aussagenlogische Wissensbasis erfüllbar ist. Deswegen besteht die Inferenz über eine PL1-Wissensbasis aus Propositionalisierung gefolgt durch Überprüfung der Erfüllbarkeit. Letzteres Problem ist NP-vollständig. Eine Aussage ist erfüllbar, wenn es eine Belegung (Interpretation, Bewertung) der Variablen gibt, für die der Wahrheitswert des gesamten Ausdrucks wahr ist.

Für die Inferenz in einer PL1-Wissensbasis in Klauselform können das Resolutionskalkül oder effiziente Erfüllbarkeitsalgorithmen (die sogenannten SAT-Solver) verwendet werden [Tha88]. Der Algorithmus WalkSAT [Sel95] führt eine stochastische lokale Suche durch, indem er bei jedem Durchgang eine nicht erfüllte Klausel auswählt und den spezifizierten Wert eines ihrer Grundatome wechselt. Er wählt zufällig zwischen zwei Methoden, wie das zu wechselnde Atom ausgewählt wird. Mit Wahrscheinlichkeit p wählt der Algorithmus das Grundatom aus,

welches die Anzahl der erfüllten Klauseln maximiert, und mit Wahrscheinlichkeit $1 - p$ ein zufälliges Grundatom. Nach n Wechselgängen startet der WalkSAT-Algorithmus mit einem neuen zufällig ausgewählten Initialzustand neu. Das ganze wird m -fach wiederholt. Die Kombination der Auswahlkriterien verhindert die Konvergenz in einem lokalen Optimum. Für weiterführende Literatur zum Thema automatisches Beweisen wird auf [Tha88] verwiesen.

2.2 Probabilistische Graphische Modelle

Probabilistische graphische Modelle ermöglichen eine kompakte Repräsentation der Verbundverteilung einer Menge von Zufallsvariablen [Rus03]. Jede diskrete Verbundverteilung kann durch eine Auflistung der Wahrscheinlichkeiten aller möglichen Kombinationen der Belegung der Zufallsvariablen beschrieben werden, angenommen jede Zufallsvariable hat endlich viele Zustände. Häufig hängt eine Zufallsvariable allerdings nur von einer kleinen Untermenge aller Variablen ab. Daher würden in einer kompletten Auflistung der Wahrscheinlichkeiten aller möglichen Kombinationen von Variablenbelegungen viele der Parameter den gleichen Wert haben. Graphische Modelle machen sich diese Beobachtung zunutze und vermeiden solche Redundanzen, indem sie bedingte Unabhängigkeiten in der Domäne explizit modellieren. Dabei werden Zufallsvariablen durch Knoten eines Graphen und Abhängigkeiten zwischen Variablen als Kanten repräsentiert. Wahrscheinlichkeiten werden anhand einer Menge von Funktionen berechnet, die über den Graphen festgelegt werden.

Ein Bayes'sches Netz, zum Beispiel, repräsentiert einen gerichteten azyklischen Graphen, dessen Kanten kausale Abhängigkeiten abbilden [Rus03]. Die Verbundverteilung eines solchen Graphen wird mit Hilfe von bedingten Wahrscheinlichkeitsverteilungen berechnet. Pro Knoten wird eine bedingte Wahrscheinlichkeitsverteilung der durch ihn repräsentierten Zufallsvariable definiert. Diese gibt die Wahrscheinlichkeit an, mit der die zugehörige Zufallsvariable einen bestimmten Zustand bei gegebener Belegung der Elternknoten annimmt.

Ein anderer weit verbreiteter Typ graphischer Modelle sind die Markov-Netze, die im Unterschied zu den Bayes'schen Netzen, durch ungerichtete Graphen repräsentiert werden. Im Folgenden wird die Theorie der Markov-Netze im Detail beschrieben, weil diese Grundlage zum Verständnis der vorliegenden Arbeit sind. Die Ausführungen hierzu orientieren sich an [Pea06].

2.2.1 Bedingte Unabhängigkeit

Sei $\mathbf{X} = (X_1, X_2, \dots, X_n) \in \mathcal{X}$ ein *Zufallsfeld*, d. h. eine endliche Menge diskreter Zufallsvariablen. Dabei ist mit \mathcal{X} der Stichprobenraum dieses Zufallsfeldes bezeichnet, der als abzählbar (endlich oder unendlich) angenommen wird. Eine Realisation der Menge von Zufallsvariablen wird durch kleine Buchstaben, wie z. B. $\mathbf{x} = (x_1, x_2, \dots, x_n)$, gegeben. Die Schreibweise $X_i = x_i$ bedeutet, dass X_i den Zustand x_i annimmt. Die Wahrscheinlichkeit des Zufallsfeldes wird durch $P(\mathbf{X} = \mathbf{x})$ bezeichnet.

Definition 2.2.1 Seien \mathbf{A} , \mathbf{B} und \mathbf{C} paarweise disjunkte Teilmengen von Variablen aus \mathbf{X} . Die Teilmenge \mathbf{A} ist bedingt unabhängig von \mathbf{B} gegeben \mathbf{C} , wenn

$$P(\mathbf{A} = \mathbf{a} | \mathbf{B} = \mathbf{b}, \mathbf{C} = \mathbf{c}) = P(\mathbf{A} = \mathbf{a} | \mathbf{C} = \mathbf{c}) \quad (2.1)$$

und $P(\mathbf{B} = \mathbf{b}, \mathbf{C} = \mathbf{c}) > 0$ gilt.

Die ternäre Relation $U(\mathbf{A}, \mathbf{C}, \mathbf{B})$ wird eingeführt, um die bedingte Unabhängigkeit von \mathbf{A} und \mathbf{B} gegeben \mathbf{C} in P zu bezeichnen. Diese muss folgende Unabhängigkeitsaxiome erfüllen:

$$\text{Symmetrie: } U(\mathbf{A}, \mathbf{C}, \mathbf{B}) \Leftrightarrow U(\mathbf{B}, \mathbf{C}, \mathbf{A}) \quad (2.2)$$

$$\text{Zerlegung: } U(\mathbf{A}, \mathbf{C}, \mathbf{B} \cup \mathbf{D}) \Rightarrow U(\mathbf{A}, \mathbf{C}, \mathbf{B}) \wedge U(\mathbf{A}, \mathbf{C}, \mathbf{D}) \quad (2.3)$$

$$\text{Schwache Vereinigung: } U(\mathbf{A}, \mathbf{C}, \mathbf{B} \cup \mathbf{D}) \Rightarrow U(\mathbf{A}, \mathbf{C} \cup \mathbf{D}, \mathbf{B}) \quad (2.4)$$

$$\text{Zusammenziehung: } U(\mathbf{A}, \mathbf{C}, \mathbf{B}) \wedge U(\mathbf{A}, \mathbf{C} \cup \mathbf{B}, \mathbf{D}) \Rightarrow U(\mathbf{A}, \mathbf{C}, \mathbf{B} \cup \mathbf{D}). \quad (2.5)$$

Wenn $P > 0$ gilt außerdem:

Schnitt:

$$U(\mathbf{A}, \mathbf{C} \cup \mathbf{D}, \mathbf{B}) \wedge U(\mathbf{A}, \mathbf{C} \cup \mathbf{B}, \mathbf{D}) \Rightarrow U(\mathbf{A}, \mathbf{C}, \mathbf{B} \cup \mathbf{D}). \quad (2.6)$$

Das Symbol \cup bedeutet in diesem Kontext Vereinigung von Mengen und sollte nicht mit logischer Disjunktion verwechselt werden.

Diese Axiome können als qualitative Inferenzregeln aufgefasst werden, aus denen neue Unabhängigkeitsbeziehungen aus bereits bekannten bedingten Unabhängigkeiten abgeleitet werden können. Dies hat sich für die Untersuchung bedingter Unabhängigkeiten als einfacher erwiesen, im Vergleich zu einer wiederholten Prüfung von Definition 2.2.1. Jede Relation $U(\cdot)$, die die Gl. (2.2) - (2.5) erfüllt, wird Teilgraphoid (engl. *semi-graphoid*) genannt. Erfüllt sie auch Gl. 2.6, dann wird sie Graphoid (engl. *graphoid*) genannt [Pea06].

Gerichtete und ungerichtete Graphen erfüllen die oben definierten Axiome. Diese Tatsache lässt vermuten, dass alle bedingten Unabhängigkeitsaussagen, die in einer Wahrscheinlichkeitsverteilung gelten, durch einen Graphen darstellbar sind, d. h. dass anhand der Graphenstruktur abzulesen ist, ob zwei Zufallsvariablen bedingt unabhängig gegeben eine dritte sind. Dies ist allerdings nicht der Fall, da die graphische Trennung ausdrucksstärker ist. Für mehr Information dazu wird auf [Gör03] verwiesen. Im Folgenden werden daher nur diejenigen Wahrscheinlichkeitsverteilungen charakterisiert, deren bedingte Unabhängigkeiten durch Graphen repräsentiert werden können.

2.2.2 Markov-Netze

Ein ungerichteter Graph $G = (\mathbf{V}, \mathbf{E})$ besteht aus einer Menge von Knoten \mathbf{V} und einer Menge von Kanten \mathbf{E} , welche Knoten aus \mathbf{V} verbinden. Seien \mathbf{A} , \mathbf{B} und \mathbf{C} paarweise disjunkte Teilmengen von Knoten in G . Die Bezeichnung $\langle \mathbf{A} | \mathbf{C} | \mathbf{B} \rangle_G$ bedeutet, dass \mathbf{A} und \mathbf{B} durch \mathbf{C} getrennt werden, d. h. jeder Pfad zwischen einem Knoten aus \mathbf{A} und einem Knoten aus \mathbf{B} mindestens einen Knoten aus \mathbf{C} enthält.

Definition 2.2.2 *Ein ungerichteter Graph $G = (\mathbf{V}, \mathbf{E})$ heißt ein bedingter Abhängigkeitsgraph (engl.: dependency map, D-map) einer Verteilung $P(\mathbf{X})$, wenn es eine eins-zu-eins Korrespondenz zwischen den Elementen des Zufallsfeldes \mathbf{X} und den Knoten \mathbf{V} gibt, so dass für alle paarweise disjunkte Teilmengen \mathbf{A} , \mathbf{B} und \mathbf{C} gilt:*

$$U(\mathbf{A}, \mathbf{C}, \mathbf{B})_P \implies \langle \mathbf{A} | \mathbf{C} | \mathbf{B} \rangle_G. \quad (2.7)$$

Ähnlich heißt G ein bedingter Unabhängigkeitsgraph (engl.: independency map, I-map) von $P(\mathbf{X})$, wenn gilt:

$$U(\mathbf{A}, \mathbf{C}, \mathbf{B})_P \iff \langle \mathbf{A} | \mathbf{C} | \mathbf{B} \rangle_G. \quad (2.8)$$

G heißt perfekter Graph (engl.: perfect map) von $P(\mathbf{X})$, wenn es sowohl bedingter Abhängigkeits- als auch Unabhängigkeitsgraph von $P(\mathbf{X})$ ist.

Ein Abhängigkeitsgraph garantiert, dass alle Knoten, die verknüpft sind, voneinander abhängig in $P(\mathbf{X})$ sind. Es kann allerdings sein, dass zwei abhängige Variablen als nicht verknüpfte Knoten dargestellt werden. Ein Unabhängigkeitsgraph garantiert, dass nicht verknüpfte Knoten voneinander unabhängigen Variablen entsprechen, nicht aber, dass alle verknüpfte Knoten abhängige Variablen darstellen. Leere Graphen sind triviale Abhängigkeitsgraphen, voll verknüpfte Graphen sind triviale Unabhängigkeitsgraphen.

Definition 2.2.3 (Markov-Netz) Ein ungerichteter Graph G heißt minimaler bedingter Unabhängigkeitsgraph (engl.: minimal I-map) oder Markov-Netz zu P , wenn G ein bedingter Unabhängigkeitsgraph zu P ist, der keine überflüssigen Kanten enthält.

Jede strikt positive Wahrscheinlichkeitsverteilung P , für die die Eigenschaften Symmetrie, Zerlegung und Schnitt gelten, hat ein eindeutig bestimmtes Markov-Netz $G_0 = (\mathbf{X}, \mathbf{E}_0)$, das man konstruieren kann, indem alle Kanten (X_i, X_j) aus dem vollständigen Graphen G auf \mathbf{X} entfernt werden, für die gilt:

$$(X_i, X_j) \notin \mathbf{E}_0 \Leftrightarrow U(X_i, \mathbf{X} - X_i - X_j, X_j)_P. \quad (2.9)$$

Dies wird paarweise Markov-Eigenschaft genannt.

Definition 2.2.4 Markov-Decke (engl.: Markov-Blanket) $\mathbf{B}_U(X_i)$ der Zufallsvariable $X_i \in \mathbf{X}$ heißt jede Variablenmenge $\mathbf{Q} \in \mathbf{X}$ für die gilt

$$U(X_i, \mathbf{Q}, \mathbf{X} - \mathbf{Q} - X_i) \text{ mit } X_i \notin \mathbf{Q}. \quad (2.10)$$

Ein Markov-Rand (engl.: Markov-Boundary) $\mathbf{R}_U(X_i)$ von X_i ist eine minimale Markov-Decke.

Für strikt positive Verteilungen besitzt jedes Element $X_i \in \mathbf{X}$ einen Markov-Rand, der gerade aus seinen Nachbarknoten besteht (lokale Markov-Eigenschaft).

Korrolar 2.2.1 Für strikt positive Verteilungen bildet die Menge der Markov-Ränder $\mathbf{R}_U(X_i)$ ein Nachbarschaftssystem $\mathbf{R}_U = \{\mathbf{R}_U(X_i) | X_i \in \mathbf{X}\}$ auf \mathbf{X} , so dass für alle Paare $X_i, X_j \in \mathbf{X}$ gilt

1. $X_i \notin \mathbf{R}_U(X_i)$
2. $X_i \in \mathbf{R}_U(X_j) \iff X_j \in \mathbf{R}_U(X_i)$.

Die Spezifizierung eines Markov-Netzes durch bedingte Unabhängigkeiten, wie hier vorgestellt, stellt eine mögliche Herangehensweise dar. Alternativ kann ein Markov-Netz durch das Vorgeben einer gemeinsamen Verteilung der Komponenten, aus welcher bedingte Wahrscheinlichkeiten errechnet werden können, charakterisiert werden [Rus03].

Die Struktur eines probabilistischen Graphen bestimmt die qualitativen Eigenschaften der Verteilung. Erst eine quantitative Parametrisierung der Verbindungsstärke der Kanten ermöglicht die Inferenz anhand eines Graphen. Das Gibbs-

Potential ermöglicht die Spezifikation eines vollständigen und konsistenten quantitativen Modells, welches die Abhängigkeiten eines beliebigen Graphen repräsentiert [Pea06]. Dabei wird die Verbundwahrscheinlichkeitsverteilung als ein normiertes Produkt von positiven Potentialfunktionen über die Variablenbelegung der Cliques des Graphen berechnet. Eine Clique wird folgenderweise definiert:

Definition 2.2.5 *Eine nichtleere Teilmenge von \mathbf{X} wird Clique C bezüglich des Nachbarschaftssystems \mathbf{R}_U auf \mathbf{X} genannt, falls für alle Paare $X_i, X_j \in C$ gilt $X_i \neq X_j \Rightarrow X_j \in \mathbf{R}_U(X_i)$, d. h. alle Paare aus der Clique benachbart sind.*

Somit ist eine Clique ein vollständiger, alle möglichen Kanten enthaltender Teilgraph. Eine Clique heißt maximal, wenn sie nicht in einer anderen Clique enthalten ist. Sei $\mathbf{C}_M = (C_1, \dots, C_m)$ die Menge aller maximalen Cliques von G . Jeder Clique C_i aus \mathbf{C}_M wird ein sogenanntes Cliquespotential $\psi_i(\mathbf{x}_{C_i})$ zugeordnet. Der Wert des Potentials $\psi_i(\mathbf{x}_{C_i})$ entspricht der Güte der Variablenbelegung \mathbf{x}_{C_i} der i -ten Clique. Je besser die Belegung, desto höher das Potential. Jede nicht negative reellwertige Funktion kann als Potentialfunktion definiert werden.

Definition 2.2.6 *Eine Wahrscheinlichkeitsverteilung P über das Zufallsfeld \mathbf{X} heißt zerlegbar oder faktorisiert bezüglich eines ungerichteten Graphen $G = (\mathbf{X}, \mathbf{E})$ genau dann, wenn sie als Produkt von nichtnegativen Funktionen auf den maximalen Cliques \mathbf{C}_M von G geschrieben werden kann, so dass:*

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_i \psi_i(\mathbf{x}_{C_i}) . \quad (2.11)$$

Die Normalisierungskonstante Z bezeichnet die Summe über alle möglichen Variablenbelegungen (Zustände) und stellt somit sicher, dass $P \in [0,1]$:

$$Z = \sum_{\mathbf{x} \in \mathcal{X}} \prod_i \psi_i(\mathbf{x}_{C_i}) . \quad (2.12)$$

Eine bezüglich G faktorisierte Verteilung P wird auch Gibbs-Verteilung bezüglich G genannt [Get07b]. Für diese gilt folgendes Theorem, dessen Beweis in [Pea06] zu finden ist:

Theorem 2.2.2 (Hammersley-Clifford Theorem) *Eine strikt positive Verteilung P ist genau dann faktorisiert bezüglich eines ungerichteten Graphen G , wenn G ein Markov-Netz zu P ist.*

Wird als Potentialfunktion $\psi_i(\mathbf{x}_{C_i}) = \exp(w_i f_i(\mathbf{x}_{C_i}))$ ausgewählt, so bekommt man folgendes logarithmisch-lineares Modell:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp \left(\sum_i w_i f_i(\mathbf{x}_{C_i}) \right), \quad Z = \sum_{\mathbf{x} \in \mathcal{X}} \exp \left(\sum_i w_i f_i(\mathbf{x}_{C_i}) \right). \quad (2.13)$$

Dabei ist das sogenannte Feature $f_i(\mathbf{x}_{C_i})$ eine beliebige reellwertige Funktion über die Variablenbelegung \mathbf{x}_{C_i} der i -ten Clique und w_i ein zur i -ten Clique gehörendes reellwertiges Gewicht. Diese Repräsentation wächst exponentiell mit der Größe der Cliques.

2.2.3 Inferenz

Eine Standardinferenzaufgabe in graphischen Modellen ist die Maximum a-posteriori (MAP) Inferenz. Bei dieser wird nach dem wahrscheinlichsten Zustand \mathbf{y} einer Menge von Anfragevariablen \mathbf{Y} gegeben die Evidenzvariablen $\mathbf{E} = \mathbf{e}$ gesucht: $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{e})$. Ein Spezialfall der MAP-Inferenz ist die MPE-Inferenz (*most probable explanation*), bei der nach dem wahrscheinlichsten Zustand aller Variablen, die nicht zu den Evidenzvariablen gehören, gesucht wird. Eine weitere Standardinferenzaufgabe ist die bedingte Inferenz. Bedingte Inferenz in graphischen Modellen bezeichnet die Berechnung der bedingten Wahrscheinlichkeitsverteilung der Anfragevariablen gegeben einer Zuweisung der Evidenzvariablen $P(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \frac{P(\mathbf{Y}, \mathbf{e})}{P(\mathbf{e})}$. Inferenz in graphischen Modellen ist NP-schwer. Es gibt effiziente Algorithmen für exakte Inferenz in baumartigen Markov-Netzen, wie zum Beispiel der Viterbi-Algorithmus [Rab89] oder *belief propagation* [Get07b]. Im allgemeinen Fall kann zur Inferenz ein approximierendes probabilistisches MCMC-Verfahren (*Markov chain Monte Carlo*), wie etwa der Gibbs-Sampler [Rus03] angewendet werden. Der Gibbs-Sampler durchwandert zufällig den Zustandsraum - den Raum möglicher vollständiger Zuweisungen des Markov-Netzes - indem er jeweils den Wert einer der Nichtevidenzvariablen, bedingt auf den aktuellen Werten der Variablen ihrer Markov-Decke, austauscht. Die entstehende Folge von Stichprobenvektoren bildet eine Markov-Kette, deren stationäre Verteilung die gesuchte gemeinsame Wahrscheinlichkeitsverteilung ist. Jeder während des Samplings besuchte gemeinsame Zustand ist eine Stichprobe (Sample), die zur Schätzung der Abfragevariable(n) beiträgt. Somit ist die Genauigkeit der MCMC-Verfahren von der Anzahl der Samples abhängig. Damit MCMC konsistente Schätzungen für bedingte Wahrscheinlichkeiten zurückgibt wird gefordert, dass die definierte Markov-Kette die Eigenschaften Ergodizität und detaillierter

Ausgleich besitzt. Eine Markov-Kette ist ergodisch, wenn jeder Zustand aperiodisch von jedem anderen Zustand aus erreichbar ist. Die Eigenschaft detaillierter Ausgleich bedeutet, dass der erwartete Übergang zwischen einem Zustandspaar in beide Richtungen gleich ist. Die Markov-Kette des Gibbs-Sampler Algorithmus besitzt diese zwei Eigenschaften, wenn alle Cliquespotentiale des Markov-Netzes streng positiv sind [Get07b].

2.2.4 Lernen

Bei graphischen Modellen gibt es zwei Lernaufgaben: Parameterlernen und Strukturlernen. Während des Parameterlernens werden die Parameter des Modells, d. h. bei Markov-Netzen die Cliquepotentiale, geschätzt, wobei die qualitative Topologie des Netzes bekannt ist. Beim Strukturlernen werden sowohl die Parameter als auch die Struktur des graphischen Modells geschätzt. Die Parameter eines Markov-Netzes können durch Gradientenabstiegs- [DP97] oder Quasi-Newton-Optimierungsverfahren [Liu89] gelernt werden. Algorithmen zum Strukturlernen für Markov-Netze sind beispielsweise in [Lee06] und [Bro09] zu finden.

2.3 Markov-Logik-Netz

Ein Markov-Logik-Netz (MLN) ist eine probabilistisch logische Repräsentation, welche Prädikatenlogik erster Stufe (Kapitel 2.1) und Markov-Netze (Kapitel 2.2) miteinander verknüpft [Dom06]. Ein MLN ermöglicht es, die harten Einschränkungen auf die Menge der möglichen Welten aus der klassischen Prädikatenlogik abzuschwächen. Es wird eine Wahrscheinlichkeit für die möglichen Welten in Abhängigkeit von der Anzahl der wahren Grundinstanzen der Formeln der prädikatenlogischen Wissensbasis bestimmt. Eine Einführung in Markov-Logik, die an [Dom06] angelehnt ist, liefert das Kapitel 2.3.1. Verfahren zum Parameterlernen und zur Inferenz in MLN, die in der frei verfügbare Software Alchemy [Kok06] implementiert sind, stellen jeweils Kapitel 2.3.2 und Kapitel 2.3.3 vor.

2.3.1 Markov-Logik

Ein MLN besteht aus einer Menge prädikatenlogischer Formeln, die eine bestimmte Domäne beschreiben, sowie einem jeder Formel zugeordneten Gewicht. Das Gewicht quantifiziert den Wahrheitsgehalt der durch die Formel ausgedrückten

Behauptung. Ein MLN kann als ein Formalismus zur Erstellung eines Markov-Netzes mit logischen Variablen verstanden werden, d.h. zur Zerlegung einer Markov'schen Verbundverteilung binärer Zufallsvariablen in Faktoren, die jeweils nur von wenigen Variablen aus \mathbf{X} abhängen. Eine so definierte Verbundverteilung $P(\mathbf{X} = \mathbf{x})$ erlaubt die Anwendung klassischer probabilistischer Inferenzalgorithmen.

Ein Markov-Logik-Netz wird wie folgt definiert [Dom06]:

Definition 2.3.1 (Markov-Logik-Netz (MLN)) *Ein Markov-Logik-Netz (MLN) \mathbf{L} ist eine Menge von Paaren*

$$\mathbf{L} = \{(w_1, F_1), \dots, (w_n, F_n)\},$$

wobei F_i eine Formel in Prädikatenlogik erster Stufe und $w_i \in \mathbb{R}$ eine reelle Zahl repräsentieren. Gegeben das MLN \mathbf{L} und eine endliche Menge von Konstanten \mathbf{K} , kann ein Markov-Netz $G_{\mathbf{L},\mathbf{K}}$ folgenderweise definiert werden:

1. $G_{\mathbf{L},\mathbf{K}}$ besitzt einen binären Knoten für jedes mögliche Grundatom aus \mathbf{L} . Der Wert eines Knotens ist 1, wenn das zu Grunde liegende Grundatom wahr ist und sonst 0.
2. $G_{\mathbf{L},\mathbf{K}}$ besitzt ein Feature f_i für jede mögliche Grundinstanz jeder Formel F_i aus \mathbf{L} . Der Wert eines Features ist 1, wenn die dazugehörige Grundinstanz der Formel wahr ist, und sonst 0. Jedes Feature f_i ist durch die mit F_i assoziierte Zahl w_i gewichtet.

Für die Formeln in einem MLN darf die Standardsyntax von PL1 verwendet werden. Freie Variablen werden als allquantifiziert im Präfix der Formel behandelt.

Durch die Substitution der logischen Konstanten $K_i \in \mathbf{K}$ anstelle der Variablen in den Formeln von \mathbf{L} werden alle möglichen Grundatome $\mathbf{X} = (X_1, \dots, X_q)$ und daraus alle möglichen Grundinstanzen aller Formeln $\mathbf{F}_G = (F_{G_1}, \dots, F_{G_p})$ aus \mathbf{L} gebildet. Dabei bekommt jedes Element aus \mathbf{F}_G das Gewicht der zu Grunde liegenden Formel aus \mathbf{L} zugeordnet. Alle Elemente X_i können auf einen Wahrheitswert x_i mit

$$(X_1 = x_1, \dots, X_q = x_q), x_i \in \{0,1\}, i = 1, \dots, q$$

und alle Elemente F_{G_k} auf ein Feature $f_k(\mathbf{x}_{F_{G_k}})$ mit

$$(F_{G_1} = f_1(\mathbf{x}_{F_{G_1}}), \dots, F_{G_p} = f_p(\mathbf{x}_{F_{G_p}})), f_k(\mathbf{x}_{F_{G_k}}) \in \{0,1\}, k = 1, \dots, p$$

abgebildet werden. Zur Erstellung eines Markov-Netzes $G_{\mathbf{L},\mathbf{K}}(\mathbf{X}, \mathbf{F}_G)$ werden die Grundatome X_1, \dots, X_q als Elemente eines binären Zufallsfeldes betrachtet, die

die Knoten eines Interaktionsgraphen bilden. Der Wert eines Knotens x_i ist genau dann 1, wenn das zu Grunde liegende Grundatom wahr ist. Es existiert genau dann eine ungerichtete Kante zwischen zwei Knoten aus $G_{\mathbf{L},\mathbf{K}}$, wenn beide Grundatome mindestens in einem Element aus \mathbf{F}_G gemeinsam vorkommen. In der Terminologie eines Markov-Feldes sind solche Knoten benachbart und die Menge aller Nachbarschaftsbeziehungen bilden ein gültiges Nachbarschaftssystem. Somit bilden die Grundatome jeder Grundinstanz einer Formel eine nicht unbedingt maximale Clique (siehe Def. 2.2.5) von $G_{\mathbf{L},\mathbf{K}}$. Die Variablenbelegung der k -ten Clique wird durch das Feature $f_k(\mathbf{x}_{F_{G_k}})$ der entsprechenden Grundinstanz der Formel aus \mathbf{F}_G und durch das ihr zugeordnete Gewicht w_k bewertet. Das Feature hängt nur von den Variablen in $\mathbf{x}_{F_{G_k}}$ ab, die zur betrachteten Clique gehören. Ein Feature $f_k(\mathbf{x}_{F_{G_k}})$ ist genau dann 1, wenn $\mathbf{x}_{F_{G_k}}$ eine mögliche Welt für F_{G_k} darstellt, d.h. wenn die Aussage von F_{G_k} über den Diskursbereich wahr ist. Als Verbundwahrscheinlichkeit möglicher Welten \mathbf{x} , die das resultierende Markov-Netz beschreibt, wird schließlich

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp \left(\sum_k w_k f_k(\mathbf{x}_{F_{G_k}}) \right), \quad (2.14)$$

definiert, wobei Z ein reeller Normierungsfaktor ist (vgl. Gl. 2.13) und die Summation über alle Grundinstanzen aller Formeln erfolgt. Das resultierende Markov-Netz wird als *belegtes* Markov-Netz bezeichnet.

Jede Realisierung des belegten MLNs $G_{\mathbf{L},\mathbf{K}}$ stellt eine mögliche Welt dar. Eine mögliche Welt, oder auch Herbrand-Interpretation (siehe Def. 2.1.2) genannt, ordnet jedem möglichen Grundatom einen Wahrheitswert zu. Folgende drei Annahmen gewährleisten, dass die Menge der möglichen Welten von (\mathbf{L}, \mathbf{K}) endlich ist und das belegte Markov-Netz $G_{\mathbf{L},\mathbf{K}}$ eine eindeutige, wohldefinierte Wahrscheinlichkeitsverteilung über diese Welten unabhängig von der Interpretation und der Domäne beschreibt [Dom06]. Die erste Annahme sichert eine eindeutige Namenszuweisung: Unterschiedliche Konstanten benennen unterschiedliche Objekte. Die zweite Annahme besagt, dass alle Objekte der Domäne sich durch Konstanten und Funktionen aus (\mathbf{L}, \mathbf{K}) repräsentieren lassen. Die dritte Annahme setzt bekannte Funktionen voraus, d. h. der Wert jeder in \mathbf{L} enthaltenen Funktion ist für jedes mögliche Tupel von Argumenten bekannt und ist ein Element in \mathbf{K} . Diese Annahme ermöglicht das Ersetzen der Funktionen durch ihre bekannten Werte bei der Bildung der Grundinstanzen von Formeln. In Folge dessen enthalten die Grundatome, die berücksichtigt werden müssen, nur Konstanten als Argumente. Das zugehörige unendliche Herbrand-Universum kann ignoriert werden, da jeder der Terme einer bekannten Konstante entspricht und Atome, die verschachtelte Terme enthalten, durch Atome repräsentiert werden, die die entsprechenden Konstanten

enthalten. Die möglichen Grundatome werden deshalb lediglich durch das Substituieren jeder Variable durch Konstanten aus \mathbf{K} und das Ersetzen jedes komplexen Terms in einem Atom durch die entsprechende Konstante gebildet. Diese drei Annahmen sind in den meisten praktischen Fällen sinnvoll und vereinfachen deutlich die Anwendung von MLN. Sie werden in dieser Arbeit als geltend angesehen. Eine Diskussion der Möglichkeiten diese Annahmen zu umgehen ist in [Dom09] zu finden.

Tabelle 2.1 zeigt beispielhaft ein aus zwei Formeln bestehendes MLN \mathbf{L}_1 , welches für eine gegebene Konstantenmenge, $\mathbf{K}_1 = \{O1, O2, R1\}$, das in Bild 2.1 dargestellte belegte Markov-Netz $G_{\mathbf{L}_1, \mathbf{K}_1}$ ergibt. Für unterschiedliche Mengen

w_i	F_i
0,6	$\forall o \forall r \quad on(o, r) \wedge road(r) \wedge hasSpeed(o, Low) \Rightarrow car(o)$
1,4	$\forall o \quad hasHeight(o, Average) \Rightarrow car(o)$

Tabelle 2.1: Beispiel-MLN \mathbf{L}_1 .

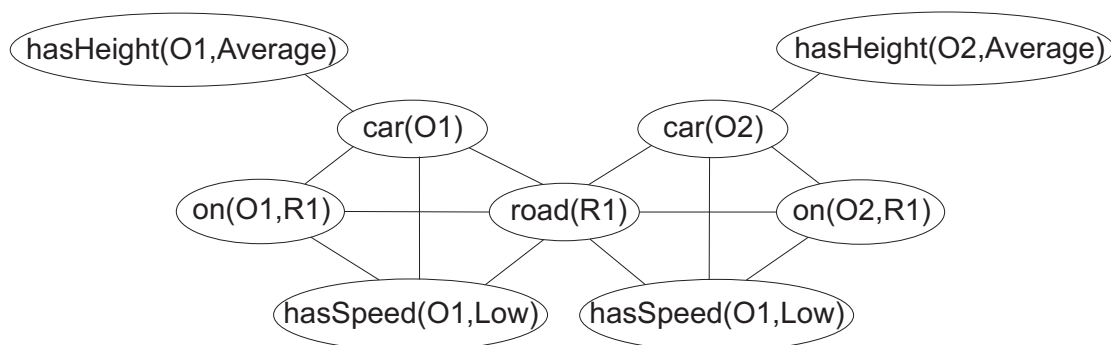


Abbildung 2.1: Das belegte Markov-Netz $G_{\mathbf{L}_1, \mathbf{K}_1}$ für das MLN \mathbf{L}_1 und die Konstantenmenge \mathbf{K}_1 aus dem eingeführten Beispiel.

logischer Konstanten werden entsprechend unterschiedlich große belegte Netze konstruiert, die jedoch gewisse strukturelle Gemeinsamkeiten aufgrund der gleichen logischen Struktur des MLN aufweisen. Gl. (2.14) bewirkt, dass eine mögliche Welt, in der eine oder mehrere der Grundinstanzen der Formeln falsch sind, weniger wahrscheinlich aber nicht unmöglich wird. Auf diese Weise gelingt es, die harten Einschränkungen auf die Menge der möglichen Interpretationen aus der klassischen Prädikatenlogik zu schwächen. Das jeder Formel zugeordnete Gewicht ist ein Maß dafür, wie präzise die entsprechende Formel Wissen über den Diskursbereich vermittelt. Je größer das Gewicht, desto größer ist die Differenz der Logitwerte³ zwischen einer möglichen Welt, in der die Formel wahr ist, und

³Ein Logitwert ist der natürliche Logarithmus eines Odds, d.h. eines Verhältnisses von Wahrscheinlichkeit P zur Gegenwahrscheinlichkeit $(1 - P)$ [Jay03].

einer, in der die Formel falsch ist (bei Festhalten aller anderen Werte). Im allgemeinen Fall werden die Gewichte anhand von Trainingsdaten gelernt oder von einem menschlichen Experten vorgegeben.

Mit einem MLN können alle aussagenlogischen probabilistischen Modelle, die sich als Produkt von Potentialfunktionen formulieren lassen, modelliert werden [Dom09]. Weiterhin ist ein MLN besonders geeignet zum Aufstellen von Modellen mit voneinander abhängigen und nicht gleichverteilten Objekten (*non-i.i.d. models*) [Rae08].

Unter den oben aufgeführten Annahmen ist die Prädikatenlogik erster Stufe ein Spezialfall von MLN, falls alle Gewichte gleich sind und gegen Unendlich streben [Dom09]. Auch wenn die Gewichte endlich groß sind, gilt, dass PL1 in MLN eingebettet ist. Nehmen wir an, dass alle Gewichte der Formeln eines MLNs \mathbf{L} positiv sind. Dabei wird nicht an Allgemeinheit verloren, da jede Formel mit negativem Gewicht w durch die negierte Formel mit Gewicht $-w$ ersetzt werden kann. Falls so eine Wissensbasis erfüllbar ist, dann gilt, dass für jede Konstantenmenge \mathbf{K} , die Belegungen, die die Wissensbasis erfüllen, den Modi der durch $G(\mathbf{L}, \mathbf{K})$ beschriebene Verteilung entsprechen. Die Modi sind die Realisierungen \mathbf{x} , die die Summe aus Gleichung 2.14 maximieren. Diese Summe wird maximiert, wenn alle Grundinstanzen aller Formeln wahr sind, d. h. die Wissensbasis erfüllt ist. Im Unterschied zu einer gewöhnlichen prädikatenlogischen Wissensbasis, kann ein MLN allerdings auch brauchbare Ergebnisse liefern, wenn Widersprüche enthalten sind [Ric06b].

2.3.2 Inferenz

Gegeben ein MLN, Evidenzvariablen und eine Menge von Anfragevariablen, kann anhand des belegten Markov-Netzes Standardinferenz, also zum Beispiel MPE-Inferenz oder bedingte Inferenz, betrieben werden. Der MPE-Zustand für ein MLN ist der Zustand, der die Summe der Gewichte der erfüllten Grundinstanzen der Formeln maximiert [Dom06]. Dieser kann mit einem gewichteten WalkSAT-Solver, z. B. dem MaxWalkSAT [Kau96], effizient approximiert werden. MaxWalkSAT führt eine stochastische lokale Suche durch, indem er bei jedem Durchgang eine nicht erfüllte Klausel auswählt und den spezifizierten Wert eines ihrer Grundatome wechselt. Dieses Atom wird mit einer gewissen Wahrscheinlichkeit zufällig ausgesucht oder so, dass, wenn sein Wahrheitswert geändert wird, die Summe der Gewichte der erfüllten Klauseln maximiert wird. Diese Kombination der Auswahlkriterien verhindert das Steckenbleiben in einem lokalen Optimum.

Bei der bedingten Inferenz wird die bedingte Wahrscheinlichkeitsverteilung der Anfragevariablen gegeben der Zuweisung der Evidenzvariablen bestimmt. Es wird

nur die minimale Untermenge des belegten Markov-Netzes aufgebaut, die notwendig für die Berechnung der bedingten Anfrage ist. Auf diesem kann anschließend ein approximierendes probabilistisches MCMC-Inferenzverfahren, wie der Gibbs-Sampler [Get07b] angewendet werden. Die bedingte Wahrscheinlichkeit eines Grundatoms X_k , wenn seine Markov-Decke $\mathbf{B}(X_k)$ im Zustand b_k ist, beträgt:

$$\begin{aligned}
 & P(X_k = x_k | \mathbf{B}(X_k) = b_k) \\
 &= \frac{\exp\left(\sum_{f_i \in \mathbf{F}_{X_k}} w_i f_i(X_k = x_k, b_k)\right)}{\exp\left(\sum_{f_i \in \mathbf{F}_{X_k}} w_i f_i(X_k = 0, b_k)\right) + \exp\left(\sum_{f_i \in \mathbf{F}_{X_k}} w_i f_i(X_k = 1, b_k)\right)},
 \end{aligned} \tag{2.15}$$

wobei \mathbf{F}_{X_k} die Menge der Grundinstanzen der Formel, in denen X_k vorkommt, ist, und $f_i(X_k = x_k, b_k)$ der Wert des binären Features der i -ten Grundinstanz der Formel für die Zustände $X_k = x_k$ und $\mathbf{B}(X_k) = b_k$ ist [Dom09].

Wenn eine Domäne gleichzeitig probabilistische und deterministische oder nah-deterministische Abhängigkeiten der Variablen aufweist, wird der Raum der möglichen Zuweisungen in Regionen unterteilt, die nicht von den anderen Regionen aus erreichbar sind. Dadurch wird die Ergodizitätsbedingung für die Markov-Kette verletzt, was die schlechten Ergebnisse des Gibbs-Sampler bei Daten solcher Domänen erklärt. Der MC-SAT Algorithmus [Poo06] adressiert diese Problematik. MC-SAT ist ein Slice-Sampling-MCMC-Algorithmus, welcher in jeder Iteration den SampleSAT [Wei04] startet, um Stichproben aus dem „Slice“ zu ziehen. SampleSAT führt bei jeder Iteration entweder den Erfüllbarkeitsalgorithmus WalkSAT [Sel95] mit Wahrscheinlichkeit p oder das Verfahren des simulierten Ausglühens (engl.: *simulated annealing*) [Gör03] mit Wahrscheinlichkeit $(1 - p)$ aus. MC-SAT konvergiert schneller als die Standard MCMC-Methoden bei vorhandenen deterministischen oder nah-deterministischen Abhängigkeiten und kann auf jedem Modell, welches als MLN darstellbar ist, angewendet werden. MC-SAT erfüllt hierbei die Kriterien für MCMC-Algorithmen, nämlich Ergodizität und detaillierter Ausgleich [Poo06].

2.3.3 Parameterlernen

Beim Parameterlernen für Markov-Netze als auch für Markov-Logik-Netze wird zwischen generativen und diskriminativen Verfahren unterschieden. Die generativen Verfahren optimieren die Verbundwahrscheinlichkeitsverteilung aller Zufallsvariablen des Modells. Im Unterschied dazu maximieren die diskriminativen Me-

thoden die bedingte Likelihood einer Menge von Ausgabevariablen gegeben die Menge der Eingangsdaten [Laf01]. In diesem Kapitel werden überwachte Lernverfahren für die Gewichte eines MLNs vorgestellt.

2.3.3.1 Generative Parameterlernverfahren

Die Gewichte eines MLNs können generativ gelernt werden, indem die Likelihood einer Trainingsdatenbank maximiert wird. Die Trainingsdatenbank ist eine relationale Datenbank, die Beispiele für „mögliche Welten“ enthält. Die Menge der Konstanten jedes Variablentyps wird als bekannt angenommen. Weiterhin gilt für die Trainingsdatenbank die *Closed World Assumption*, d. h. nur positive Information wird explizit aufgelistet; bei Fehlen eines Grundprädikats wird dessen Wahrheitswert als falsch angenommen [Gen87]. Gegeben eine Trainingsdatenbank können die Parameter eines MLNs mit Standardverfahren für Markov-Netze gelernt werden. Hat die i -te Formel $n_i(\mathbf{x})$ wahre Grundinstanzen der Formel, folgt aus Gleichung 2.14 die Ableitung der Log-Likelihood nach den Gewichten:

$$\frac{\partial}{\partial w_i} \log P_{\mathbf{w}}(\mathbf{X} = \mathbf{x}) = n_i(\mathbf{x}) - \sum_{\mathbf{x}'} P_{\mathbf{w}}(\mathbf{X} = \mathbf{x}') n_i(\mathbf{x}'), \quad (2.16)$$

wobei die Summe über alle Datenbanken \mathbf{x}' gebildet wird und $P_{\mathbf{w}}(\mathbf{X} = \mathbf{x}')$ die Verteilung für den Parametervektor $\mathbf{w} = (w_1, \dots, w_i)$ ist. Die i -te Komponente des Gradienten ist somit die Differenz aus der Anzahl der wahren Grundinstanzen der Formel i in den Daten und dessen Erwartungswert anhand des Modells. Beide Komponenten der Gleichung 2.16 sind leider schwierig exakt zu berechnen. Das Bestimmen der Anzahl der wahren Grundinstanzen einer prädikatenlogischen Formel ist ein NP-schweres Problem [Dom06]. Für die Berechnung des Erwartungswertes muss über das Modell inferiert werden. Eine effizientere Alternative ist die Optimierung der Pseudo-Likelihood an Stelle der Likelihood [Bes75]. Ist \mathbf{x} eine Trainingsdatenbank und x_k der Wahrheitswert des k -ten Grundatoms, dann ist die Pseudo-Log-Likelihood von \mathbf{x} für gegebenen Gewichtsvektor \mathbf{w} :

$$\log P_{\mathbf{w}}^*(\mathbf{X} = \mathbf{x}) = \sum_{k=1}^n \log P_{\mathbf{w}}(X_k = x_k | \mathbf{B}(X_k)), \quad (2.17)$$

wobei $\mathbf{B}(X_k)$ der Zustand der Markov-Decke (siehe Def. 2.2.4) von X_k ist, d. h. die Belegung der Wahrheitswerte der Grundatome mit denen X_k gemeinsam in einer Formel vorkommt. Der Gradient der Pseudo-Log-Likelihood beträgt folglich:

$$\begin{aligned} \frac{\partial}{\partial w_i} (\log P_{\mathbf{w}}^*(\mathbf{X} = \mathbf{x})) = & \sum_{k=1}^n [n_i(\mathbf{x}) - P_{\mathbf{w}}(X_k = 0 | \mathbf{B}(X_k))n_i(\mathbf{x}_{[X_k=0]}) \\ & - P_{\mathbf{w}}(X_k = 1 | \mathbf{B}(X_k))n_i(\mathbf{x}_{[X_k=1]})], \end{aligned} \quad (2.18)$$

wobei $n_i(\mathbf{x}_{[X_k=1]})$ die Anzahl der wahren Grundinstanzen der i -ten Formel ist, wenn $X_k = 1$ gilt und die restlichen Daten unverändert bleiben. Entsprechendes gilt für $n_i(\mathbf{x}_{[X_k=0]})$. Die Berechnung von Gleichung 2.18 erfordert keine Inferenz über das Modell. Die Parameter lassen sich effektiv mit Gradientenabstieg- oder Quasi-Newton-Verfahren schätzen. Das generative Lernen von MLN-Gewichten in Alchemy [Kok06] verwendet die Pseudo-Log-Likelihood mit dem L-BFGS Algorithmus [Liu89]. Jedes Gewicht wird während der Trainingsphase mit einem Gausschen Prior bestraft, um eine Überanpassung zu verhindern.

2.3.3.2 Diskriminative Parameterlernverfahren

Bei den diskriminativen Parameterlernverfahren werden die Prädikate der Trainingsdatenbank in Evidenzprädikate \mathbf{X} und Anfrageprädikate \mathbf{Y} unterteilt. Das Ziel ist, solche MLN-Gewichte zu lernen, mit welchen anhand der Evidenz die Anfrageprädikate richtig inferiert werden. Die bedingte Likelihood für \mathbf{Y} gegeben \mathbf{X} lautet

$$P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp \left(\sum_i w_i n_i(\mathbf{x}, \mathbf{y}) \right), \quad (2.19)$$

wobei die Normalisierungskonstante $Z_{\mathbf{x}}$ über alle mögliche Welten, die die Evidenz \mathbf{x} erfüllen, normalisiert und $n_i(\mathbf{x}, \mathbf{y})$ die Anzahl der wahren Grundinstanzen der i -ten Klausel, die eines der Suchanfragenprädikate enthält, ist. Die Summe wird über die Menge aller MLN-Klauseln mit mindestens einer Grundinstanz, die eines der Anfrageprädikate enthält, gebildet.

Beim diskriminativen Parameterlernen werden die Gewichte durch Minimieren der negativen bedingten Log-Likelihood (engl.: *conditional log-likelihood (CLL)*) berechnet. In [Sin05] wurde gezeigt, dass das Einsetzen der bedingten Log-Likelihood bessere Ergebnisse erzielt als das der Pseudo-Log-Likelihood. Der

Gradient der negativen bedingten Log-Likelihood beträgt:

$$\begin{aligned} \frac{\partial}{\partial w_i}(-\log P_{\mathbf{w}}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})) &= \\ &= -n_i(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{y}'} P_{\mathbf{w}}(\mathbf{Y} = \mathbf{y}' | \mathbf{X} = \mathbf{x}) n_i(\mathbf{x}, \mathbf{y}') \quad (2.20) \\ &= E_{\mathbf{w}, \mathbf{y}}[n_i(\mathbf{x}, \mathbf{y})] - n_i(\mathbf{x}, \mathbf{y}), \end{aligned}$$

wobei der Erwartungswert $E_{\mathbf{w}, \mathbf{y}}[n_i(\mathbf{x}, \mathbf{y})]$ die erwartete Anzahl von wahren Grundinstanzen der i -ten Formel, die zumindest ein Anfragenprädikat enthält, ist. Der Erwartungswert hängt von dem jeweiligen Gewichtsvektor \mathbf{w} ab, die Summe wird über die möglichen Welten von \mathbf{y} bei bekannter Belegung \mathbf{x} gebildet. Der Erwartungswert lässt sich durch die Anzahl der wahren Grundinstanzen der Formel in dem MPE-Zustand approximieren.

In der Literatur wurden mehrere Algorithmen zum diskriminativen Parameterlernen von Markov-Logik-Netzen vorgestellt. Der in [Sin05] beschriebene Algorithmus baut auf dem Gradientenabstiegalgorithmus Voted Perceptron [Col02] auf, indem der Viterbi-Algorithmus [Rab89] zur Bestimmung des MPE-Zustandes durch den MaxWalkSAT-Solver [Kau96] ersetzt wird. Parameterlernen in Markov-Logik-Netzen ist ein konvexes Optimierungsproblem, weshalb das globale Optimum durch Gradientenabstiegsverfahren garantiert gefunden werden kann [Bib09]. Allerdings können bei schlecht konditionierten Problemen⁴ sehr viele Iterationen notwendig sein, um das Optimum tatsächlich zu finden.

Ein anderes diskriminatives Parameterlernverfahren für Markov-Logik-Netze ist das Diagonal-Newton-Verfahren [Low07]. Das Newton-Verfahren ist in der Lage, bei quadratischen Funktionen das globale Optimum in einem einzigen Schritt zu erreichen. Es verwendet in jedem Durchgang zur Aktualisierung der Gewichte \mathbf{w} den durch die inverse Hessematrix skalierten Gradienten. Die Hessematrix der negativen CLL eines MLNs ist die Kovarianzmatrix [Dom09]. Das Diagonal-Newton-Verfahren trifft die Annahme, dass alle Einträge der Hessematrix, die nicht auf der Hauptdiagonale liegen, Null betragen. Dadurch wird die Hessematrix leicht invertierbar. Gleichzeitig impliziert aber diese Annahme, dass alle Klauseln unkorreliert sind, was in der Praxis oft nicht der Fall ist.

Ein weiteres Verfahren zum diskriminativen Parameterlernen für Markov-Logik-Netze ist das Vorkonditionierte Skalierte Konjugierte Gradientenverfahren (PSCG = *preconditioned scaled conjugate gradient*), vorgestellt in [Low07]. Dieses hat

⁴Ein Problem ist für das Parameterlernproblem schlecht konditioniert, wenn die Konditionszahl, das heißt das Verhältnis zwischen dem größten und kleinsten Eigenwert der Hessematrix, sehr verschieden von 1 ist.

eine bessere Performanz im Vergleich zum Voted-Perceptron-Algorithmus aus [Sin05]. Die Suchrichtungen bei den konjugierten Gradientenverfahren werden bei jeder Iteration so ausgewählt, dass die Gradienten in den vorherigen Suchrichtungen Null bleiben. Die Schrittgröße beim PSCG wird anhand der Hessematrix skaliert. Die Vorkonditionierung, eine lineare Transformation, die die Konditionszahl⁴ des Problems verbessert, verwendet bei PSCG die inverse diagonale Hessematrix. Für detailliertere Information zu den verschiedenen Algorithmen für Parameterlernen wird auf [Dom09] verwiesen.

3 Modellierung

Die Domäne einer Verkehrssituation ist von großer Komplexität geprägt. Um Wissen darin adäquat zu repräsentieren und zu kommunizieren, wird ein gemeinsames Vokabular für die Bedeutung relevanter Begriffe und deren Relationen benötigt. Dieses Vokabular wird durch eine Ontologie festgelegt, die in diesem Kapitel vorgestellt wird. Die Ontologie wird in Prädikatenlogik der ersten Stufe formalisiert. Auf ihrer Basis werden mehrere Modelle entwickelt. Jedes Modell verwendet einen Teil der Ontologiebegriffe, um prädikatenlogische Formeln aufzustellen. Dabei ist zwischen *Axiomen* und *Regeln* zu unterscheiden. Die Axiome repräsentieren Wissen, welches nicht aus anderen Begriffen abgeleitet werden kann. Sie gelten als stets wahr und bekommen im Markov Logik Kontext ohne eine Trainingsphase ein sehr großes positives Gewicht zugewiesen. Die Regeln repräsentieren unsicheres Wissen. Für diese wird anhand von Trainingsdaten ein zugehöriges Gewicht gelernt. Das Gewicht spiegelt die Häufigkeit wider, mit der die Formel in den Trainingsdaten erfüllt wurde.

Die in diesem Kapitel vorgestellten Modelle „Objektklassifikation“, „Relationale Objektklassifikation“, „Inferierte Objektrelationen“ und „Direkte Objektrelationen“ modellieren mit Hilfe von Axiomen und Regeln gewisse Aspekte der Domäne einer Verkehrssituation. An dieser Stelle werden die Modelle noch nicht als Markov-Logik-Netze bezeichnet, da die Formeln noch keine Gewichte zugeordnet bekommen haben. Das Parameterlernen für die Regeln der Modelle wird in Kapitel 6.2 beschrieben. Die resultierenden gewichteten Modelle werden dann zusammen mit den Axiomen zu mehreren verschiedenen Markov-Logik-Netzen in Kapitel 7.1 und Kapitel 7.2 kombiniert und zur Inferenz von Objektclassen und Objektrelationen anhand simulierter bzw. realer Daten eingesetzt. Bei der Beschreibung der Modelle wird die Syntax der Markov Logik [Ric06b] angewendet.

Es folgt eine kurze Einführung zum Begriff der Ontologie mit anschließender Beschreibung der Konzepte und Relationen der entwickelten Ontologie einer Verkehrssituation.

3.1 Ontologie

Der Begriff „Ontologie“ stammt ursprünglich aus der Philosophie und steht für die Lehre vom Sein – genauer von den Möglichkeiten und Bedingungen des Seienden.

Dieser Begriff wurde in die Informatik mit einer neuen Bedeutung überführt und spielt in allen mit Wissen befassten Bereichen, wie künstliche Intelligenz, Datenbanken und Informationssysteme eine zunehmend wichtige Rolle. Um Wissen innerhalb eines komplexen Diskursbereichs zu repräsentieren und zu kommunizieren, wird ein gemeinsames Vokabular für die Bedeutung relevanter Begriffe und deren Relationen benötigt. Dieses Vokabular kann durch eine Ontologie festgelegt werden.

Die bekannteste Definition bezeichnet eine *Ontologie* als eine „explizite formale Spezifikation einer gemeinsamen Konzeptualisierung“ [Gru93]. Diese beschreibt demnach einen Diskursbereich mit Hilfe von formal definierten Begriffen sowie Formeln über deren Zusammenhang. Die Formeln ermöglichen die Ableitung von Rückschlüssen anhand des vorhandenen Wissens mittels Inferenz (logisches Folgen).

Die Begriffe innerhalb einer Ontologie sind in der Regel in Form einer hierarchischen Untergliederung, der sogenannten Taxonomie, gegeben und werden auch als *Konzepte* bezeichnet (z.B. *Fahrzeug*, *Straße*). *Relationen* repräsentieren Querbeziehungen unter den Baumstrukturen der Konzepte (z.B. *Fahrzeug fährt auf Straße*). Auch für Relationen kann eine Taxonomie definiert werden. Ein weiterer Bestandteil einer Ontologie sind *Axiome*. Das sind Aussagen über die Domäne, die immer wahr sind. Axiome repräsentieren Wissen, das nicht aus anderen Begriffen abgeleitet werden kann. *Instanzen* der definierten Konzepte, auch als Individuen bezeichnet, stellen das konkrete zur Verfügung stehende Wissen dar (z.B. *O2* als Instanz des Konzepts *Fahrzeug*). Eine Ontologie bildet zusammen mit ihren Individuen eine Wissensbasis [Gua98]. Wie detailliert eine Ontologie aufgebaut wird, hängt davon ab, von wem und wofür sie eingesetzt wird.

Für das Entwickeln und Prüfen von Ontologien sind eine Reihe von Sprachen, Methoden und Werkzeuge entstanden [GP03]. Die Modellierungsmethoden basieren meistens auf Frames, Prädikatenlogik erster Stufe oder Beschreibungslogik. Die Anwendungsfelder für Ontologien lassen sich in drei Gruppen aufteilen: Kommunikation, Inferenz und Repräsentation sowie Wiederverwendung von Wissen [Gru02]. Eine Ontologie stellt das dargestellte Wissen in einer für den Menschen lesbaren und vom Rechner verarbeitbaren Form bereit und unterstützt somit die Kommunikation auf allen drei Ebenen – zwischen Menschen, zwischen Software-Agenten oder zwischen Mensch und Software-Agent. Die Aussagen über den konkreten Bereich werden nicht im Programmcode „versteckt“, sondern *explizit* formuliert. Sie können somit ohne großen Aufwand geändert werden, falls neues Wissen über den Bereich verfügbar wird. Aufgrund der Ableitungsregeln der Ontologie können logische Schlüsse anhand des vorhandenen Wissens automatisch gezogen werden. Allerdings ist das Erstellen einer Ontologie ein aufwändi-

ger Prozess, weshalb die Wiederverwendung von Wissen ein wichtiger Punkt ist. Ein ontologisches Modell des Verkehrsbereichs kann beispielweise von mehreren Anwendungsbereichen in der Fahrerassistenz von Kreuzungserkennung über Objektklassifikation bis Situationsinterpretation und Verhaltensgenerierung benötigt werden. In einer solchen Repräsentation werden Begriffe, wie Verkehrsteilnehmer, Verkehrsinfrastruktur oder Verkehrsregeln mit ihren jeweiligen Unterklassen und Relationen definiert. Wird eine Ontologie sorgfältig entwickelt, kann sie in anderen Anwendungsbereichen wiederverwendet werden. Außerdem lassen sich auch mehrere bereits existierende Ontologien, die Teile des interessierenden Bereichs beschreiben, zu einer großen Ontologie verschmelzen [Sta04].

Zusammenfassend lässt sich sagen, dass eine Ontologie die explizite Wissensrepräsentation in komplexen Diskursbereichen ermöglicht. Eine solche Repräsentation weist Vorteile im Bezug auf Wartung, Erweiterbarkeit und Flexibilität auf, sowie auch in der Kommunikation und Wiederverwendbarkeit von Wissen.

Im Folgenden werden die Konzepte, Relationen und Instanzen der in dieser Arbeit entwickelten Ontologie vorgestellt. Abbildung 3.1 zeigt diese, wobei in Kästchen Konzepte dargestellt sind und beschriftete Pfeile Relationen benennen.

3.1.1 Konzepte

Die Konzepte der entwickelten Ontologie repräsentieren Objekte und Objektattribute. Es wird zwischen Objekten des Straßennetzes in der Ebene – wie Straßen, Spuren und Markierungen – und 3d-Objekten unterschieden. Die ersteren werden lediglich durch den Oberbegriff der befahrbaren Ebene (*drivableArea*) und eine Unterteilung in Straße (*road*) und nicht-Straße (*roadside*) modelliert. Eine detaillierte Ontologie des Straßennetzes wurde in [Hum09] entwickelt und könnte bei Bedarf in die hier vorgestellte Ontologie integriert werden.

Das Konzept *sceneObject* ist das allgemeinste 3d-Objektkonzept der Ontologie, wie in Abbildung 3.1 veranschaulicht. In einer taxonomischen Struktur werden 3d-Objekte in Kraftfahrzeuge (*motorizedTP*, *TP* = *Traffic Participant*), nichtmotorisierte Verkehrsteilnehmer (*unmotorizedTP*), Objekte der Infrastruktur (*infrastructureObject*), die auf der befahrbaren Ebene liegen oder eine Höhe über Grund kleiner 2m haben (z. B. Baum, Verkehrsschild, Gebäude), sowie Objekte mit Höhe über Grund über 2m (*inAirIrrelObject*) (z. B. Brücke) unterteilt. Die Klassen der motorisierten und nichtmotorisierten Verkehrsteilnehmer werden weiter entsprechend in Auto (*car*), Minibus (*van*), LKW, Bus oder Straßenbahn (*utilityVehicle*), Motorradfahrer (*motorcyclist*), Fahrradfahrer (*bicyclist*) und Fußgänger (*pedestrian*) spezialisiert. Instanzen des Konzepts *sceneObject* sind detektierte 3d-Objekte der

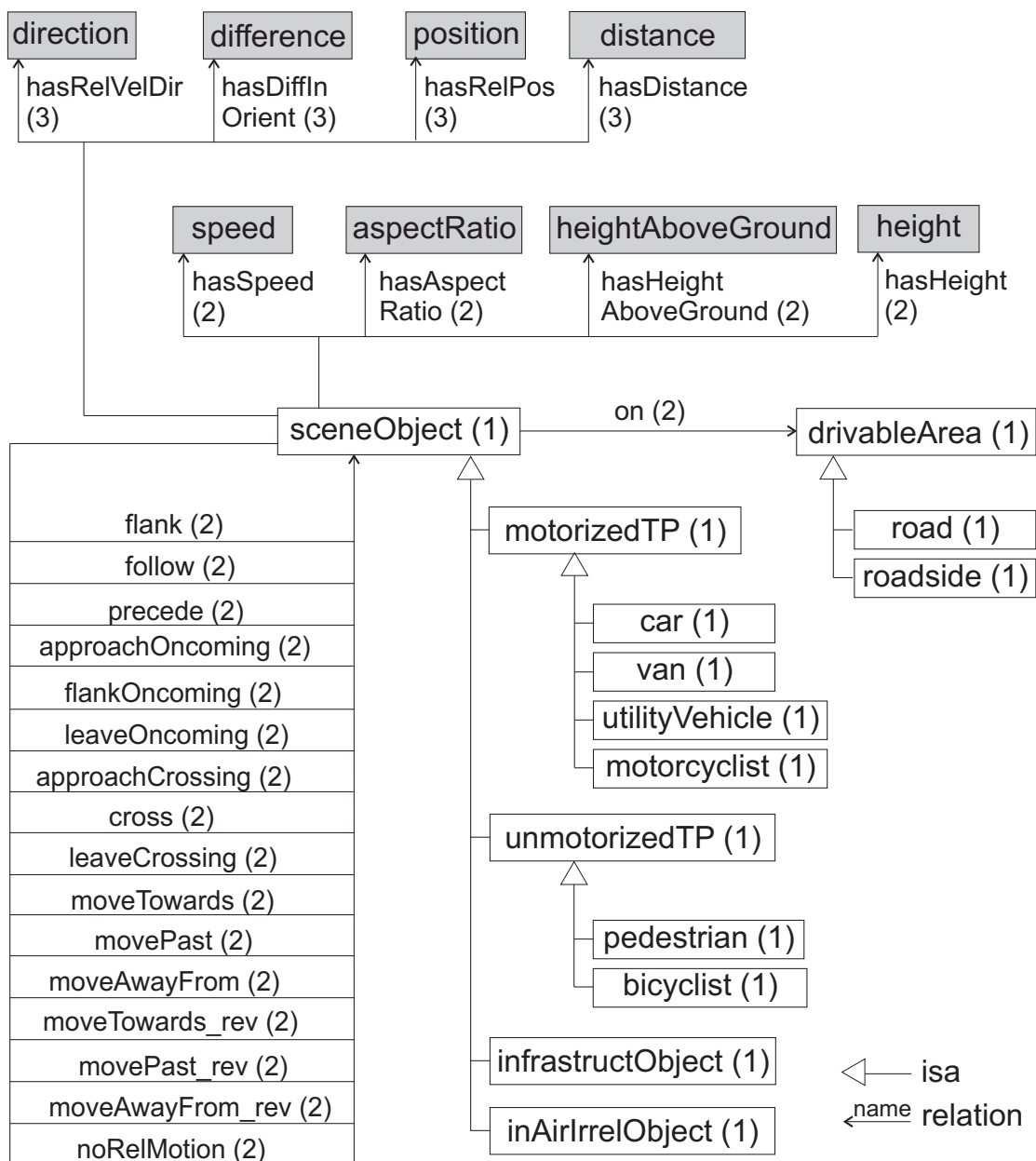


Abbildung 3.1: Konzepte und Relationen der Ontologie einer Verkehrssituation. Dunkle Rechtecke enthalten Objektattributkonzepte, helle Rechtecke enthalten Objektkonzepte. In Klammern ist die Stelligkeit des jeweiligen Prädikats dargestellt.

Szene. Innerhalb einer Spezialisierungsebene sind die Objektkonzepte als paarweise disjunkt und in ihrer Vereinigung erschöpfend modelliert.

Objektmerkmale – wie Form, relative räumliche Position oder Bewegung – werden qualitativ beschrieben, indem ihr quantitativer Wertebereich zweckmäßig in eine Menge diskreter, symbolischer Ausprägungen unterteilt wird. Ziel der qualitativen

Repräsentation ist es, die Daten auf das Notwendige zu reduzieren und damit deren Aussagekraft und die Effizienz der Berechnungen zu erhöhen [Coh02],[Cle97]. Die 3d-Objektattribute der Domäne lassen sich in einfache und relative gruppieren. Die *einfachen* Objektattribute der Ontologie beschreiben Eigenschaften eines Objekts, wie das Verhältnis der Höhe zu Breite eines Objekts (Längenverhältnis), die Höhe der Bounding-Box eines Objekts, die Höhe über der befahrbaren Ebene der linken unteren Ecke der Bounding-Box des Objekts und die Objektgeschwindigkeit. Die *relativen* Objektattribute beschreiben eine Beziehung zweier Objekte. Folgende relative Attribute werden modelliert: relative Bewegungsrichtung, Differenz der Geschwindigkeitsrichtungen, relative Position und relativer Abstand. Wie in Abbildung 3.1 veranschaulicht wird für jedes dieser Objektattribute ein Ontologiekonzept definiert (z. B. *height*, *speed*, *position*).

Die symbolischen Werte, die jedes Attribut einnehmen darf, sind die Instanzen des jeweiligen Konzepts. Das Attribut Objekthöhe *height* hat beispielsweise vier Instanzen *Small*, *Average*, *Large* und *VeryLarge*. Eine solche qualitative Abstraktion wird für alle eingeführten Objektattribute vorgenommen. Diese wird in Kapitel 4 detailliert beschrieben. Auf eine Darstellung der möglichen Instanzen in Abbildung 3.1 wurde aus Gründen der Übersichtlichkeit verzichtet. Die Attributenkonzepte der Ontologie sowie auch ihre mögliche Instanzen sind als erschöpfend und sich gegenseitig ausschließend modelliert.

3.1.2 Relationen

Relationen stellen Querbeziehungen zwischen Konzepten dar (*sceneObject hasHeight height*) und werden durch Relationspfeile in Abbildung 3.1 visualisiert. In der hier aufgebauten Ontologie lassen sich die Relationen in Attributen- und Objektrelationen gruppieren.

Die *Attributrelationen* repräsentieren Beziehungen zwischen Objekten und Attributen:

- Längenverhältnis haben: *hasAspectRatio*,
- Objekthöhe haben: *hasHeight*,
- Höhe über Grund haben: *hasHeightAboveGround*,
- Geschwindigkeit haben: *hasSpeed*,
- relative Bewegungsrichtung haben: *hasRelVelDir*,
- Differenz der Geschwindigkeitsrichtungen haben: *hasDiffInOrient*,

- relative Position haben: *hasRelPos*,
- relativen Abstand haben: *hasDistance*.

Die *Objektrelationen* repräsentieren elementare Geschehen zwischen zwei Objekten. Die Objektrelation *on* ist die einzige Relation, die ein 3d-Objekt mit einem Ebenenobjekt in Beziehung setzt (sceneObject *on* drivableArea). Alle anderen Objektrelationen beschreiben Zusammenhänge zwischen zwei 3d-Objekten und werden im Folgenden vorgestellt. Bei zwei sich bewegenden Objekten werden die modellierten Objektrelationen in Abhängigkeit von der relativen Bewegungsrichtung in drei Gruppen eingeteilt:

- beide Objekte bewegen sich in gleicher Richtung
 - folgen: *follow*
 - flankieren: *flank*
 - vorausfahren: *precede*
- beide Objekte bewegen sich in entgegengesetzter Richtung
 - sich entgegenkommend annähern: *approachOncoming*
 - entgegenkommend flankieren: *flankOncoming*
 - sich voneinander entgegengesetzt entfernen: *leaveOncoming*
- beide Objekte bewegen sich in kreuzender Richtung
 - sich kreuzend annähern: *approachCrossing*
 - kreuzen: *cross*
 - sich kreuzend entfernen: *leaveCrossing*

Die Umkehrrelation von *follow* ist *precede*.

Für den Fall, dass eines der Objekte sich bewegt und das andere nicht, werden folgende Objektrelationen und entsprechende Umkehrrelationen eingeführt:

- sich einem stehenden Objekt annähern: *moveTowards*
- ein stehendes Objekt passieren: *movePast*
- sich von einem stehenden Objekt entfernen: *moveAwayFrom*
- das stehende Objekt wird angenähert: *moveTowards_rev*

- das stehende Objekt wird passiert: *movePast_rev*
- das stehende Objekt wird zurückgelassen: *moveAwayFrom_rev*

Die Objektrelation *noRelMotion* gilt zwischen zwei sich nicht bewegenden 3d-Objekten. Diese wird aus Gründen der Vollständigkeit eingeführt, damit für jedes 3d-Objektpaar eine Aussage über das Geschehen gemacht werden kann.

Die modellierten Geschehen sind sehr allgemein aufgefasst. Sie beschreiben die relative Bewegung zweier 3d-Objekte unabhängig von derer Klasse oder objekt-klassenabhängiger Infrastruktur (wie z. B. Spur). Die so modellierte relative Bewegungsbeschreibung könnte somit auch auf andere Domänen übertragen werden.

Abbildung 3.2 visualisiert die Bedeutung der Objektrelationen. Die Objektrelationen gelten für alle definierten Objektklassen, im Bild sind beispielhaft zwei Autos gezeichnet. Das blaue Auto stellt das Hauptobjekt und das graue das Referenzobjekt der jeweiligen Relation dar. Für jede Relation werden mehrere mögliche Szenen mit Hilfe der grauen Autos dargestellt, in denen die entsprechende Objektrelation gültig wäre. Für die kreuzenden Geschehen ist nur ein Winkel von 90° veranschaulicht, was allerdings als Beispiel zu sehen ist, da ansonsten beliebige Winkel berücksichtigt werden. Demnach ist z. B. unter *follow* die Bewegungsrelation zwischen einem Objekt und einem sich in die gleiche qualitative Richtung vorausbewegenden Objekt zu verstehen. Dabei muss das vordere Objekt nicht direkt vor dem hinteren Objekt sein (sich in der gleichen Spur bewegen), dieses kann links oder rechts vorne sein.

3.1.3 Formalisierung der Ontologie

Die Ontologie wird in Prädikatenlogik erster Stufe formalisiert. Dabei werden Prädikatensymbole mit entsprechender Stelligkeit für die Objektkonzepte sowie auch für die Attribut- und Objektrelationen definiert. Für die Attributkonzepte und deren symbolischen Werte werden Terme deklariert (z. B. *speed*, *VeryLow*). Die Benennungen der Prädikatensymbole und Terme entsprechen den Bezeichnungen in Abbildung 3.1. In der Abbildung wird auch die Stelligkeit der Prädikate in Klammern angegeben. Dabei sind alle Argumente der Prädikate typisiert.

Für die Objektkonzepte der Ontologie werden einstellige Prädikate, z. B. *sceneObject(object)*, *drivableArea(plane)*, eingeführt.

Die einfachen Attributrelationen, die Eigenschaften eines Objekts beschreiben (z. B. Objektgeschwindigkeit), werden mit zweistelligen Prädikaten formalisiert (z. B. *hasSpeed(object,speed)*). Die erste Variable des jeweiligen Prädikats ist vom

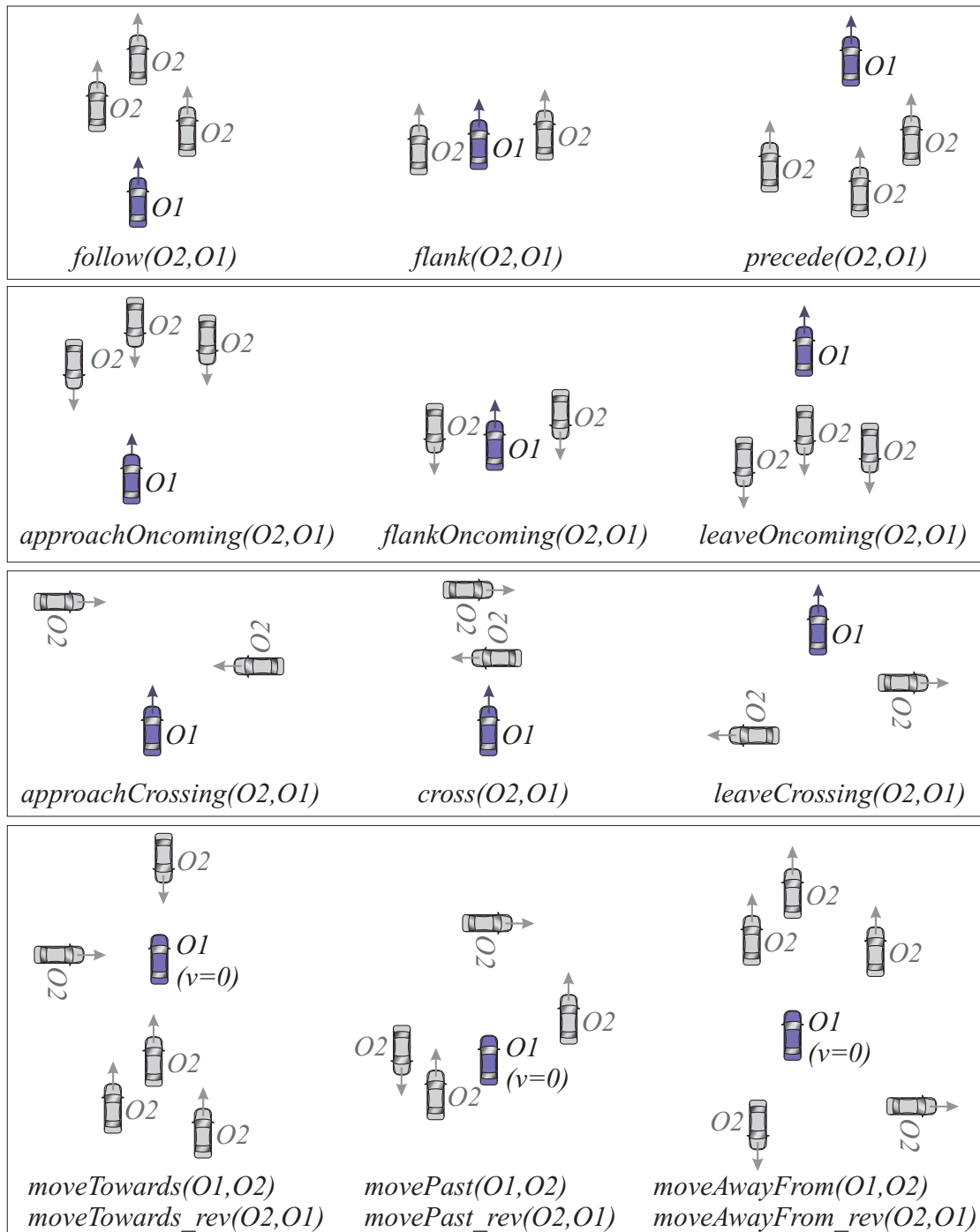


Abbildung 3.2: Exemplarische Verkehrsszenen, die die Bedeutung der Objektrelationen beispielhaft mit zwei Autos visualisieren. Das blaue Auto stellt das Hauptobjekt und das graue das Referenzobjekt der jeweiligen Relation dar.

Typ 3d-Objekt. Das zweite Argument wird vom entsprechenden Attributenterm besetzt.

Die relativen Attributrelationen (z. B. relativer Abstand) beschreiben Beziehungen

zweier Objekte und werden durch dreistellige Prädikate formalisiert. Diese haben als Argumente zwei Variablen vom Typ 3d-Objekt und der jeweilige Attributen-term, z. B. *hasDistance(object,object,distance)*.

Für die Objektrelationen werden zweistellige Prädikate deklariert. Diese haben jeweils zwei Variablen von Typ 3d-Objekt, z. B. *follow(object,object)*, oder eine Variable vom Typ 3d-Objekt und eine von Typ Ebenenobjekt, wie bei *on(object,plane)*.

Für alle Prädikate mit zwei Variablen vom Typ *object* ist die Reihenfolge der Objekte bei der Instantiierung von Bedeutung. Der erste Eintrag bezeichnet das Referenzobjekt und der zweite das Hauptobjekt, d. h. die Objektrelation *follow(O2,O1)* würde man beispielsweise folgendermaßen lesen „*O1* folgt *O2*“ oder die relative Attributrelation *hasRelPos(O2,O1,S)* - „*O1* befindet sich südlich von *O2*“. Dies ist besonders bei nichtsymmetrischen Relationen von größter Wichtigkeit.

Die Axiome der Ontologie, die z. B. die Objekttaxonomie oder Symmetrieeigenschaften der Relationen festlegen, werden in den Kapiteln zu den Modellen vorgestellt. Diese sind prädikatenlogische Formeln, die sichere Aussagen über die Domäne repräsentieren. Dieses Vorwissen kann nicht aus anderen Begriffen abgeleitet werden und spiegelt Modellierungsentscheidungen wider. Die entwickelte Ontologie dient als Grundlage beim Aufbau der Modelle, die in den nächsten Kapiteln vorgestellt werden.

3.2 Modell: Objektklassifikation (OK)

In diesem Kapitel wird ein Modell zur Klassifikation von Objekten einer Verkehrsszene beschrieben. Dabei werden alle Konzepte für die 3d-Objekte und die einfachen Objektattribute aus dem Ontologievokabular verwendet. Im Folgenden werden die Axiome und Regeln des Modells aufgeführt.

3.2.1 Axiome

Die Taxonomie der 3d-Objekte aus Abbildung 3.1 wird durch Axiome formalisiert. Die Konzepte innerhalb einer Spezialisierungsebene sind als paarweise disjunkt und in ihrer Vereinigung erschöpfend modelliert. Beispielhaft werden hier die Axiome für die Spezialisierungen von *unmotorizedTP* aufgeführt, die restlichen Taxonomiezweige werden analog beschrieben:

$$\begin{aligned} \forall o \quad & \text{unmotorizedTP}(o) \Rightarrow \text{pedestrian}(o) \vee \text{bicyclist}(o). \\ \forall o \quad & \text{pedestrian}(o) \Rightarrow \text{unmotorizedTP}(o) \wedge \neg \text{bicyclist}(o). \\ \forall o \quad & \text{bicyclist}(o) \Rightarrow \text{unmotorizedTP}(o). \end{aligned}$$

Der Punkt am Ende der Formel definiert diese laut der Markov-Logik-Syntax als Axiome. Das Ausrufezeichen ist bei Markov-Logik das Zeichen für Negation.

Die Beziehungen zwischen der Relation *hasHeightAboveGround* und der oberen Ebene der 3d-Objekttaxonomie werden ebenfalls durch Axiome festgelegt. Zum Beispiel besagt eins der Axiome, dass ein sich auf dem Boden befindendes Objekt relevant ist und somit einer der Objektklassen motorisierter Verkehrsteilnehmer, nicht motorisierter Verkehrsteilnehmer oder Infrastrukturobjekt angehört. Die harte Modellierung an dieser Stelle führt dazu, dass beispielsweise ein Auto auf einer Brücke direkt als ein nichtrelevantes Objekt in großer Höhe über dem Boden klassifiziert wird.

3.2.2 Regeln

Der Zusammenhang zwischen den Objektklassen und den restlichen drei einfachen Attributrelationen – Objekthöhe, -geschwindigkeit und -längenverhältnis, wird durch Regeln formalisiert. Für jede der drei Attributrelationen wird je ein Block von acht Regeln geschrieben, welche auf der rechten Seite die acht 3d-Objektkonzepte von höchster Spezialisierung beinhalten. Beispielhaft hier einige der Regeln für *hasAspectRatio*:

$$\begin{aligned} \forall o, r \quad & \text{hasAspectRatio}(o, +r) \Rightarrow \text{car}(o) \\ \forall o, r \quad & \text{hasAspectRatio}(o, +r) \Rightarrow \text{pedestrian}(o) \\ \forall o, r \quad & \text{hasAspectRatio}(o, +r) \Rightarrow \text{infrastrObject}(o) \end{aligned}$$

Das Pluszeichen vor der Variable r bedeutet, dass während der Trainingsphase alle möglichen symbolischen Werte des Attributs *aspectRatio* automatisch nacheinander substituiert werden und für jede dieser Regeln ein eigenes Gewicht gelernt wird. Die Regeln bekommen laut der Markov-Logik-Syntax, im Gegensatz zu den Axiomen, keinen Punkt am Ende.

3.3 Modell: Relationale Objektklassifikation (ROK)

Das Modell „Relationale Objektklassifikation“ ist als eine Erweiterung von dem in Kapitel 3.2 vorgestellten Modell zu sehen. Es baut auf den Axiomen und Regeln des Modells „Objektklassifikation“ auf und setzt darüberhinaus Objektklassen und Objektrelationen der Ontologie in Beziehung.

3.3.1 Axiome

Zusätzlich zu den in Kapitel 3.2.1 eingeführten Axiomen werden in dem Modell „Relationale Objektklassifikation“ die Struktur und Eigenschaften der Objektrelationen der Ontologie als klassische prädikatenlogische Formeln modelliert.

Alle Objektrelationen sind *irreflexiv*, d. h. kein Objekt steht in Relation zu sich selbst. Die Objektrelationen sind durch zweistellige Prädikate formalisiert. Für eine beliebige zweistellige Relation rel_i sieht das Irreflexivitätsaxiom folgendermaßen aus:

$$\forall o \quad ! rel_i(o, o).$$

Weiterhin ist die Eigenschaft Symmetrie der Objektrelationen mit Axiomen festgelegt. Symmetrische Relationen sind: *flank*, *flankOncoming*, *approachOncoming*, *leaveOncoming*, *noRelMotion*. Die restlichen sind nicht symmetrisch. Für die symmetrischen Objektrelationen gibt es folgendes Axiom

$$\forall o1, o2 \quad rel_i(o1, o2) \Leftrightarrow rel_i(o2, o1).$$

und für die nicht symmetrischen folgendes

$$\forall o1, o2 \quad rel_i(o1, o2) \Rightarrow ! rel_i(o2, o1).$$

Für einige der Objektrelationen werden auch Umkehrrelationen postuliert, wie z. B. für *follow*:

$$\forall o1, o2 \quad follow(o1, o2) \Leftrightarrow precede(o2, o1).$$

oder auch für die Relationen zwischen einem stehenden und einem sich bewegendem Objekt.

3.3.2 Regeln

Das Modell „Relationale Objektklassifikation“ enthält die Regeln aus Kapitel 3.2.2, welche die 3d-Objekt-konzepte mit den Attributen der einzelnen Objekte kombinieren. Hinzu kommen folgende Regeln, die Objektklassen und Objektrelationen in Bezug setzen.

Die Objektrelation *on* ordnet jedem 3d-Objekt der Szene eine momentane Position zu – auf der Straße oder auf der Straßenseite. Die verschiedenen Objektklassen haben charakteristische Aufenthaltsorte in einer Verkehrsszene in Abhängigkeit von der Objektgeschwindigkeit. Diese Zusammenhänge beschreiben folgende Regeln:

$$\begin{aligned}
\forall o,y \quad & on(o,y) \wedge road(y) \wedge * hasSpeed(o,Zero) \Rightarrow bicyclist(o) \\
\forall o,y \quad & on(o,y) \wedge roadside(y) \wedge * hasSpeed(o,Zero) \Rightarrow bicyclist(o) \\
\forall o,y \quad & on(o,y) \wedge road(y) \wedge * hasSpeed(o,Zero) \Rightarrow pedestrian(o) \\
\forall o,y \quad & on(o,y) \wedge roadside(y) \wedge * hasSpeed(o,Zero) \Rightarrow pedestrian(o) \\
\forall o,y \quad & on(o,y) \wedge road(y) \wedge * hasSpeed(o,Zero) \Rightarrow motorizedTP(o) \\
\forall o,y \quad & on(o,y) \wedge roadside(y) \wedge * hasSpeed(o,Zero) \Rightarrow motorizedTP(o) \\
\forall o,y \quad & on(o,y) \wedge road(y) \wedge * hasSpeed(o,Zero) \Rightarrow infrastrObject(o) \\
\forall o,y \quad & on(o,y) \wedge roadside(y) \wedge * hasSpeed(o,Zero) \Rightarrow infrastrObject(o)
\end{aligned}$$

Der Asterisk * vor dem Prädikat *hasSpeed* bedeutet, dass während der Trainingsphase jeweils ein Gewicht für die Regel mit dem positiven Prädikat *hasSpeed(o,Zero)* und mit dem negierten Prädikat *!hasSpeed(o,Zero)* gelernt wird.

Eine andere Gruppe von Regeln verbindet die relative Bewegung zweier Objekte mit deren Klassenzugehörigkeit:

$$\begin{aligned}
\forall o1,o2 \quad & !(o1=o2) \wedge rel_i(o1,o2) \Rightarrow motorizedTP(o1) \wedge motorizedTP(o2) \\
\forall o1,o2 \quad & !(o1=o2) \wedge rel_i(o1,o2) \Rightarrow motorizedTP(o1) \wedge unmotorizedTP(o2) \\
\forall o1,o2 \quad & !(o1=o2) \wedge rel_i(o1,o2) \Rightarrow unmotorizedTP(o1) \wedge motorizedTP(o2) \\
\forall o1,o2 \quad & !(o1=o2) \wedge rel_i(o1,o2) \Rightarrow unmotorizedTP(o1) \wedge unmotorizedTP(o2) \\
\forall o1,o2 \quad & !(o1=o2) \wedge rel_i(o1,o2) \Rightarrow !infrastrObject(o1) \wedge !infrastrObject(o2)
\end{aligned}$$

Anstelle von rel_i werden alle Objektrelationen eingesetzt, mit Ausnahme der jeweiligen Umkehrrelationen und der Relation *noRelMotion*.

3.4 Modell: Inferierte Objektrelationen (IR)

Das Modell „Inferierte Objektrelationen“ (IR) modelliert die Abhängigkeiten zwischen den relativen Objektattributen *hasRelPos*, *hasDistance* und *hasDiffInOrient* und allen Objektrelationen außer den Umkehrrelationen und *noRelMotion*. Während der Lernphase der Regeln dieses Modells werden formale Definitionen der Objektrelationen und deren Eigenschaften gelernt.

3.4.1 Axiome

Die Instanzen der relativen Objektattribute *hasRelPos*, *hasDistance* und *hasDiffInOrient*, vorgestellt in Kapitel 4, werden durch eine erste Gruppe von

Axiomen im Modell IR als paarweise disjunkt formalisiert. Dafür wird pro Attributenprädikat $pred_i$ eine Menge von Formeln definiert, die der Anzahl n der möglichen symbolischen Instanzen $Inst_i$ des Attributs entspricht. Jede dieser Axiome ist nach dem folgenden Muster, z. B. für $n = 3$, aufgebaut:

$$\forall o1, o2 \quad pred_i(o1, o2, Inst_1) \Rightarrow !(o1=o2) \wedge !pred_i(o1, o2, Inst_2) \wedge !pred_i(o1, o2, Inst_3).$$

Die Instanzen in jeder Formel werden permutiert. Es wird gefordert, dass die Variablen $o1$ und $o2$ unterschiedliche Objekte repräsentieren.

Beispielsweise kann die Attributrelation *hasDistance* fünf symbolische Werte annehmen *VeryClose*, *Close*, *Medium*, *Far* und *VeryFar*. Hier sind die ersten zwei der fünf Formeln für dieses Prädikat aufgelistet:

$$\begin{aligned} \forall o1, o2 \quad hasDistance(o1, o2, VeryClose) \Rightarrow & !(o1=o2) \wedge !hasDistance(o1, o2, Close) \wedge \\ & !hasDistance(o1, o2, Medium) \wedge !hasDistance(o1, o2, Far) \wedge \\ & !hasDistance(o1, o2, VeryFar). \end{aligned}$$

$$\begin{aligned} \forall o1, o2 \quad hasDistance(o1, o2, Close) \Rightarrow & !(o1=o2) \wedge !hasDistance(o1, o2, VeryClose) \wedge \\ & !hasDistance(o1, o2, Medium) \wedge !hasDistance(o1, o2, Far) \wedge \\ & !hasDistance(o1, o2, VeryFar). \end{aligned}$$

Eine weitere Gruppe Axiome beschreibt Symmetrieeigenschaften der verwendeten Objektattribute. Die Formeln für die dreistelligen Prädikate sind analog aufgebaut zu den Formeln für zweistellige Prädikate aus Kapitel 3.3.1. Die Objektattribute *hasDistance* und *hasDiffInOrient* sind symmetrisch, d. h. es gilt, dass z. B. wenn ein Objekt sehr nah zu einem zweiten Objekt ist, dann ist der zweite Objekt auch sehr nah zum ersten. Dies wird auf folgende Weise formalisiert:

$$\begin{aligned} \forall o1, o2, dist \quad hasDistance(o1, o2, dist) & \Leftrightarrow hasDistance(o2, o1, dist). \\ \forall o1, o2, diff \quad hasDiffInOrient(o1, o2, diff) & \Leftrightarrow hasDiffInOrient(o2, o1, diff). \end{aligned}$$

Die Symmetrieeigenschaften rühren aus der Vorgehensweise bei der Bestimmung der Attribute, die in Kapitel 4 beschrieben ist.

Darüber hinaus wird der Zusammenhang zwischen den Objektrelationen und dem Attribut Objektgeschwindigkeit durch zwei Axiome formalisiert:

$$\begin{aligned} \forall o1, o2 \quad !(o1=o2) \wedge !hasSpeed(o1, Zero) \wedge !hasSpeed(o2, Zero) \Rightarrow & follow(o1, o2) \vee \\ & follow(o2, o1) \vee flank(o1, o2) \vee flank(o2, o1) \vee approachCrossing(o1, o2) \vee \\ & approachCrossing(o2, o1) \vee approachOncoming(o1, o2) \vee \\ & approachOncoming(o2, o1) \vee leaveCrossing(o1, o2) \vee leaveCrossing(o2, o1) \vee \\ & leaveOncoming(o1, o2) \vee leaveOncoming(o2, o1) \vee cross(o1, o2) \vee \\ & cross(o2, o1) \vee flankOncoming(o1, o2) \vee flankOncoming(o2, o1). \end{aligned}$$

$$\forall o1,o2 \quad !(o1=o2) \wedge hasSpeed(o1,Zero) \wedge !hasSpeed(o2,Zero) \Rightarrow movePast(o1,o2) \vee moveAwayFrom(o1,o2) \vee moveTowards(o1,o2).$$

Dabei wird lediglich zwischen Geschwindigkeit Null und verschieden Null unterschieden. Die Axiome legen fest, welche der Objektrelationen gelten dürfen, wenn sich beide Objekte bewegen oder wenn eins der Objekte steht.

3.4.2 Regeln

Regeln beschreiben die Zusammenhänge zwischen den Instanzen der Attribute und den Objektrelationen. Folgende Regeln gibt es für unsymmetrische Relationen, wie zum Beispiel *follow*,

$$\begin{aligned} \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o1,o2,+p) \Rightarrow follow(o1,o2) \\ \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o1,o2,+p) \Rightarrow follow(o2,o1) \\ \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o2,o1,+p) \Rightarrow follow(o1,o2) \\ \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o2,o1,+p) \Rightarrow follow(o2,o1) \\ \forall o1,o2,dist \quad &!(o1=o2) \wedge hasRelDist(o1,o2,+dist) \Rightarrow follow(o1,o2) \\ \forall o1,o2,d \quad &!(o1=o2) \wedge hasDiffInOrient(o1,o2,+d) \Rightarrow follow(o1,o2) \end{aligned}$$

und folgende für symmetrische Relationen

$$\begin{aligned} \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o1,o2,+p) \Rightarrow flank(o1,o2) \\ \forall o1,o2,p \quad &!(o1=o2) \wedge hasRelPos(o1,o2,+p) \Rightarrow flank(o2,o1) \\ \forall o1,o2,dist \quad &!(o1=o2) \wedge hasRelDist(o1,o2,+dist) \Rightarrow flank(o1,o2) \\ \forall o1,o2,d \quad &!(o1=o2) \wedge hasDiffInOrient(o1,o2,+d) \Rightarrow flank(o1,o2) \end{aligned}$$

Das Pluszeichen vor den Variablen bedeutet, dass während der Trainingsphase alle möglichen symbolischen Werte des jeweiligen Attributes automatisch nacheinander substituiert werden und für jede dieser Formeln ein eigenes Gewicht gelernt wird. Dadurch wird eine probabilistisch-logische Definition für jede der Objektrelationen gelernt.

3.5 Modell: Direkte Objektrelationen (DR)

Im Unterschied zu dem Modell „Inferierte Objektrelationen“ werden bei dem Modell „Direkte Objektrelationen“ (DR) die Definitionen für die Objektrelationen der Ontologie, ähnlich wie in [Lul08], manuell vorgegeben. Die Objektrelationen werden in Abhängigkeit von zwei der relativen Objektattribute definiert. Dabei werden die Relationen in zwei Gruppen aufgeteilt. Die erste Gruppe enthält alle Relationen, die Beziehungen zwischen zwei sich bewegenden Objekten beschreiben.

Diese werden in Abhängigkeit von den qualitativen Werten der Objektattribute relative Position *hasRelPos* und relative Geschwindigkeitsrichtung *hasRelVelDir* definiert. Die sich daraus ergebende Matrix für die Relationen zwischen zwei sich bewegenden Objekten ist in Abbildung 3.3 veranschaulicht. Die darin verwendeten Abkürzungen für die Objektrelationen listet Tabelle 3.1 auf.

Tabelle 3.1: Abkürzungen der Objektrelationen, welche in den Definitionsmatrizen in den Abbildungen 3.3 und 3.4 verwendet werden.

Abkürzung	Objektrelation
<i>fo</i>	<i>follow</i>
<i>fl</i>	<i>flank</i>
<i>pre</i>	<i>precede</i>
<i>aOn</i>	<i>approachOncoming</i>
<i>flOn</i>	<i>flankOncoming</i>
<i>lOn</i>	<i>leaveOncoming</i>
<i>aCr</i>	<i>approachCrossing</i>
<i>cr</i>	<i>cross</i>
<i>lCr</i>	<i>leaveCrossing</i>
<i>mT</i>	<i>moveTowards</i>
<i>mP</i>	<i>movePast</i>
<i>mAF</i>	<i>moveAwayFrom</i>
<i>mT_r</i>	<i>moveTowards_rev</i>
<i>mP_r</i>	<i>movePast_rev</i>
<i>mAF_r</i>	<i>moveAwayFrom_rev</i>

Die zweite Gruppe enthält alle Objektrelationen mit einem stehenden und einem sich bewegenden Objekt. Diese werden anhand der qualitativen Werte der Objektattribute relative Position *hasRelPos* und relativer Abstand *hasDistance* definiert. Die resultierende Matrix mit den Abkürzungen aus Tabelle 3.1 zeigt Abbildung 3.4. Die Umkehrrelationen der zweiten Gruppe Objektrelationen folgen direkt aus den Definitionen der Hauptrelationen. Diese wurden aus Übersichtsgründen in der Abbildung nicht dargestellt.

Die Notwendigkeit für diese Gruppenunterscheidung beruht darauf, dass die relative Geschwindigkeitsrichtung nur für zwei sich bewegende Objekte bestimmt werden kann. Die relative Position zweier Objekte dagegen wird auf die Bewegungsachse des Referenzobjekts der Relation bezogen, so dass eine sinnvolle Aussage gemacht werden kann, auch wenn sich das Hauptobjekt der Relation in Ruhe befindet. Einzelheiten zu der Gewinnung der relativen Objektattribute und deren abstrahierten Symbolwerten sind in Kapitel 4 zu finden.

Die Auswertung des jeweiligen Attributenpaares anhand der Definitionsmatrizen

	<i>hasRelPos</i>							
<i>hasRelVelDir</i>	<i>N</i>	<i>NE</i>	<i>E</i>	<i>SE</i>	<i>S</i>	<i>SW</i>	<i>W</i>	<i>NW</i>
<i>Parallel_N</i>	<i>pre</i>	<i>pre</i>	<i>fl</i>	<i>fo</i>	<i>fo</i>	<i>fo</i>	<i>fl</i>	<i>pre</i>
<i>Oblique_NE</i>	<i>cr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>aCr</i>	<i>aCr</i>	<i>aCr</i>
<i>Perp_E</i>	<i>cr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>aCr</i>	<i>aCr</i>
<i>Oblique_SE</i>	<i>cr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>aCr</i>
<i>Parallel_S</i>	<i>aOn</i>	<i>aOn</i>	<i>flOn</i>	<i>lOn</i>	<i>lOn</i>	<i>lOn</i>	<i>flOn</i>	<i>aOn</i>
<i>Oblique_SW</i>	<i>cr</i>	<i>aCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>
<i>Perp_W</i>	<i>cr</i>	<i>aCr</i>	<i>aCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>
<i>Oblique_NW</i>	<i>cr</i>	<i>aCr</i>	<i>aCr</i>	<i>aCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>	<i>lCr</i>

Abbildung 3.3: Definitionsmatrix der Relationen mit zwei sich bewegenden Objekten.

	<i>hasRelPos</i>							
<i>hasDistance</i>	<i>N</i>	<i>NE</i>	<i>E</i>	<i>SE</i>	<i>S</i>	<i>SW</i>	<i>W</i>	<i>NW</i>
<i>Zero</i>	<i>mT</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>	<i>mAF</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>
<i>VeryClose</i>	<i>mT</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>	<i>mAF</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>
<i>Close</i>	<i>mT</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>	<i>mAF</i>	<i>mP</i>	<i>mP</i>	<i>mP</i>
<i>Medium</i>	<i>mT</i>	<i>mT</i>	<i>mP</i>	<i>mAF</i>	<i>mAF</i>	<i>mAF</i>	<i>mP</i>	<i>mT</i>
<i>Far</i>	<i>mT</i>	<i>mT</i>	<i>mP</i>	<i>mAF</i>	<i>mAF</i>	<i>mAF</i>	<i>mP</i>	<i>mT</i>
<i>VeryFar</i>	<i>mT</i>	<i>mT</i>	<i>mP</i>	<i>mAF</i>	<i>mAF</i>	<i>mAF</i>	<i>mP</i>	<i>mT</i>

Abbildung 3.4: Definitionsmatrix der Relationen mit einem stehenden und einem sich bewegenden Objekt. Aus Übersichtsgründen wurden die entsprechenden Umkehrrelationen hier nicht eingetragen.

des Modells DR ist als ein Vorverarbeitungsschritt zu betrachten. Die so bestimmten Objektrelationen werden in die Eingangsdaten für die Inferenz mit einem MLN aufgenommen. Die Domäne dieses Markov-Logik-Netzes enthält daher nicht die für die Bestimmung der Objektrelationen notwendigen relativen Objektattribute, sondern lediglich die Prädikate der Relationen.

4 Qualitative Abstraktion der Objektattribute

Evidenzen bezeichnen vorhandene Informationen über konkrete Objekte in einer zu analysierenden Verkehrsszene. Je nachdem, ob man mit simulierten oder realen Daten arbeitet, werden die entsprechenden Informationen als logische Konstanten manuell vorgegeben oder aus den quantitativen Sensordaten mittels einer qualitativen Abstraktion ermittelt. Der quantitative Wertebereich der Objektattribute wird dazu zweckmäßig diskretisiert und auf logische Konstanten abgebildet. Die Objektattribute, die durch die Ontologie aus Kapitel 3.1 festgelegt sind, werden in *einfache* - die sich auf ein Objekt beziehen, und *relative* - die sich auf zwei Objekte beziehen, unterteilt. Im Folgenden wird die qualitative Abstraktion der Objektattribute vorgestellt. Ein Überblick über Methoden zur qualitativen Repräsentation findet sich in [Coh01]. Die Instanzen jedes Attributs sind als erschöpfend und sich gegenseitig ausschließend modelliert.

4.1 Qualitative Abstraktion der einfachen Objektattribute

Einfache Attribute beschreiben Eigenschaften eines Objekts, wie Höhe oder Geschwindigkeit. Diese werden anhand der Bounding-Box des Objekts bestimmt. Unter Bounding-Box wird hier ein Rechteck, welches alle sichtbaren Teile eines 3d-Objekts umschließt, verstanden.

Objektlängenverhältnis Das Attribut Objektlängenverhältnis (*aspectRatio*) charakterisiert die Form der Bounding-Box eines Objekts und wird aus dem Verhältnis der Höhe zu Breite der Bounding-Box berechnet. Um die Anzahl und Größe der qualitativen Wertebereiche dieses Attributs festzulegen, wurde eine Untersuchung durchgeführt. Dazu wurde das Längenverhältnis der Bounding-Boxen von etwa 3000 Objekten der Klassen Auto, Lastkraftwagen (LKW), Minibus, Motorradfahrer, Fahrradfahrer, Fußgänger, Verkehrsschild, Straßenschild, Ampel, Straßenlaterne, Baum und Gebäude ausgewertet und nach Klasse sortiert. Alle Objekte waren manuell annotiert und stammen aus eigenen Daten oder aus der Da-

tenbank LabelMe [Rus08]. Anhand der Ergebnisse dieser Untersuchung wurden 12 klassencharakteristische symbolische Wertebereiche festgelegt, die in Tabelle 4.1 gezeigt sind.

Tabelle 4.1: Ergebnisse der Untersuchung zur qualitativen Abstraktion des Merkmals Längenverhältnis l in [%] im Bezug auf die Objektklassen.

l [%]	<i>aspectRatio</i>	mögliche Objektklassen
$0 \leq l \leq 15$	<i>AR0_15</i>	Infrastrukturobjekt
$15 < l \leq 60$	<i>AR15_60</i>	Auto (Seite)
$60 < l \leq 90$	<i>AR60_90</i>	Auto (Vorne/Hinten), LKW, Van
$90 < l \leq 110$	<i>AR90_110</i>	LKW, Van
$110 < l \leq 140$	<i>AR110_140</i>	Radfahrer
$140 < l \leq 190$	<i>AR140_190</i>	Rad-, Motorradfahrer
$190 < l \leq 230$	<i>AR190_230</i>	Fußgänger, Rad-, Motorradfahrer
$230 < l \leq 260$	<i>AR230_260</i>	Fußgänger, Motorradfahrer
$260 < l \leq 320$	<i>AR260_320</i>	Fußgänger
$320 < l \leq 420$	<i>AR320_420</i>	Infrastrukturobjekt
$420 < l \leq 500$	<i>AR420_500</i>	Infrastrukturobjekt
$l > 500$	<i>AR500_</i>	Infrastrukturobjekt

Höhe über Grund Das Attribut Höhe über Grund (*heightAboveGround*) bezeichnet die Höhe über der befahrbaren Ebene der linken unteren Ecke der Bounding-Box des Objekts. Dieses wird in drei symbolische Werte unterteilt: auf der befahrbaren Ebene (*OnGround*), in der Luft bis 2m (*InAirRelevant*) und in der Luft höher 2m (*InAirIrrelevant*). Tabelle 4.2 veranschaulicht die qualitative Abstraktion dieses Merkmals.

Tabelle 4.2: Qualitative Abstraktion des Merkmals Höhe über Grund.

g [m]	<i>heightAboveGround</i>
$0 \leq g \leq 0,4$	<i>OnGround</i>
$0,4 < g \leq 2$	<i>InAirRelevant</i>
$g > 2$	<i>InAirIrrelevant</i>

Objekthöhe Das Merkmal Objekthöhe (*height*) bezeichnet die Höhe der Bounding-Box des Objekts und wird zweckmäßig in vier Wertebereiche unterteilt, wie in Tabelle 4.3 veranschaulicht.

Objektgeschwindigkeitsbetrag Das Attribut Objektgeschwindigkeit (*speed*) wird qualitativ in sechs Wertebereiche abstrahiert, wie in Tabelle 4.4 dargestellt. Der Bereich niedriger Geschwindigkeiten ist vergleichsweise feiner diskretisiert, da der Hauptfokus dieser Arbeit auf innerstädtischen Verkehrssituationen liegt.

Tabelle 4.3: Qualitative Abstraktion des Merkmals Objekthöhe.

h [m]	<i>height</i>
$0 \leq h \leq 1$	<i>Small</i>
$1 < h \leq 2,2$	<i>Average</i>
$2,2 < h \leq 3,5$	<i>Large</i>
$h > 3,5$	<i>VeryLarge</i>

Tabelle 4.4: Qualitative Abstraktion des Merkmals Geschwindigkeitsbetrag.

v [km/h]	<i>speed</i>
$0 \leq v \leq 0,1$	<i>Zero</i>
$0,1 < v \leq 10$	<i>VeryLow</i>
$10 < v \leq 30$	<i>Low</i>
$30 < v \leq 60$	<i>Medium</i>
$60 < v \leq 90$	<i>High</i>
$90 < v$	<i>VeryHigh</i>

Objektgeschwindigkeitsrichtung Die Objektgeschwindigkeitsrichtung wird als Winkel ψ aus der Sicht des Ego-Fahrzeugs bestimmt, wobei die Längsachse in Blickrichtung des Ego-Fahrzeugs die Referenzachse bildet. Abbildung 4.1 veranschaulicht die entsprechenden Winkel am Beispiel einer Szene mit drei Objekten. Die Abbildung zeigt bewusst lediglich das Ego-Fahrzeug als solches, da die anderen zwei 3d-Objekte beliebig sein dürfen und hier zu einem punktförmigen Objekt abstrahiert sind.

4.2 Qualitative Abstraktion der relativen Objektattribute

Die relativen Attribute beschreiben eine Relation zwischen zwei Objekten, wie beispielsweise die relative Position. Alle relativen Objektattribute aus der Ontologie werden für jedes mögliche Objektpaar des Diskursbereichs bestimmt. Das Ego-Fahrzeug wird dabei auch als Objekt der Szene betrachtet. Bei der Bestimmung der relativen Position und der relativen Geschwindigkeitsrichtung wird die

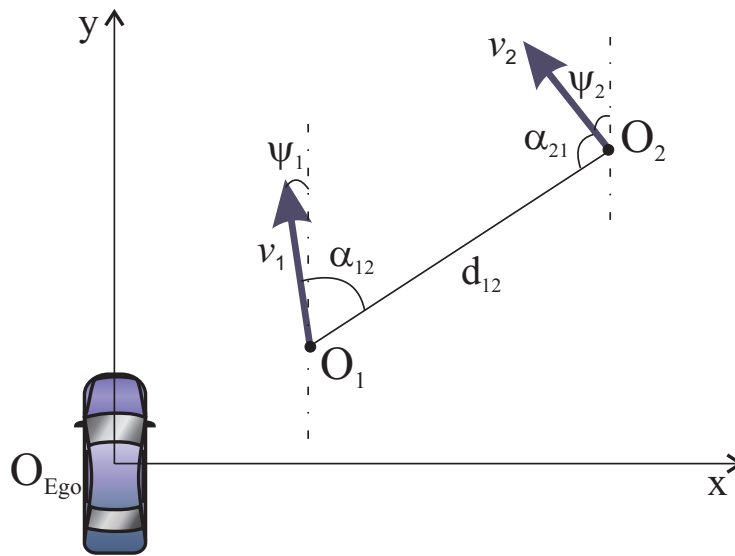


Abbildung 4.1: Beispielszene mit drei Verkehrsteilnehmern – das Ego-Fahrzeug O_{Ego} und zwei beliebige Objekte O_1 und O_2 . Die Längsachse in Blickrichtung des Ego-Fahrzeugs bildet die Referenzachse. Dargestellt sind die Geschwindigkeit v mit ihrer Richtung ψ , der relative Abstand d sowie die relative Position der beiden Objekte, definiert durch den Winkel α .

Bewegungsachse des Referenzobjekts als Bezugsachse betrachtet, so dass eine Koordinatentransformation notwendig ist.

Differenz der Geschwindigkeitsrichtungen Das relative Objektattribut Differenz der Geschwindigkeitsrichtungen (*difference*) zweier Objekte O_1 und O_2 wird anhand der Differenz $\Delta\psi = |\psi_1 - \psi_2|$ berechnet. Dabei sind ψ_1 und ψ_2 die jeweiligen Geschwindigkeitsrichtungen der Objekte entsprechend Abbildung 4.1 mit der Längsachse in Blickrichtung des Ego-Fahrzeugs als Bezugsachse. Die Differenz der Geschwindigkeitsrichtungen ist aufgrund des Betrags ein symmetrisches Attribut. Die qualitative Abstraktion der auf 180° normierten Differenz $\Delta\psi$ stellt Tabelle 4.5 dar. Der symbolische Wert *NoDataDiffInOrient* gilt, wenn mindestens eins der beiden Objekte steht.

Relative Geschwindigkeitsrichtung Bei der Bestimmung der relativen Geschwindigkeitsrichtung (*direction*) zweier Objekte wird im Unterschied zum Attribut Differenz der Geschwindigkeitsrichtungen als Bezugsachse die Bewegungsrichtung des jeweiligen Referenzobjekts verwendet. Dazu wird die Winkeldifferenz $\Delta\psi_{ij}^r$ aus der Geschwindigkeitsrichtung des Hauptobjekts j und der des Referenzobjekts i gebildet. Am Beispiel der Abbildung 4.1 beträgt die relative Ge-

Tabelle 4.5: Qualitative Abstraktion der Differenz der Geschwindigkeitsrichtungen $\Delta\psi$.

$\Delta\psi$ [°]	<i>difference</i>
$0 \leq \Delta\psi \leq 30$	<i>Equal</i>
$30 < \Delta\psi \leq 150$	<i>Crossing</i>
$150 < \Delta\psi \leq 180$	<i>Opposite</i>
–	<i>NoDataDiffInOrient</i>

schwindigkeitsrichtung des Objekts O_2 im Bezug auf Objekt O_1 $\Delta\psi_{12}^r = \psi_2 - \psi_1$. Die Differenz wird auf Werte zwischen 180° und -180° normiert und entsprechend Abbildung 4.2 qualitativ abstrahiert. Wenn mindestens eines der Objekte steht, kann die obige Differenz nicht berechnet werden. In solchen Fälle gilt der symbolische Wert *NoDataRelVelDir*.

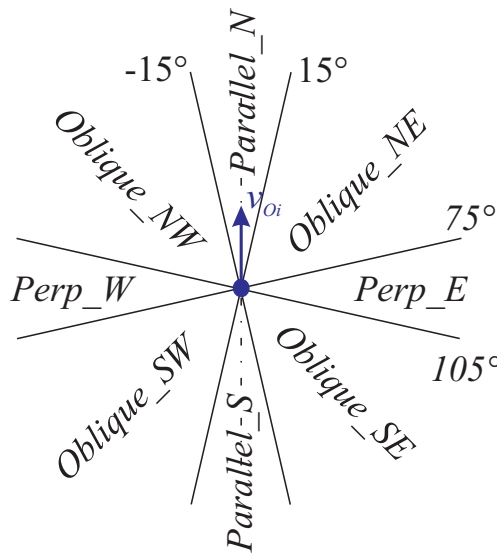


Abbildung 4.2: Diskretisierung der Winkeldifferenz $\Delta\psi_{ij}^r$ mit entsprechenden Instanzen der relativen Geschwindigkeitsrichtung für das Objekt O_i als Referenzobjekt. Die Bewegungsrichtung des Referenzobjekts v_{O_i} ist die Bezugsachse zur Bestimmung der relativen Geschwindigkeitsrichtung.

Relative Position Die relative Position zweier Objekte (*position*) wird durch den Winkel α_{ij} zwischen der Bewegungsrichtung des Referenzobjekts O_i und der Verbindungslinie zwischen dem Referenz- und dem Hauptobjekt bestimmt. In Abbildung 4.1 sind exemplarisch die Winkel α_{12} und α_{21} veranschaulicht. Der Winkel α_{12} gibt die relative Position von O_2 im Bezug auf O_1 an. In diesem Fall ist O_2 das Hauptobjekt, O_1 ist das Referenzobjekt. Umgekehrt bei α_{21} ist O_2 das Referenzobjekt und O_1 das Hauptobjekt, so dass in diesem Fall die Bewegungsachse von

O_2 als Bezugsachse verwendet wird. Der Winkel α wird für jedes mögliche Objektpaar der Szene berechnet, auf Werte zwischen 180° und -180° normiert und entsprechend Abbildung 4.3 qualitativ abstrahiert. Der Winkel α kann für zwei stehende Objekte oder bei einem stehenden Referenzobjekt und einem sich bewegendem Hauptobjekt nicht bestimmt werden. In solchen Fällen ist der symbolische Wert *NoDataRelPos* gültig.

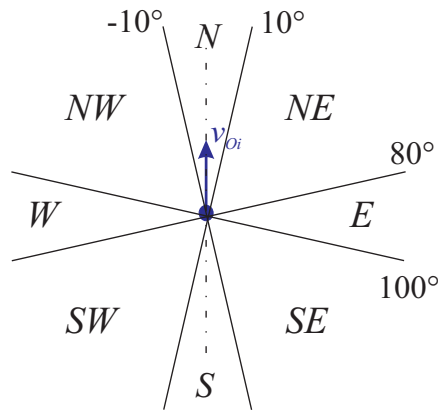


Abbildung 4.3: Diskretisierung des Winkels α_{ij} zur Bestimmung der relativen Position des Hauptobjekts O_j zum Referenzobjekt O_i mit den entsprechenden Instanzen. Die Bewegungsrichtung des Referenzobjekts v_{O_i} ist die Bezugsachse.

Relativer Abstand Der relative Abstand (*distance*) entspricht der euklidischen Distanz d_{ij} zweier Objekte, wie in Abbildung 4.1 visualisiert. Die qualitative Diskretisierung dieses Merkmals in fünf symbolischen Werte ist in der Abbildung 4.4 veranschaulicht.

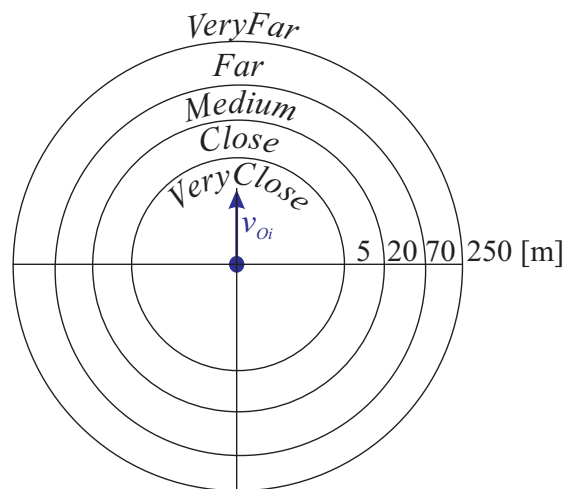


Abbildung 4.4: Qualitative Abstraktion des relativen Abstands d_{ij} .

5 Statistische Gütekriterien

Dieses Kapitel stellt die statistischen Gütekriterien vor, die bei der quantitativen Auswertung der Inferenzergebnisse verwendet werden. Diese Qualitätsmaße werden sowohl für die Klassifikation anhand simulierter Daten, als auch für die Klassifikation anhand experimentell gewonnener Daten berechnet.

Es gibt viele Gütekriterien, die bei der Bewertung der Leistung eines Klassifikators in Frage kommen. Bei verschiedenen Forschungsgebieten sind unterschiedliche Kriterien beliebt. Darüber hinaus beeinflusst die Art des Klassifikators, ob zwei- oder mehrklassig, ob eine Ja/Nein oder eine reellwertige Ausgabe, die Wahl eines Gütemaßes.

Bei Ja/Nein-Klassifikatoren oder für eine konkrete Unterscheidungsschwelle bei einem reellwertigen Klassifikator lassen sich Wahrheitsmatrizen aufstellen [Gör03]. In eine Wahrheitsmatrix werden die Entscheidungen einer binären Klassifikation in vier Möglichkeiten eingeteilt: *richtig positiv (RP)* (engl. *true positive*), für ein als korrekt positiv klassifiziertes Objekt; *falsch positiv (FP)* (engl. *false positive*), für ein fälschlich als positiv klassifiziertes Objekt; *richtig negativ (RN)* (engl. *true negative*), für ein korrekt als negativ klassifiziertes Objekt und *falsch negativ (FN)* (engl. *false negative*), für ein fälschlich als negativ klassifiziertes Objekt. Für Klassifikationen mit mehr als zwei Klassen werden in der Wahrheitmatrix Häufigkeiten lediglich für richtig positive und falsch negative Entscheidungen eingetragen [Abu10], was in Abbildung 5.1 veranschaulicht ist.

Wahrheit	Schätzung		Ges.	Wahrheit	Schätzung				Ges.
	c^+	c^-			c_1	c_2	\dots	c_n	
c^+	RP	FN	E^+	c_1	RP_1	FN_1	\dots	FN_1	E_1
c^-	FP	RN	E^-	c_2	FN_2	RP_2	\dots	FN_2	E_2
Ges.	\hat{E}^+	\hat{E}^-	E	\dots	\dots	\dots	\dots	\dots	\dots
				c_n	FN_n	FN_n	\dots	RP_n	E_n
				Ges.	\hat{E}_1	\hat{E}_2	\dots	\hat{E}_n	E

Abbildung 5.1: Wahrheitsmatrix für eine Klassifikation mit zwei (links) oder mehr als zwei (rechts) Objektklassen.

Anhand der Häufigkeiten in einer Wahrheitsmatrix lassen sich Qualitätsmaße zur Klassifikationsbewertung, wie Richtig-Positiv-Rate oder auch Sensitivität (der An-

teil der richtig als positiv klassifizierten Objekte bezogen auf alle tatsächlich positiven Objekte), Richtig-Negativ-Rate oder auch Spezifität (der Anteil der richtig als negativ klassifizierten Objekte bezogen auf alle in Wirklichkeit negativen Objekte), Genauigkeit und kombinierte Maße, wie z. B. das F-Maß (engl. *Fscore*) berechnen. Für einen Überblick der gängigen Kriterien zur Beurteilung der Güte eines Klassifikators wird auf [Sok09] verwiesen.

Für die quantitative Auswertung der Inferenzergebnisse werden in der vorliegenden Arbeit die Gütekriterien Mehrklassen-Genauigkeit (engl. *multi-class accuracy*), Fläche unter der ROC-Kurve für Mehrklassen-Probleme (engl. *multi-class AUC*) und die mittlere bedingte logarithmische Likelihood (engl. *conditional log-likelihood*) verwendet. Für die bedingte Log-Likelihood wird im Weiteren die Abkürzung des englischen Begriffs CLL als Bezeichnung verwendet. Für die Fläche unter der ROC-Kurve wird ebenfalls die englische Abkürzung AUC (engl. *area under curve*) genutzt. Die Mehrklassen-Genauigkeit ist ein robustes Maß für die Gesamtleistung eines Mehrklassen-Klassifikators [Sok09]. Die Mehrklassen-AUC wird für Klassifikation mit mehr als zwei Klassen bestimmt und ist unempfindlich gegenüber einer großen Anzahl von richtig negativen Entscheidungen [Bib09]. Die mittlere CLL misst direkt die Qualität der inferierten Wahrscheinlichkeiten eines reellwertigen Klassifikators [Dom09]. Die folgenden drei Kapitel beschreiben die Berechnung der ausgewählten Gütemaße zur Beurteilung der Klassifikationsleistung.

5.1 Mehrklassen-Genauigkeit

Die Genauigkeit (engl. *accuracy*) wird in Veröffentlichungen aus dem Bereich des maschinellen Lernens oft als Kriterium zum Vergleichen der Klassifikationsleistung verschiedener Algorithmen verwendet. Die Genauigkeit eines Modells kann interpretiert werden als der Erwartungswert einer korrekten Klassifikation für ein beliebiges Beispiel [Abu10].

Die Mehrklassen-Genauigkeit wird von der Formel für binäre Genauigkeit abgeleitet [Abu10] und lautet:

$$\text{Acc} = \sum_{i=1}^n \frac{RP_i}{E} = \sum_{i=1}^n \frac{E_i}{E} \frac{RP_i}{E_i}, \quad (5.1)$$

wobei die Bezeichnungen RP_i und E denjenigen aus der Wahrheitsmatrix in Abbildung 5.1 entsprechen. Die Mehrklassen-Genauigkeit ist somit der mit dem Klassenprior gewichtete Anteil der durch das Modell korrekt klassifizierten Objekte der i -ten Klasse. Die Genauigkeit liefert eine aussagekräftige Vergleichsgröße, nur

dann, wenn sie für die gleiche Gesamtanzahl der Objekte in der Testmenge E berechnet wird.

5.2 Mehrklassen-AUC

Wird die Unterscheidungsschwelle bei einem binären probabilistischen Klassifikator variiert, lässt sich die Klassifikationsleistung durch eine ROC-Kurve (engl. *Receiver Operating Characteristic*) graphisch veranschaulichen [Faw06]. Auf der Abszisse wird die Falsch-Positiv-Rate und auf der Ordinate die Richtig-Positiv-Rate für jeden möglichen Schwellwert aufgetragen. Die resultierende angenäherte ROC-Kurve ist typischerweise gekrümmt aufsteigend. Die Fläche unter dieser Kurve (engl. *Area Under Curve*) ist ein skalares Gütemaß für die Klassifikationsleistung. Die AUC beträgt zwischen 0 und 1. Für eine aussagekräftige Klassifikation sollte die AUC einen Wert zwischen 0.5 und 1 annehmen. Eine zufallsgesteuerte Klassifikation entspricht der Diagonalen zwischen den Punkten (0,0) und (1,1) der ROC-Kurve.

Das Heranziehen der ROC-Kurve zur Beurteilung von Klassifikationsaufgaben mit mehr als zwei Klassen stellt ein nichttriviales Problem dar [Lac03]. In der Literatur wird als Gütekriterium bei Mehrklassen-Probleme die Mehrklassen-AUC verwendet [Abu10]. Es wird für eine gegebene Testmenge von Objekten pro Klasse eine AUC berechnet, wobei die aktuell betrachtete Klasse als positiv und die restlichen Objektklassen als negativ definiert werden (engl. *one-vs-rest approach*). Die Mehrklassen-AUC für die Testmenge ist die mit dem Klassenprior gewichtete Summe dieser AUCs. Die AUC ist äquivalent zu der Wahrscheinlichkeit, dass durch den Klassifikator eine zufällig ausgewählte positive Instanz höher bewertet wird wie eine zufällig ausgewählte negative Instanz [Faw06].

5.3 Bedingte Log-Likelihood (CLL)

Bei der Inferenz in graphischen Modellen wird die bedingte Wahrscheinlichkeitsverteilung der Anfragevariablen, gegeben einer Zuweisung der Evidenzvariablen berechnet. Das belegte Markov-Netz $G_{\mathbf{L},\mathbf{K}}$ wird anhand des entsprechenden MLNs \mathbf{L} und der Evidenzdaten erstellt. Die Menge der Konstanten \mathbf{K} wird durch die aktuelle Evidenz vorgegeben. Als Ergebnis der Inferenz wird für jedes Grundatom X_k , $k = 1, \dots, n$, des Anfrageprädikats Q die Wahrscheinlichkeit $P_{X_k} = P(X_k = 1|Q, G_{\mathbf{L},\mathbf{K}})$ dafür berechnet, dass die durch X_k gemachte Aussage korrekt ist. Gegeben eine binäre Referenzdatei $t_Q = (t_{X_1}, \dots, t_{X_n})$, die die

Richtigkeit der Aussagen der Grundatome angibt, kann die CLL von jedem Grundatom des Abfrageprädikats Q bestimmt werden. Die CLL eines Grundatoms X_k beträgt $\log P_{X_k}$, wenn $t_{X_k} = 1$ und $\log(1 - P_{X_k})$, wenn $t_{X_k} = 0$ [Elk12]. Es lässt sich somit pro Abfrageprädikat ein Vektor

$$CLL_Q = (CLL_{X_1}, \dots, CLL_{X_n})$$

erstellen, dessen Mittelwert $CLLm_Q$ und Standardabweichung $CLLstd_Q$ ein Maß für die Güte der inferierten Wahrscheinlichkeit für dieses Abfrageprädikat bei gegebenen Evidenzdaten sind [Bib09]. Die CLL ist eine negative logarithmische Größe und somit um so besser je größer der Wert, d. h. je näher die Zahl an der Null ist.

Damit ein Qualitätsmaß pro Model für eine bestimmte Testmenge vorliegt, wird über alle Anfrageprädikate, z. B. für die acht Objektklassen, gemittelt, so dass ein Mittelwert der mittleren CLL und ein Mittelwert der Standardabweichungen der CLL der einzelnen Anfrageprädikate berechnet werden.

6 Lernen der Modellparameter

Die Gewichte zu den Regeln der im vorherigen Kapitel vorgestellten Markov-Logik Modelle werden mit Hilfe eines Parameterlernverfahrens geschätzt. Während der Trainingsphase wird zu jeder Regel ein Gewicht gelernt, das dem Wahrheitsgehalt dieser Regel bezüglich der Trainingsdaten entspricht. Im Folgenden werden die Trainingsdatenbank und das Parameterlernen für die einzelnen Modelle näher erläutert.

6.1 Trainingsdaten

Die Trainingsdaten bestehen aus Grundatomen. Sie sind aus 672 manuell segmentierten Objekten aus 100 Bildern von Verkehrsszenen in der Stadt, auf Autobahnen und Landstraßen entstanden. Bei der Auswahl der Bilder wurde auf Vielfältigkeit geachtet, so dass möglichst viele verschiedene Verkehrsteilnehmer und Verkehrssituationen darin vorkommen. Tabelle 6.1 gibt einen Überblick über die Klassen der gelabelten Objekte in den Daten.

Tabelle 6.1: Klasse und Anzahl der Objekte der Trainingsdaten.

Objektklasse	Anzahl
<i>car</i>	217
<i>van</i>	20
<i>utilityVehicle</i>	9
<i>motorcyclist</i>	5
<i>bicyclist</i>	31
<i>pedestrian</i>	65
<i>infrastrObject</i>	260
<i>inAirIrrelObject</i>	65

Die Objekte kommen in unterschiedlichsten Ansichten vor. Auch nur teilweise sichtbare Objekte wurden mit der entsprechenden Klasse annotiert. Für alle segmentierten Objekte wurden die wahre Objektklasse, die korrekten Ausprägungen für jede Attributrelation und die im Bild bestehenden Objektrelationen angegeben.

Dabei wurden beim Annotieren nur die Konzepte der tiefsten Spezialisierungsebene der Objektklassen verwendet. Das Objektlängenverhältnis wurde anhand der manuellen konturgenauen Segmentierung berechnet. Alle anderen Angaben wurden aus der gesamten Verkehrssituation im Bild abgeschätzt. Der Bewegungszustand der Objekte wurde anhand der Sequenz, aus der das Bild stammt, bestimmt.

Aus den so generierten Daten wurden anschließend automatisiert die entsprechenden Grundatome erzeugt und in eine Trainingsdatei geschrieben. Abbildung 6.1 zeigt beispielhaft ein segmentiertes Trainingsbild. Tabelle 6.2 enthält eine (nicht vollständige) Liste der aus diesem Bild generierten Trainingsdaten. Über die Diskretisierungsstufen der einzelnen Objektattribute wurde in Kapitel 4 ausführlich berichtet.

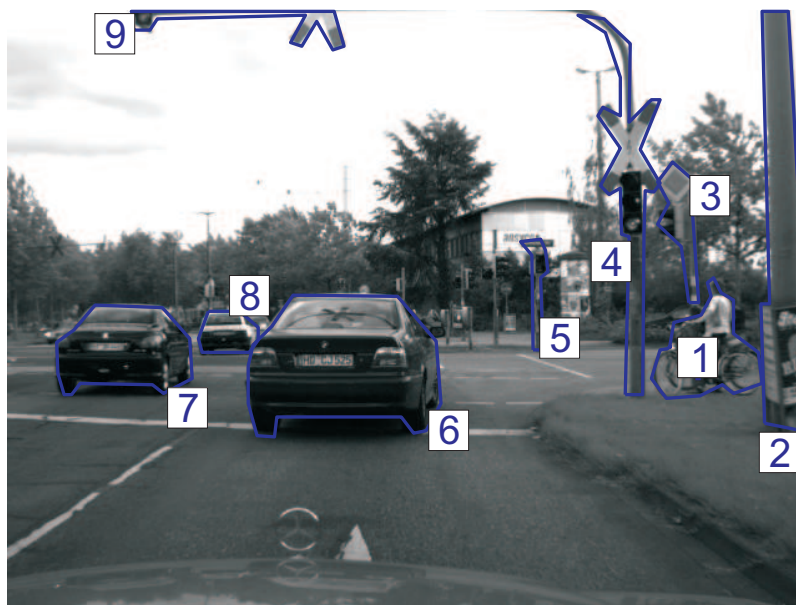


Abbildung 6.1: Manuell segmentiertes Trainingsbild Nr. 51.

Tabelle 6.2: Auszug aus den Trainingsdaten vom Bild Nr. 51.

<i>bicyclist(O1)</i>	<i>hasRelPos(O6,Ego,S)</i>
<i>hasHeight(O1,Average)</i>	<i>hasRelPos(O6,O1,E)</i>
<i>hasHeightAboveGround(O1,OnGround)</i>	<i>hasDiffInOrient(Ego,O6,Equal)</i>
<i>hasAspectRatio(O1,AR90_110)</i>	<i>hasDistance(O1,O6,Close)</i>
<i>hasSpeed(O1,Zero)</i>	<i>follow(O7,O6)</i>
<i>car(O6)</i>	<i>follow(O8,O6)</i>
<i>car(O7)</i>	<i>follow(O8,O7)</i>
<i>car(O8)</i>	<i>movePast(O1,O6)</i>
<i>infrastrObject(O2)</i>	<i>moveAwayFrom(O1,O7)</i>
<i>hasRelPos(Ego,O6,N)</i>	<i>moveAwayFrom(O1,O8)</i>

Die Trainingsdaten enthalten alle Grundatome von allen Prädikaten aus den Regeln der im vorigen Kapitel vorgestellten Modelle. Beim Parameterlernen wird für jedes MLN eine entsprechende Untermenge der Trainingsdaten verwendet.

6.2 Vergleichsuntersuchung der Lernverfahren

Während des Parameterlernens wird jede Regel in Konjunktive Normalform (siehe Kapitel 2.1.3) umgewandelt und zu jeder der Klauseln wird ein Gewicht gelernt [Dom09]. Das Gewicht spiegelt die Häufigkeit wider, mit der diese Klausel in den Trainingsdaten erfüllt wurde. Beispielsweise wurden folgende Gewichte für die symbolische Ausprägung des Objektlängenverhältnisses *AR60_90* gelernt:

- +2.4 $hasAspectRatio(o, AR60_90) \Rightarrow car(o)$
- 3.6 $hasAspectRatio(o, AR60_90) \Rightarrow pedestrian(o)$.

Aus der Größe der Gewichte lässt sich schließen, dass diese bestimmte qualitative Formbeschreibung für ein Auto, aber nicht für einen Fußgänger charakteristisch ist. Das Zusammenspiel der Gewichte bei einer nachfolgenden Inferenz ist allerdings schon bei wenigen Formel so komplex, dass ein eindeutiges Nachvollziehen der Inferenzergebnisse anhand der Gewichte praktisch unmöglich ist [Dom06].

In der Literatur wird üblicherweise diskriminativen Parameterlernverfahren eine höhere Klassifikationsleistung attestiert [Rus03]. Im Gegensatz zu den generativen Lernverfahren, die die Verbundwahrscheinlichkeit aller Zufallsvariablen eines Modells optimieren, maximieren die diskriminativen die bedingte Likelihood der Menge der Anfragevariablen gegeben der Menge der Eingangsdaten. Die Aufgabe der in dieser Arbeit betrachteten MLN-Modelle kann als eine Klassifikationsaufgabe aufgefasst werden. Aus diesem Grund werden im Folgenden hauptsächlich die diskriminativen Parameterlernverfahren aus Kapitel 2.3.3.2 untersucht. Ergebnisse generativer Lernalgorithmen werden lediglich aus Gründen der Vollständigkeit veranschaulicht.

Die Ergebnisdatei des Parameterlernens für ein Modell enthält alle Regeln und die zugehörigen gelernten Gewichte des Modells. Folgende Bezeichnungen werden eingeführt: TR-OK für die Ergebnisdatei des Modells „Objektklassifikation“, TR-ROK für die Ergebnisdatei des Modells „Relationale Objektklassifikation“ und TR-IR für das Modell „Inferierte Objektrelationen“. Als Ergänzung bekommen diese Bezeichnungen gegebenenfalls eine Abkürzung des verwendeten Trainingsalgorithmus und/oder der jeweiligen Iterationsanzahl.

Die in Kapitel 2.3.3.2 eingeführten diskriminativen Lernverfahren für MLN *Voted Perceptron* (VP) [Sin05], *Diagonal Newton* (DN) [Dom09] und *Preconditioned Scaled Conjugate Gradient* (PSCG) [Low07] wurden anhand der Regeln des Modells MLN „Relationale Objektklassifikation“, vorgestellt in Kapitel 3.3.2, untersucht. Dabei wurden die im vorherigen Kapitel beschriebenen Trainingsdaten verwendet. Als Anfragevariablen wurden bei allen drei diskriminativen Lernverfahren die acht Objektklassen der tiefsten Taxonomieebene angegeben - *car*, *van*, *utilityVehicle*, *motorcyclist*, *bicyclist*, *pedestrian*, *infrastrObject* und *inAirIrrelObject*. Diese wurden auch bei der anschließenden Inferenz als Anfrage genutzt. Es wurde jeweils über 100 Iterationen trainiert. Zusätzlich wurden die Regeln des Modells MLN „Relationale Objektklassifikation“ für die Dauer von 100 Iterationen generativ trainiert und mit den oben erwähnten Anfragevariablen inferiert. Bei den experimentellen Untersuchungen wurde die frei verfügbare Plattform *Alchemy* [Kok06] eingesetzt. Tabelle 6.3 veranschaulicht für jedes der vier Verfahren die Dauer der Trainingsphase in Minuten für TR-ROK. Das Parameterlernen wurde auf einem Laptop Intel Core 2 mit 2,40GHz Taktfrequenz und 4GByte RAM durchgeführt.

Tabelle 6.3: Dauer der Trainingsphase in [min] von TR-ROK für die vier untersuchten Lernalgorithmen über 100 Iterationen.

	Lerndauer [min]				Anzahl Grundatome	Anzahl Grundklauseln
	VP	DN	PSCG	Gen		
TR-ROK	18,56	16,39	16,18	20,12	5054	6161

Eine Ergebnisdatei des Parameterlernens zusammen mit den dazugehörigen Axiomen des Modells definieren ein Markov-Logik-Netz, welches zur Klassifikation verwendet werden kann (s. Def. 2.3.1). Zusammen mit einer Menge von Evidenzdaten lässt sich ein belegtes Markov-Netz erstellen und Wahrscheinlichkeiten für eine Menge von Anfragevariablen inferieren. An dieser Stelle dient die Inferenz der Bewertung der Güte der mit den verschiedenen Algorithmen zum Parameterlernen bestimmten Gewichte. Aufgrund dieses Vergleichs ist ein Parameterlernverfahren auszuwählen, mit dem anschließend die Regeln aller zu untersuchenden Modelle trainiert werden.

Die aus dem Training über 100 Iterationen resultierenden Dateien TR-ROK VP, TR-ROK DN, TR-ROK PSCG und TR-ROK Gen werden durch die Axiome des Modells „Relationale Objektklassifikation“ ergänzt und definieren so die vier Markov-Logik-Netze - MLN ROK VP, MLN ROK DN, MLN ROK PSCG und MLN ROK Gen, die zur Inferenz genutzt werden. Dabei werden die simulierten Evidenzdaten aus Kapitel 7.1.2 eingesetzt. Zur Inferenz wird der MC-SAT Algo-

rithmus verwendet, welcher den MCMC-Algorithmus mit einem SAT-Solver kombiniert (siehe Kapitel 2.3.2). Die vier Markov-Logik-Netze wurden für drei unterschiedliche Iterationsschritte des Gibbs-Samplers inferiert. Als Qualitätsmaße für die Inferenzergebnisse dienen der Mittelwert der mittleren CLL der acht Anfrageprädikate, der Mittelwert der Standardabweichungen der CLL der acht Anfrageprädikate und die Mehrklassen-AUC für die acht Anfrageprädikate. Diese Maße werden pro Testbild berechnet und anschließend über alle 5 Testbilder gemittelt. Die Abbildungen 6.2-6.4 zeigen die Ergebnisse der vier Markov-Logik-Netze für jedes der drei Gütekriterien, wobei jede der Legenden die auf die Lernverfahren abgekürzten Namen der vier MLN auflistet.

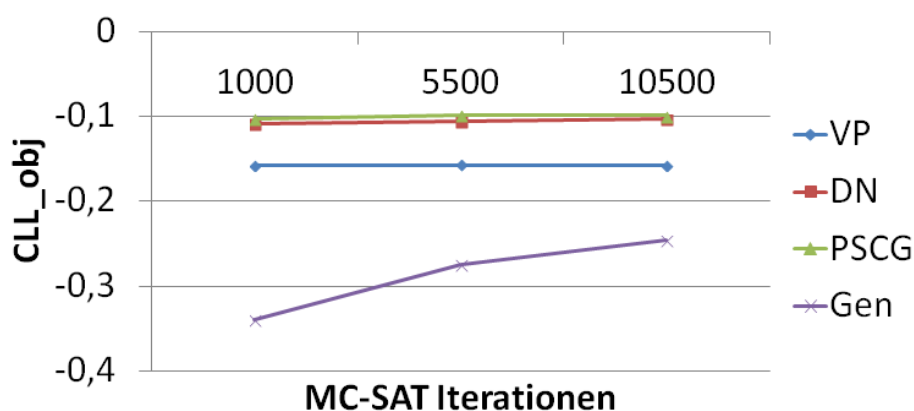


Abbildung 6.2: Mittelwert der mittleren CLL der acht Anfrageprädikate.

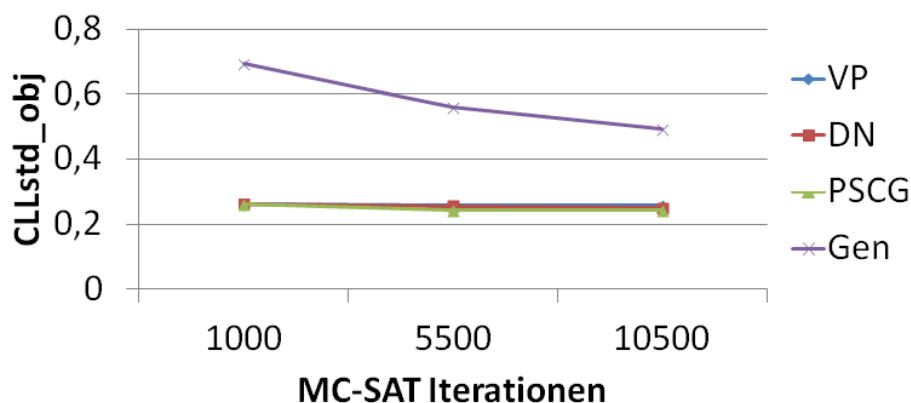


Abbildung 6.3: Mittelwert der Standardabweichung der CLL der acht Anfrageprädikate.

Das generative Parameterlernverfahren schneidet, wie erwartet, bei allen Qualitätsmaßen deutlich schlechter ab, als die diskriminativen Lernalgorithmen, welche eine im allgemeinen vergleichbare Leistung erzielen. MLN ROK PSCG liefert allerdings die besten Ergebnisse. Für die weiteren Untersuchungen der Modelle

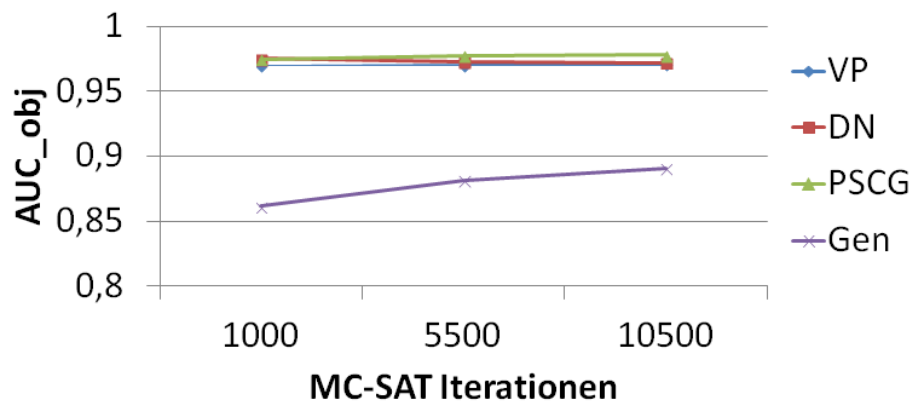


Abbildung 6.4: Mittlere Mehrklassen-AUC der acht Anfrageprädikate.

in dieser Arbeit wird deshalb das diskriminative konjugierte Gradientenverfahren PSCG eingesetzt.

Mit dem ausgewählten Parameterlernverfahren PSCG wurden anschließend die Regeln des Modells „Objektklassifikation“ und des Modells „Relationale Objektklassifikation“ jeweils für 50, 100 und 200 Iterationen trainiert. Die resultierenden Dateien sind entsprechend TR-OK50, TR-OK100, TR-OK200 und TR-ROK50, TR-ROK100, TR-ROK200 benannt. Die Regeln des Modells „Inferierte Objektrelationen“ wurden ebenfalls mit dem PSCG-Algorithmus für 100 Iterationen trainiert.

Im nächsten Kapitel werden diese Ergebnisdateien als inhaltliche Bausteine von mehreren Markov-Logik-Netze verwendet. Inferenzergebnisse der Modelle anhand von simulierten und experimentell gewonnenen Daten werden vorgestellt und diskutiert.

7 Inferenzergebnisse

Dieses Kapitel präsentiert Ergebnisse für die Inferenz einer relationalen Situationsbeschreibung anhand simulierter und experimentell gewonnener Daten. Abbildung 7.1 gibt einen Überblick über das entwickelte, auf Markov-Logik-Netzen basierende, System. Der verwendete Formalismus bedingt zunächst eine Überführung der vorhandenen Information über konkrete Objekte in einer zu analysierenden Verkehrsszene von der Signal- auf die Symbolebene. Dazu werden die entsprechenden Evidenzen, je nachdem ob mit simulierten oder realen Daten gearbeitet wird, entweder als logische Konstanten manuell vorgegeben oder aus den quantitativen Sensordaten mittels einer qualitativen Abstraktion (s. Kapitel 4) ermittelt. Die logischen Konstanten repräsentieren Objekte des Diskursbereichs (z. B. *O1*) oder Instanzen von Objektattributen (z. B. *OnGround*, *NW*). Diese Konstanten dienen zur Erstellung einer Menge von wahren Grundatomen (z. B. *sceneObject(O1)*, *hasRelPos(O1,O2,NW)*), die als Evidenzdaten bezeichnet werden und als Eingangsdaten zusammen mit dem MLN-Modell für die Inferenz genutzt werden, wie in Abbildung 7.1 veranschaulicht.

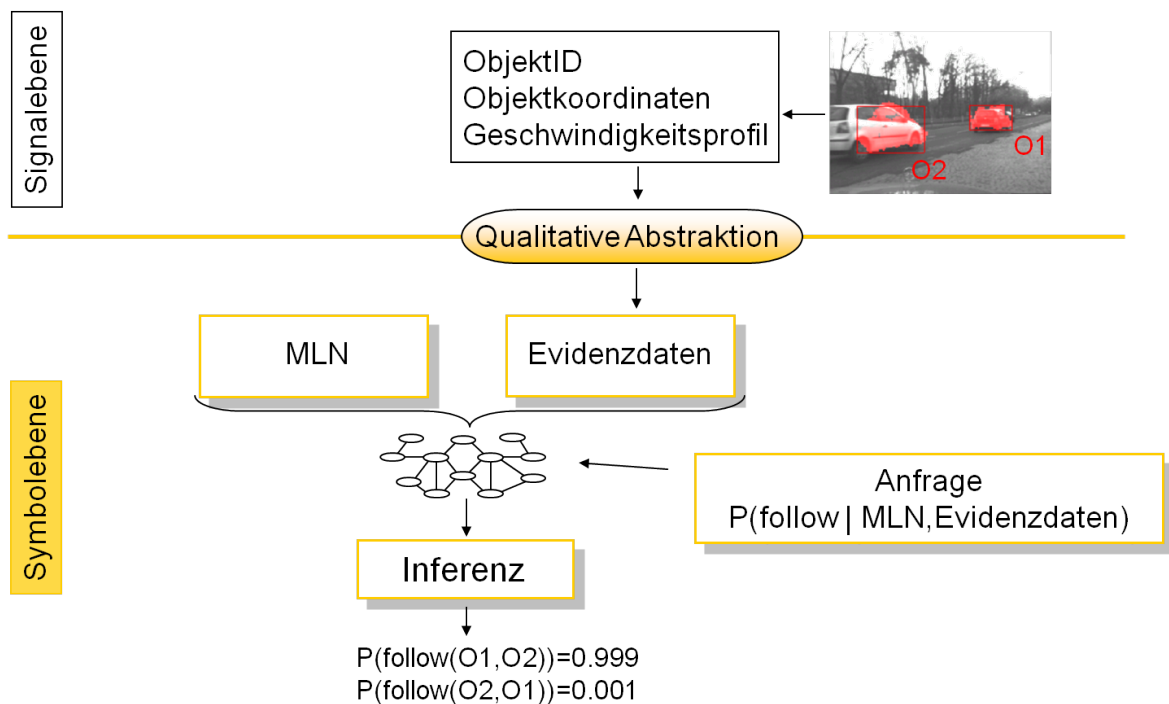


Abbildung 7.1: Systemüberblick für die Inferenz mit Markov-Logik-Netzen.

In Kapitel 3 wurden mehrere ontologiebasierte Modelle zur Klassifikation von Objekten oder Objektrelationen einer Verkehrssituation vorgestellt. Außer dem Modell „Direkte Objektrelationen“, welches mit Hilfe einer Definitionsmatrix deterministisch spezifiziert wird, bestehen alle aus prädikatenlogischen Formeln, die in Axiome und Regeln unterteilt sind. Mit Hilfe eines Parameterlernverfahrens werden Gewichte zu den Regeln der Modelle, wie in Kapitel 6 erläutert, geschätzt. Durch Kombination der Axiome und gewichteten Regeln der Modelle werden mehrere Markov-Logik-Netze erstellt, die zur Inferenz der Objektklassen und Objektrelationen in der Verkehrsszene eingesetzt werden. Eine detaillierte Beschreibung der untersuchten Markov-Logik-Netze findet sich in den Kapiteln 7.1.3 und 7.2.3.

Jedes MLN, gemeinsam mit den Evidenzdaten für ein Bild, wird automatisiert in ein belegtes Markov-Netz übersetzt, an welches geeignete Anfragen gestellt werden können. Mittels Inferenz anhand des belegten Markov-Netzes werden bedingte Wahrscheinlichkeiten für alle Grundatome der Anfrageprädikate geschätzt. Gegeben, zum Beispiel, das Anfrageprädikat *follow* und die Konstantenmenge $\{O1, O2\}$ werden als Ergebnis die Wahrscheinlichkeiten $P(\text{follow}(O1, O1))$, $P(\text{follow}(O1, O2))$, $P(\text{follow}(O2, O1))$ und $P(\text{follow}(O2, O2))$ geschätzt. Diese Wahrscheinlichkeiten liefern den geschätzten Wahrheitsgrad der Aussage, die durch das jeweilige Grundatom gemacht wird (z. B. die Wahrscheinlichkeit, dass das Objekt *O1* das Objekt *O2* folgt). Die Inferenz wird für alle Bilder nacheinander durchgeführt. Für die Inferenzergebnisse werden je Testbildmenge jedes der Gütekriterien aus Kapitel 4 berechnet. Die Inferenz wird mit dem MC-SAT Algorithmus durchgeführt, welcher besonders für Markov-Logik-Netze geeignet ist, die sowohl gewichtete Regeln als auch Axiome enthalten [Poo06].

7.1 Inferenz anhand simulierter Daten

Dieses Kapitel stellt das MLN OK, welches auf das Modell „Objektklassifikation“ basiert, und das MLN ROK, welches auf das Modell „Relationale Objektklassifikation“ basiert, vor und evaluiert ihre Objektklassifikationsleistung anhand einer Menge manuell gelabelten Objekten in Bildern.

7.1.1 Manuell annotierte Testdaten

Zum Test der verschiedenen Klassifikationsmodelle wurden in 5 Bildern, die nicht zum Trainieren verwendet wurden, insgesamt 60 3D-Objekte manuell annotiert.

Die Testbilder stellen Verkehrsszenen von Innenstadtkreuzungen und der Autobahn dar. Abbildung 7.2 veranschaulicht exemplarisch Testbild 1 mit den, wie bei den Trainingsdaten, konturgenau annotierten Objekten. Die minimale Anzahl manuell segmentierter 3D-Objekte in den Testbildern ist 7 und die maximale 17. Es



Abbildung 7.2: Testbild 1 mit den manuell segmentierten 3D-Objekten.

gibt insgesamt 20 Kraftfahrzeuge, 11 nichtmotorisierte Verkehrsteilnehmer, 20 Infrastrukturobjekte und 9 Objekte in großer Höhe über Grund. Eine detailliertere Aufschlüsselung der Objektklassen ist in Tabelle 7.1 zu finden.

Tabelle 7.1: Klasse und Anzahl der manuell annotierten 3D-Objekte der Testbilder.

Objektklasse	Anzahl
<i>car</i>	17
<i>van</i>	1
<i>utilityVehicle</i>	2
<i>motorcyclist</i>	0
<i>bicyclist</i>	3
<i>pedestrian</i>	8
<i>infrastrObject</i>	20
<i>inAirIrrelObject</i>	9

Für alle manuell segmentierten 3D-Objekte wurden die wahre Objektklasse, die korrekten Ausprägungen der einfachen Objektattribute und die im Bild beste-

henden Objektrelationen angegeben. Beim Annotieren der Klasse wurden für jedes Objekt das allgemeinste 3D-Konzept der Ontologie *sceneObject* sowie die Konzepte der tiefsten Spezialisierungsebene der Objektklassentaxonomie (siehe Bild 3.1) verwendet. Ähnlich wie bei der Generierung der Trainingsdaten wurde das Objektlängenverhältnis anhand der manuellen, konturgenauen Segmentierung berechnet. Die restlichen Angaben wurden aus der gesamten Verkehrssituation im Bild abgeschätzt. Der Bewegungszustand der Objekte wurde anhand der zugehörigen Videosequenz bestimmt. Die Objektrelationen wurden erschöpfend für alle möglichen Objektpaare im Bild annotiert.

7.1.2 Evidenzdaten

Aus den so generierten Testdaten wurden anschließend zu jedem Testbild automatisch die Evidenzdaten bestehend aus wahren Grundatomen erzeugt. Ein Auszug der aus dem Bild in Abbildung 7.2 generierten Evidenzdaten ist in Tabelle 7.2 veranschaulicht. Zu beachten ist, dass für alle Objekte die korrekten spezifischen

Tabelle 7.2: Auszug aus den Evidenzdaten für Testbild 1. Die korrekten spezifischen Klassen der 3D-Objekte sind, bis auf die des Ego-Fahrzeugs, auskommentiert. Die Objektbenennungen sind hier aus Gründen der Übersicht vereinfacht dargestellt.

<i>road(Pl_r)</i>	// <i>pedestrian(O1)</i>
<i>roadside(Pl_rs)</i>	<i>sceneObject(O1)</i>
	<i>on(O1,Pl_rs)</i>
// <i>Ego(O0)</i>	<i>hasSpeed(O1,Zero)</i>
<i>car(O0)</i>	<i>hasHeightAboveGround(O1,OnGround)</i>
<i>sceneObject(O0)</i>	<i>hasHeight(O1,Average)</i>
<i>on(O0,Pl_r)</i>	<i>hasAspectRatio(O1,AR230_260)</i>
<i>hasSpeed(O0,Medium)</i>	<i>noRelMotion(O1,O2)</i>
...	<i>noRelMotion(O2,O1)</i>
<i>follow(O4,O0)</i>	<i>movePast(O1,O4)</i>
<i>precede(O0,O4)</i>	<i>movePast_rev(O4,O1)</i>
<i>flank(O5,O0)</i>	<i>moveTowards(O1,O5)</i>
<i>flank(O0,O5)</i>	<i>moveTowards_rev(O5,O1)</i>
...	...

Klassen (z. B. *pedestrian*), bis auf die des Ego-Fahrzeugs, auskommentiert sind. Somit können die simulierten Evidenzdaten als Eingangsdaten für die Inferenz der Klassen der Szenenobjekte dienen.

7.1.3 Methodik

Anhand der simulierten Eingangsdaten wird die Performanz zweier Markov-Logik-Netze zur Objektklassifikation untersucht. Das erste modelliert die Objektklassen in Abhängigkeit von den einfachen Objektattributen, wie im Modell „Objektklassifikation“ in Kapitel 3.2 beschrieben, und wird im Folgenden als MLN OK bezeichnet. Das zweite Markov-Logik-Netz setzt darüber hinaus die Objektklassen mit den Objektrelationen der Ontologie in Beziehung. Dieses MLN basiert auf dem Modell „Relationale Objektklassifikation“ aus Kapitel 3.3 und wird im Weiteren als MLN ROK bezeichnet. Tabelle 7.3 veranschaulicht die inhaltlichen Bausteine der zwei Markov-Logik-Netze. Das MLN OK enthält die Axiome (AX-OK) und die gewichteten Regeln (TR-OK) des Modells „Objektklassifikation“. Das MLN ROK erweitert das Basismodell um die Axiome (AX-ROK) und die gewichteten Regeln (TR-ROK) des Modells „Relationale Objektklassifikation“.

Tabelle 7.3: Inhalte von MLN OK und MLN ROK.

	MLN OK	MLN ROK
AX-OK	✓	✓
TR-OK	✓	✓
AX-ROK		✓
TR-ROK		✓

Wie in Kapitel 6.2 beschrieben, wurden die Gewichte der Regeln der Modelle „Objektklassifikation“ und „Relationale Objektklassifikation“ mit dem diskriminativen konjugierten Gradientenverfahren PSCG [Low07] jeweils für die Dauer von 50, 100 und 200 Iterationen gelernt. Die resultierenden Parametersätze wurden als TR-OK50, TR-OK100, TR-OK200 sowie TR-ROK50, TR-ROK100 und TR-ROK200 bezeichnet. Diese werden entsprechend der Iterationszahl geordnet als Inhalte der zwei oben beschriebenen Markov-Logik-Netze integriert. Dadurch entstehen in Abhängigkeit von der Dauer des Parameterlernens folgende Markov-Logik-Netze – MLN OK50, MLN OK100, MLN OK200 und MLN ROK50, MLN ROK100, MLN ROK200. Jedes dieser sechs Markov-Logik-Netze wird zur Inferenz der Objektklassen mit den simulierten Evidenzdaten eingesetzt. Als Anfragevariablen bei der Inferenz werden die acht Objektklassen der tiefsten Taxonomieebene verwendet – *car*, *van*, *utilityVehicle*, *motorcyclist*, *bicyclist*, *pedestrian*, *infrastrObject* und *inAirIrrelObject*.

Die Inferenz wurde auf einem Laptop Intel Core 2 mit 2,40GHz Taktfrequenz und 4GByte RAM durchgeführt. Es wurde der MC-SAT Inferenzalgorithmus [Poo06] der frei verfügbaren Software Alchemy [Kok06] verwendet.

7.1.4 Ergebnisse

Dieses Kapitel stellt die Ergebnisse der Inferenz der oben eingeführten Markov-Logik-Netze mit den simulierten Evidenzdaten aus Kapitel 7.1.1 als Eingangsdaten vor. Bevor es mit der Vorstellung und Diskussion der erzielten Ergebnisse weitergeht, wird zuerst das vorliegende Schätzproblem nochmal kurz umrissen. Für jedes MLN und jedes der fünf manuell annotierten Testbilder wird der Inferenzalgorithmus mit acht unterschiedlichen Iterationsdauern zwischen 100 und 10500 Schritten aufgerufen. Die Inferenzausgabe beinhaltet eine Liste aller möglichen Grundatome von jeder der angefragten Objektklassen mit je einer zugeordneten Wahrscheinlichkeit. Anhand der Inferenzergebnisse werden pro MLN und Iterationsdauer für alle fünf Testbilder die in Kapitel 5 vorgestellten Gütekriterien CLL, Mehrklassen-AUC und -Genauigkeit berechnet.

Das Ziel dieser Untersuchung ist es, die Ergebnisse von MLN OK und MLN ROK zu vergleichen, um einerseits festzustellen, ob die Berücksichtigung der Objektrelationen in der Modellierung zu einer Verbesserung der Objektklassifikation führt, und andererseits, welchen Einfluss die Anzahl der Iterationen des Parameterlernens auf die Inferenzergebnisse hat.

Tabelle 7.4 zeigt einen Auszug aus den Inferenzergebnissen von MLN OK100 und MLN ROK100 für das in Abbildung 7.2 dargestellte Testbild 1 bei 500 Iterationen von MC-SAT. Darin veranschaulicht sind die mit beiden Markov-Logik-Netzen inferierten Wahrscheinlichkeiten für die Grundatome, welche die jeweils korrekte Klasse der Objekte repräsentieren. Die erste Spalte der Tabelle enthält diese wahren Grundatome. Die Einträge sind folgendermaßen zu lesen – die Aussage des Grundatoms *pedestrian(O2)*, dass das Objekt *O2* ein Fußgänger ist (was richtig ist), ist laut MLN OK100 mit der Wahrscheinlichkeit 0,59 und laut MLN ROK100 mit 0,93 korrekt. Die in der Tabelle aufgelisteten Wahrscheinlichkeiten sind bis auf wenige Einträge, die im Folgenden kurz diskutiert werden, vergleichbar. Zunächst werden die Objekte *O1* bis *O3* betrachtet. Sie stellen stehende Fußgänger dar, die mit dem Modell MLN ROK100 besser von Infrastrukturobjekten unterschieden werden können. Die auffallend geringe Wahrscheinlichkeit für den stehenden Fußgänger *O3* ist auf dessen Objektattribut „Längenverhältnis“ zurückzuführen, welches bei der eingenommenen Perspektive für ein starres Infrastrukturobjekt spricht. Weiterhin wird auf das Fahrzeug in der linken Bildhälfte eingegangen, welches durch die Verdeckung in die beiden Objekte *O10* und *O11* aufgeteilt wird. Während das MLN OK100 dem Objekt *O11* eine vernachlässigbare Wahrscheinlichkeit für die Klasse *car* zuweist, wird durch die zusätzliche Modellierung von Objektrelationen die korrekte Klassifikation mit 0,57 ermöglicht. Da die Objekte *O10* und *O11* auch mit einer gewissen Wahrscheinlichkeit der Klasse *van* zugeordnet werden, wird das Ergebnis noch verbessert, wenn die Objekte auf der Ebene

Tabelle 7.4: Inferierte Wahrscheinlichkeiten für die korrekten Klassen von 11 Objekten aus Testbild 1, gewonnen mit MLN OK100 und MLN ROK100 bei 500 MC-SAT Iterationen.

wahre Grundatome	MLN OK100	MLN ROK100
<i>pedestrian(O1)</i>	0,50799	0,65998
<i>pedestrian(O2)</i>	0,59399	0,93196
<i>pedestrian(O3)</i>	0,01605	0,05604
<i>car(O4)</i>	0,99995	0,96165
<i>car(O5)</i>	0,88396	0,74798
<i>car(O6)</i>	0,99795	0,95795
<i>car(O7)</i>	0,93196	0,95595
<i>inAirIrrelObject(O8)</i>	0,99995	0,99995
<i>undefObstacle(O9)</i>	0,98395	0,99995
<i>car(O10)</i>	0,67398	0,73798
<i>car(O11)</i>	4,9995e-05	0,57399

der Klasse *vehicle* klassifiziert werden. Das MLN ROK100 weist dieser Klasse für beide Objekte bereits eine Wahrscheinlichkeit von 0,98 zu, während sich die Wahrscheinlichkeit von MLN OK100 für *O10* lediglich auf 0,77 erhöht und für *O11* unverändert bleibt.

Für die Gesamtbeurteilung der Modelle wird pro MLN für jede Iterationsdauer des Inferenzalgorithmus ein Wert für die drei in Kapitel 5 eingeführten Gütekriterien ausgewertet. Diese Maße werden je Testbild berechnet und anschließend über alle fünf Testbilder gemittelt. Abbildung 7.3 veranschaulicht die Werte für den Mittelwert der mittleren CLL der acht Abfrageprädikate. Der Mittelwert der Standardabweichungen der CLL der acht Abfrageprädikate wird in Abbildung 7.4 dargestellt. Der Mittelwert der mittleren CLL ist ein direktes Maß für die Güte der inferierten Wahrscheinlichkeiten der acht angefragten Objektklassen. Die CLL ist eine negative logarithmische Größe. Somit liefert das MLN ROK mit steigenden Inferenziterationen bessere Ergebnisse im Vergleich zu MLN OK, sowohl für den Mittelwert der mittleren CLL als auch für den Mittelwert der Standardabweichungen der CLL der acht Abfrageprädikate. Die Ergebnisse des MLN ROK100 erreichen eine hohe Güte und konvergieren dabei in beiden Fällen schon bei 500 Iterationsschritten und somit bei deutlich weniger Iterationen des Inferenzalgorithmus als MLN ROK50 und MLN ROK200.

Abbildung 7.5 zeigt die Mehrklassen-AUC [Abu10] für die acht angefragten Objektklassen pro MLN und Iterationsdauer für die simulierten Objekte der fünf Testbilder. Die Mehrklassen-AUC für die Testmenge ist die mit dem Klassenprior ge-

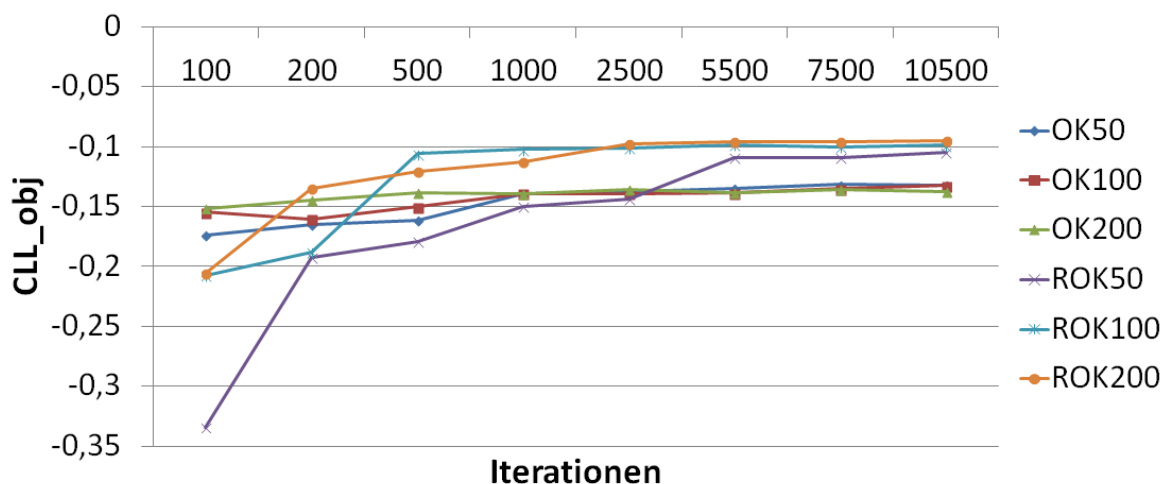


Abbildung 7.3: Mittelwert der mittleren CLL der acht angefragten Objektklassen.

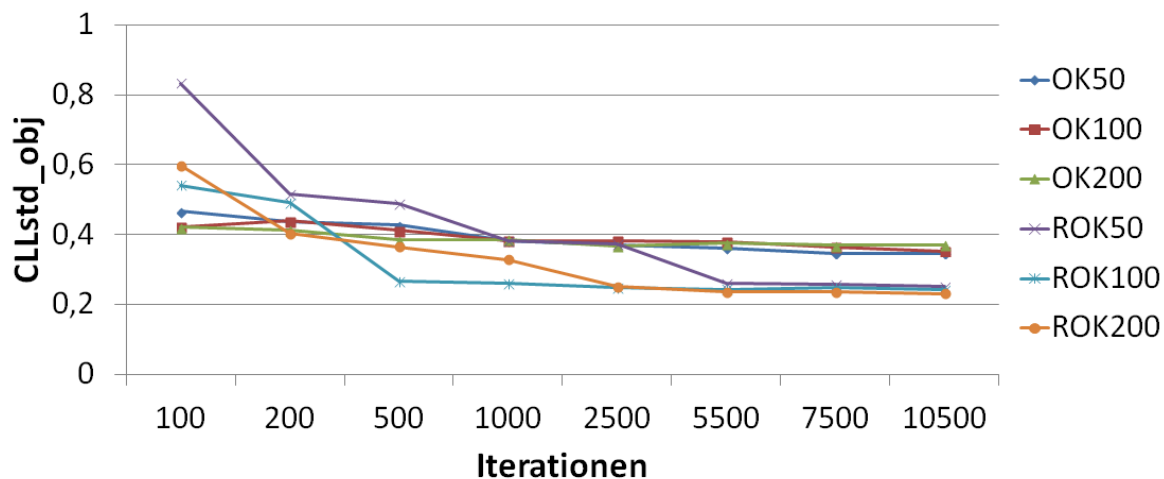


Abbildung 7.4: Mittelwert der Standardabweichung der CLL der acht angefragten Objektklassen.

wichtete Summe der *one-vs-rest* AUCs. Die Mehrklassen-AUC des MLN ROK200 erreicht für Iterationen ab 2500 den höchsten AUC-Wert mit etwa 0,98. Leicht tiefer, mit 0,976 für 2500 Iterationen, liegt der Mehrklassen-AUC-Wert des MLN ROK100, welches allerdings schon bei 500 Iterationen einen Wert von 0,973 erreicht, im Unterschied zu 0,961 von MLN ROK200, 0,945 von MLN ROK50 und 0,955 von MLN OK100. Obwohl alle MLN hohe Werte aufweisen, liefern die Modelle mit relationalem Anteil mit zunehmender Iterationszahl bessere Ergebnisse für die Mehrklassen-AUC als das MLN OK.

Als drittes Qualitätsmaß wurde die in Gleichung (5.1) definierte Mehrklassen-Genauigkeit [Abu10] für die acht Abfrageprädikate pro MLN und Iterationsdauer für die simulierte Testmenge (60 Objekte) berechnet. Dieses Maß ist ein Indiz für

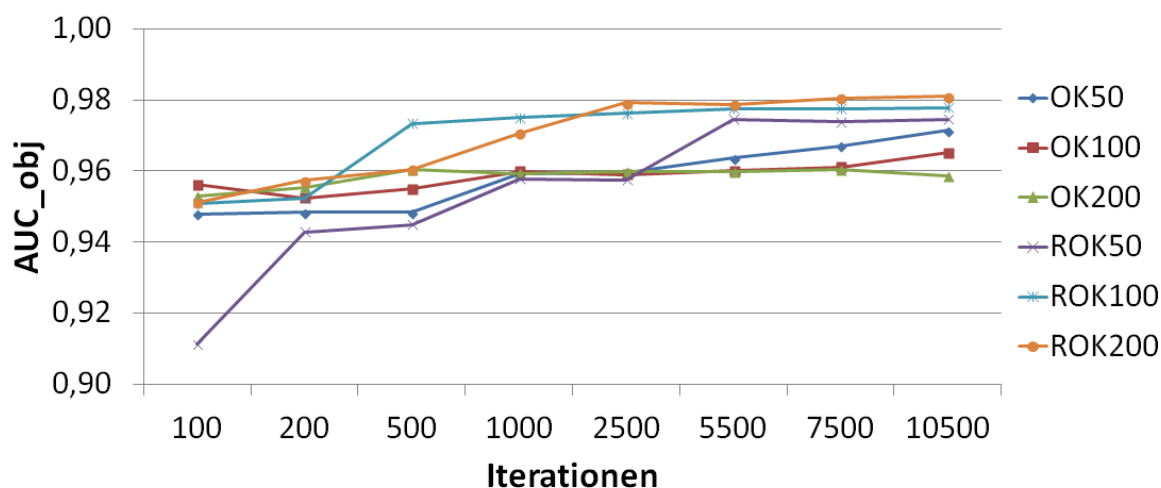


Abbildung 7.5: Mehrklassen-AUC der acht angefragten Objektklassen.

die Gesamtleistung eines Mehrklassen-Klassifikators. Die Objektklassifikation mit beiden vorgestellten Markov-Logik-Netzen liefert insgesamt gute Ergebnisse worauf die minimale Mehrklassen-Genauigkeit von 87% hindeutet. Die Genauigkeit von MLN OK100 und MLN ROK100 ist für alle untersuchten Iterationsdauern vergleichbar und liegt bei etwa 92%, wobei MLN ROK100 bei einigen Iterationsschritten auch 93% erreicht.

Anhand der simulierten Daten gelingt die Validierung der zwei vorgestellten Markov-Logik-Netze zur Objektklassifikation. Zusammenfassend lässt sich sagen, dass das MLN Relationale Objektklassifikation eine bessere Klassifikationsleistung liefert, insofern der Inferenzalgorithmus zumindest 500 Iterationen durchläuft. Dieses MLN ist durch die gewichteten Regeln, die Wissen über Objektklassen und Objektrelationen kodieren, deutlich komplexer im Vergleich zu MLN OK. Dies erklärt ebenfalls die erhöhte Mindestanzahl an Iterationen, die für eine gute Parameterschätzung erforderlich ist.

Während die Inferenzergebnisse von MLN OK von der unterschiedlichen Parameterlerndauer für die Gewichte der Regeln nicht stark beeinflusst werden, sind die Ergebnisse von MLN ROK in dieser Hinsicht differenzierter. Das MLN ROK100 konvergiert schon bei 500 Iterationen von MC-SAT und somit deutlich schneller im Vergleich zu MLN ROK50 und MLN ROK200. Neben der schnelleren Konvergenz indizieren die Ergebnisse auch keinen Nachteil in der Güte der Klassifikation, so dass im Weiteren die Lerndauer der Modellparameter auf 100 Iterationen gesetzt und nicht mehr explizit bei der MLN-Benennung angegeben wird.

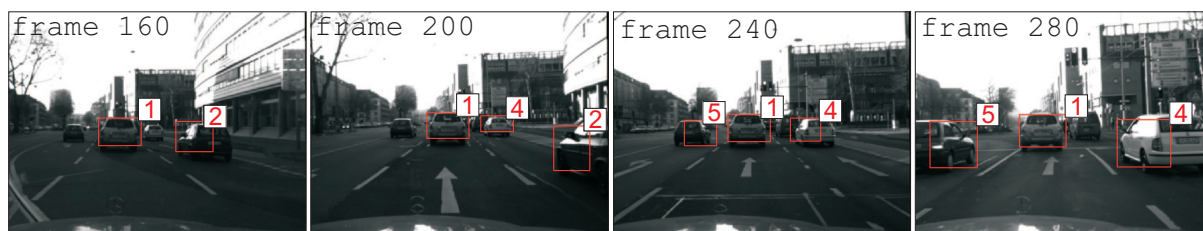
7.2 Inferenz anhand experimentell gewonnener Daten

Im vorherigen Kapitel wurde anhand simulierter Daten gezeigt, dass Wissen über die Objektrelationen zu einer Verbesserung der Objektklassifikation führt. Bei dieser Untersuchung wurden die bestehenden Relationen manuell vorgegeben, mit dem Ziel den Zusammenhang zwischen Objektklasse und -relation im Allgemeinen zu analysieren. Die Frage, wie diese Relationen anhand von gemessenen Sensordaten gewonnen werden können, wird in diesem Kapitel aufgegriffen. Zu diesem Zweck werden zwei Modelle zur Erkennung von Objektrelationen, vorgestellt in Kapitel 3, untersucht. Bei dem Modell „Inferierte Objektrelationen“ (IR) werden die Relationen anhand relativer Objektattribute probabilistisch-logisch inferiert, während das Modell „Direkte Objektrelationen“ (DR) die Definitionen für die Relationen in Abhängigkeit von den relativen Objektattributen fest vorgibt. Das untersuchte Modell MLN ROK wird durch das Modell IR erweitert. Die resultierenden Modelle werden anhand experimentell gewonnener Daten validiert.

7.2.1 Experimentell gewonnene Testdaten

Zum Testen mit experimentell gewonnenen Daten wurden drei geeignet vorverarbeitete, innerstädtische Videosequenzen verwendet. Die Sequenzen wurden mit dem nach vorne gerichteten Stereosystem des Versuchsfahrzeugs „AnnieWay“ des Instituts für Mess- und Regelungstechnik am Karlsruher Institut für Technologie aufgenommen. Eine detaillierte Beschreibung der Versuchsplattform ist in [Kam08] zu finden. Die ersten zwei der Videosequenzen wurden mit dem Algorithmus aus [Bac08] und die dritte mit dem in [Len11] präsentierten Algorithmus bewegungsbasiert segmentiert. Zu jeder der segmentierten 3D-Objekthypothesen werden quantitative Werte für die Größe der Bounding-Box, Bewegung und Position im Raum geliefert. Diese Messreihen werden zunächst zeitlich geglättet. Danach werden für jedes Bild Geschwindigkeitsbetrag und -richtung für jede Objekthypothese sowie auch relative Geschwindigkeitsrichtung, relative Position, relativer Abstand und Orientierungsdifferenz für jedes mögliche Paar Objekthypothesen berechnet. Dabei wird das Ego-Fahrzeug ebenfalls als Objekt in der Szene betrachtet, so dass dessen relative Objektattribute zu jeder segmentierten Objekthypothese ebenfalls ausgewertet werden.

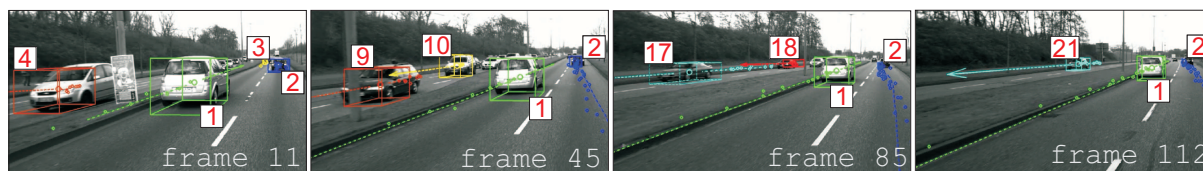
Sequenz 1 und Sequenz 2 zeigen jeweils die Anfahrt auf eine Kreuzung. Sequenz 3 ist auf einer innerstädtischen Umgehungsstraße mit Gegenverkehr aufgenommen worden. Das Ego-Fahrzeug ist bei allen drei untersuchten Sequenzen



(a) Sequenz 1



(b) Sequenz 2



(c) Sequenz 3

Abbildung 7.6: Bilder der Testsequenzen mit segmentierten Objekthypothesen.

in Fahrt. Abbildung 7.6 stellt je Sequenz mehrere repräsentative Einzelbilder mit Bounding-Boxes der im Vorfeld segmentierten Objekthypothesen dar. In Tabelle 7.5 sind statistische Informationen zu den drei Videosequenzen zusammengestellt, wie beispielsweise Anzahl der Einzelbilder und Gesamtanzahl der verfolgten Objekthypothesen je Sequenz. Bei den Angaben zur Anzahl der Objekte wird das Ego-Fahrzeug nicht berücksichtigt. In den ersten beiden Sequenzen werden wenige Objekte über längere Zeit in gleiche Fahrtrichtung verfolgt. In der kürzesten Sequenz 3 gibt es insgesamt deutlich mehr initialisierte Objekthypothesen. Das liegt daran, dass auch Objekte im Gegenverkehr über die Zeit verfolgt werden. Diese verschwinden relativ schnell aus dem Bildbereich, während kontinuierlich neue Objekte auftauchen. Dadurch ist auch die durchschnittliche Anzahl der Objekte pro Einzelbild für Sequenz 3 höher. Tabelle 7.5 zeigt in der letzten Spalte ebenfalls die auftretenden Objektklassen in jeder Sequenz sowie die Anzahl der annotierten Objekte pro Klasse. Davon ist ersichtlich, dass die zur Verfügung gestellten segmentierten Videosequenzen neben der Klasse *car* nur eine geringe Anzahl anderer Objektklassen beinhalten. Nichtsdestotrotz ermöglichen diese Daten eine erste Validierung der entwickelten Modelle zur Inferenz einer relationalen Situationsrepräsentation unter realen Bedingungen.

Um eine quantitative Auswertung der Inferenzergebnisse durchführen zu können, wurden in jedem Einzelbild der drei Sequenzen die korrekten Klassen der segmentierten Objekthypothesen sowie auch die bestehenden Relationen für alle möglichen Objektpaare (inklusive das Ego-Fahrzeug) manuell annotiert.

In Tabelle 7.6 ist die Anzahl der gelabelten Objektrelationen pro Sequenz, inklusive der Paarrelationen des Ego-Fahrzeugs, veranschaulicht. Zur Vereinfachung der Tabelle wurden die Umkehrrelationen ausgelassen. In Abhängigkeit von den aufgenommenen Verkehrsszenarien kommen unterschiedliche Objektrelationen in den Sequenzen vor.

Tabelle 7.5: Statistische Angaben zu den segmentierten realen Sequenzen.

Seq.	Anzahl Frames pro Seq.	Anzahl ObjektIDs pro Seq.	Anzahl Objekte je Frame			Annotierte Objektklassen (Anzahl pro Seq)	
			min	max	\emptyset		
1	293	5	1	4	2,4	<i>car</i> <i>undefObstacle</i>	(686) (5)
2	258	7	1	3	2,3	<i>car</i> <i>bicyclist</i> <i>undefObstacle</i>	(572) (7) (5)
3	104	21	1	6	3,8	<i>car</i> <i>undefObstacle</i>	(384) (13)

Tabelle 7.6: Anzahl der annotierten Objektrelationen je Sequenz. Die Umkehrrelationen sind zur Vereinfachung der Darstellung nicht aufgeführt.

Objektrelationen	Anzahl Objektrelationen		
	Seq 1	Seq 2	Seq 3
<i>follow</i>	623	574	483
<i>flank</i>	22	108	2
<i>approachOncoming</i>	-	-	462
<i>flankOncoming</i>	-	-	24
<i>leaveOncoming</i>	-	-	508
<i>approachCrossing</i>	-	236	18
<i>cross</i>	-	55	10
<i>leaveCrossing</i>	-	311	15
<i>moveTowards</i>	175	12	-
<i>movePast</i>	189	5	-
<i>moveAwayFrom</i>	91	19	-

7.2.2 Evidenzdaten

Die experimentell gewonnenen, quantitativen Testdaten werden in einem Abstraktionsschritt in Evidenzdaten überführt, die als Eingangsdaten für die Inferenz dienen. Die segmentierten 3D-Objekthypothesen in einem Bild zusammen mit dem Ego-Fahrzeug sind die konkreten Objekte des Diskursbereichs und werden als logische Konstanten auf die begriffliche Ebene überführt (z. B. *O0*, *O1*). Deren quantitativen, einfachen und relativen Objektattribute werden ebenfalls durch die jeweilige qualitative Repräsentation auf logische Konstanten abgebildet (z. B. *Small*, *OnGround*, *Close*, *NE*). Mit diesen Konstanten werden durch Substitution der Prädikatenvariablen Grundatome gebildet, wie zum Beispiel *sceneObject(O1)*, *hasHeight(O1,Small)* oder *hasDistance(O1,O2,Close)*. Das Ebenenkonzept *drivableArea* und seine zwei Unterkonzepte werden simuliert und die entsprechenden Grundatome generiert.

Mit den so gebildeten Grundatomen werden, aufgrund der Struktur der im nächsten Kapitel vorgestellten zu untersuchenden Modelle, zwei Sätze von Evidenzdaten für jedes Einzelbild der Testsequenzen erstellt. Die Evidenzdaten für beide bereits anhand simulierter Daten untersuchten Markov-Logik-Netze behalten den in Kapitel 7.1.2 beschriebenen Aufbau. Sie enthalten eine Menge wahrer Grundatome, welche die 3D-Objekte als das allgemeinste Objektkonzept *sceneObject* einführen und deren einfachen Attribute sowie Objektrelationen beschreiben. Die bestehenden Objektrelationen werden mit dem Modell „Direkte Objektrelationen“ anhand der relativen Objektattribute bestimmt. Somit können diese Evidenzdaten, ähnlich wie in Kapitel 7.1, bei der Inferenz von Objektklassen verwendet werden.

Der zweite Satz von Evidenzdaten enthält ebenfalls je 3D-Objekt, ein Grundatom, welches es als Objekt der Klasse *sceneObject* deklariert, sowie Grundatome, die die einfachen Objektattribute einführen. Im Gegensatz zu den oben beschriebenen Evidenzdaten sind darin allerdings keine Grundatome von bereits ausgewerteten Objektrelationen aufgelistet, sondern lediglich die Grundatome der relativen Objektattribute – *hasRelPos*, *hasDistance* und *hasDiffInOrient*. Mit den so gewonnenen Evidenzdaten pro Bild und dem im nächsten Kapitel beschriebenen MLN IROK lassen sich daher nicht nur Objektklassen, sondern auch Objektrelationen inferieren.

7.2.3 Methodik

Dieses Kapitel stellt die Modelle vor, die anhand der experimentell gewonnenen Daten untersucht werden. Zunächst listet Tabelle 7.7 die inhaltlichen Bausteine der betrachteten Markov-Logik-Netze auf. Das MLN „Objektklassifikation“ (OK)

und das MLN „Relationale Objektklassifikation“ (ROK) werden unverändert aus Kapitel „Inferenz anhand simulierter Daten“ übernommen. Bei dieser ersten Validierung des Modells ROK, welches Objektklassen und -relationen in Beziehung setzt, wurden die bestehenden Relationen zwischen den Objekten im Bild manuell gelabelt. Auf eine Erkennung der Objektrelationen wurde in diesem Fall aufgrund der fehlenden quantitativen Messungen bewusst verzichtet. In diesem Kapitel wird das MLN zur relationalen Objektklassifikation um die Erkennung der Objektrelationen aus gemessenen Daten ergänzt.

Tabelle 7.7: Zusammensetzung der anhand experimentell gewonnener Daten untersuchten Markov-Logik-Netze.

	MLN OK	MLN ROK	MLN IR	MLN IROK
AX-OK	✓	✓		✓
TR-OK	✓	✓		✓
AX-ROK		✓		✓
TR-ROK		✓		✓
AX-IR			✓	✓
TR-IR			✓	✓

Hierzu werden zwei Modelle zur Klassifikation der Relationen anhand von Objektattributen experimentell getestet. Bei dem Modell „Inferierte Objektrelationen“ (IR) werden die Relationen anhand automatisch gelernter Definitionen probabilistisch-logisch inferiert, während das Modell „Direkte Objektrelationen“ (DR) diese anhand manuell erstellter Objektattributenmatrix eindeutig definiert.

Das Modell DR modelliert, wie in Kapitel 3.5 beschrieben, die fest vorgegebenen Definitionen der Relationen in Abhängigkeit von der relativen Position und der relativen Geschwindigkeitsrichtung beziehungsweise der relativen Position und dem relativen Abstand zweier Objekte. Zur Bestimmung der Objektrelationen mit diesem Modell werden die quantisierten Werte für die Objektattribute ausgewertet und zu Grundatomen abstrahiert. Diese werden den Evidenzdaten hinzugefügt und als Eingangsdaten bei der Inferenz der Objektklassen mit dem MLN ROK verwendet.

Das MLN IR basiert auf dem Modell Inferierte Objektrelationen (Kapitel 3.4) und beinhaltet dadurch gelernte Definitionen für die Objektrelationen in Abhängigkeit von den relativen Objektattributen relative Position, relativer Abstand und Differenz der Geschwindigkeitsrichtungen. Das MLN IR setzt sich aus den Axiomen (AX-IR) und die gewichteten Regeln (TR-IR) des Modells aus Kapitel 3.4 zusammen und ermöglicht eine probabilistisch-logische Inferenz der Objektrelationen. Die Gewichte der Regeln dieses Markov-Logik-Netzes werden ebenfalls mit dem

diskriminativen konjugierten Gradientenverfahren [Low07] für die Dauer von 100 Iterationen gelernt.

Das MLN ROK wird um die Inhalte des MLN IR zu MLN IROK erweitert. Dadurch wird es möglich, während der Inferenz anhand experimentell gewonnener Daten mit einem Modell eine gemeinsame Klassifikation (engl.: *collective classification*) der bestehenden Objektrelationen und der Objektklassen durchzuführen. Die Modellierung und Inferenz mit MLN ROK und MLN IROK ist in Abbildung 7.7 als schematischer Überblick nochmals verdeutlicht.

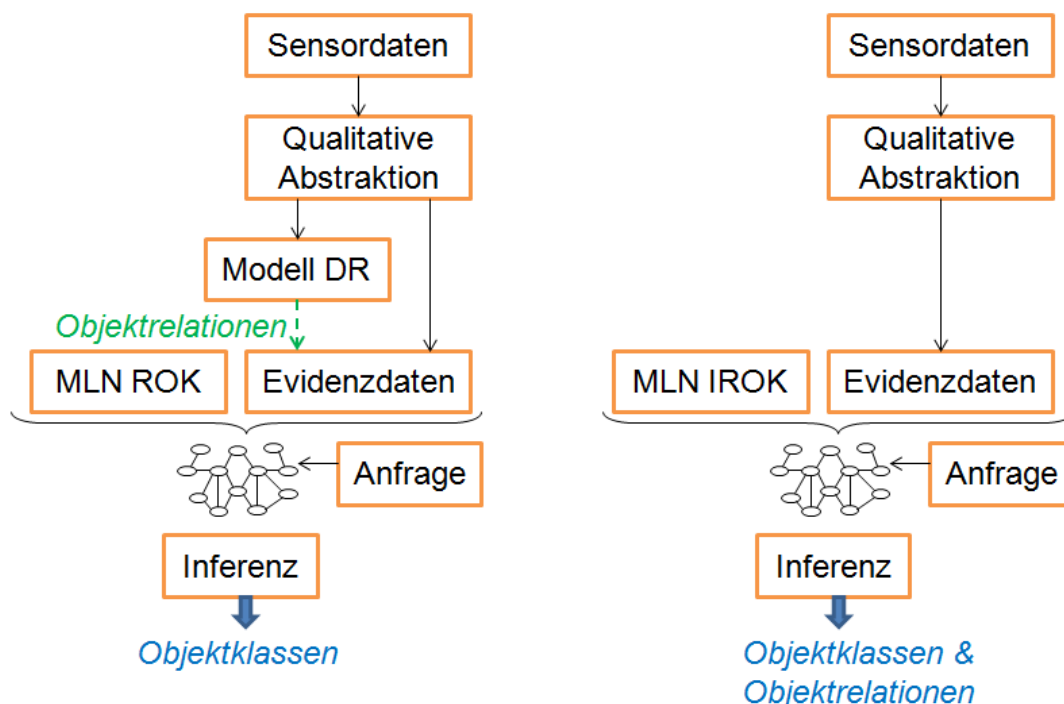


Abbildung 7.7: Modellierung und Inferenz mit MLN ROK und MLN IROK.

Die vier Markov-Logik-Netze aus Tabelle 7.7 werden zur Inferenz der Objektklassen und/oder Objektrelationen mit den Evidenzdaten aus dem vorherigen Kapitel als Eingangsdaten eingesetzt. Die Menge der Anfrageprädikate bei der Inferenz unterscheidet sich je nach MLN und ist in Tabelle 7.8 aufgelistet.

Zur Inferenz wird der MC-SAT Algorithmus [Poo06] der frei verfügbaren Software Alchemy [Kok06] verwendet. Die Inferenz wird an einem Intel Core 2 Laptop mit 2,40GHz Taktfrequenz und 4GByte RAM durchgeführt.

7.2.4 Ergebnisse

Dieses Kapitel stellt die Ergebnisse der Inferenz der im vorherigen Kapitel eingeführten Markov-Logik-Netze mit den realen Evidenzdaten der drei Stereosequen-

Tabelle 7.8: Anfrageprädikate bei der Inferenz der anhand experimentell gewonnener Daten untersuchten Markov-Logik-Netze.

Anfragevariablen	MLN OK	MLN ROK	MLN IR	MLN IROK
<i>car</i>	✓	✓		✓
<i>van</i>	✓	✓		✓
<i>utilityVehicle</i>	✓	✓		✓
<i>motorcyclist</i>	✓	✓		✓
<i>bicyclist</i>	✓	✓		✓
<i>pedestrian</i>	✓	✓		✓
<i>infrastrObject</i>	✓	✓		✓
<i>inAirIrrelObject</i>	✓	✓		✓
<i>vehicle</i>	✓	✓		✓
<i>vulnerableTP</i>	✓	✓		✓
<i>follow</i>			✓	✓
<i>flank</i>			✓	✓
<i>moveTowards</i>			✓	✓
<i>movePast</i>			✓	✓
<i>moveAwayFrom</i>			✓	✓
<i>approachOncoming</i>			✓	✓
<i>flankOncoming</i>			✓	✓
<i>leaveOncoming</i>			✓	✓
<i>approachCrossing</i>			✓	✓
<i>cross</i>			✓	✓
<i>approachCrossing</i>			✓	✓

zen als Eingangsdaten vor. Der Inferenzalgorithmus MC-SAT wird für jedes der Markov-Logik-Netze und jede der drei Sequenzen mit vier unterschiedlichen Iterationsdauern aufgerufen. Als Ausgabe der Inferenz wird eine Wahrscheinlichkeit für jedes mögliche Grundatom aller angefragten Objektklassen und/oder Objektrelationen geliefert. Anhand der Inferenzergebnisse werden pro MLN, Iterationsdauer und Sequenz gemittelt über alle Frames der Sequenz die in Kapitel 5 vorgestellten Gütekriterien CLL, Mehrklassen-AUC und -Genauigkeit berechnet. Um eine vergleichbare Auswertung dieser Gütemaße für die Erkennung der Objektrelationen mit dem Modell DR zu ermöglichen, wurden die Definitionsmatrizen aus den Abbildungen 3.3 und 3.4 geeignet in ein MLN überführt, das nur aus Axiomen besteht. Als Anfragevariablen bei der anschließenden Inferenz wurden die elf Objektrelationen aus Tabelle 7.8 angegeben. Für die so gewonnenen Inferenzergebnisse wurden die drei Gütekriterien berechnet.

Die Gütekriterien werden für die inferierten Objektklassen und Objektrelationen getrennt ermittelt. Folgende Namenskonvention pro Sequenz wird verwendet:

- CLL_{rel} ist der Mittelwert der mittleren CLL der elf angefragten Objektrelationen.
- $CLLstd_{rel}$ ist der Mittelwert der Standardabweichungen der CLL der elf angefragten Objektrelationen.
- AUC_{rel} ist die Mehrklassen-AUC, berechnet für die in der entsprechenden Sequenz vorkommenden Objektrelationen.
- ACC_{rel} ist die Mehrklassen-Genauigkeit der in der entsprechenden Sequenz vorkommenden Objektrelationen.
- CLL_{obj} ist der Mittelwert der mittleren CLL der acht Abfrageprädikate *car*, *van*, *utilityVehicle*, *motorcyclist*, *bicyclist*, *pedestrian*, *infrastrObject* und *inAirIrrelObject*.
- $CLLstd_{obj}$ ist der Mittelwert der Standardabweichungen der CLL dieser acht Objektklassen.
- ACC_{obj8} ist die Mehrklassen-Genauigkeit der acht oben erwähnten Objektklassen.
- ACC_{obj4} ist die Mehrklassen-Genauigkeit der vier Objektklassen der obersten Taxonomieebene *vehicle*, *vulnerableTP*, *infrastrObject* und *inAirIrrelObject*.

Im Folgenden werden die Inferenzergebnisse für die drei Sequenzen vorgestellt. Hierzu werden jeweils zunächst die Gütekriterien der Objektklassen diskutiert, bevor die Betrachtung der Ergebnisse der Objektrelationen stattfindet. Anschließend erfolgt die ganzheitliche Bewertung der Ergebnisse der Sequenzen unter dem Gesichtspunkt der Modellwahl.

Sequenz 1 repräsentiert das Szenario der Kreuzungsanfahrt. Wie in Tabelle 7.6 dargestellt, zeichnet sich die Sequenz durch eine hohe Zahl an Relationen der Klasse *moveTowards*, *movePast* und *moveAwayFrom* aus. Dies ist auf das vermehrte Vorkommen von relativ zu dem Egofahrzeug stehenden Fahrzeugen im Bereich des Lichtsignals zurückzuführen. Außerdem hat die vorliegende Situation zur Folge, dass nur eine geringe Zahl der acht möglichen Objektklassen in der Sequenz enthalten sind.

In Abbildung 7.8 und 7.9 sind der Verlauf von Mittelwert CLL_{obj} und Standardabweichung des Mittelwerts $CLL_{std_{obj}}$ der CLL für die Modelle OK, ROK und IROK dargestellt. Alle Modelle zeigen eine Konvergenz der Klassifikationsleistung für eine Iterationsanzahl größer gleich 5000. Das beste Resultat bei 10000 Inferenziterationen erzielt das Modell IROK mit einer CLL_{obj} von -0,05 im Gegensatz zu dem Modell ROK mit -0,07. Beide sind damit deutlich besser als das Modell OK mit einem Wert von -0,11. Die Resultate werden durch $CLL_{std_{obj}}$ bestätigt, für die IROK mit 0,2 ebenfalls den niedrigsten Wert, verglichen mit 0,39 (ROK) und 0,38 (OK), aufweist.

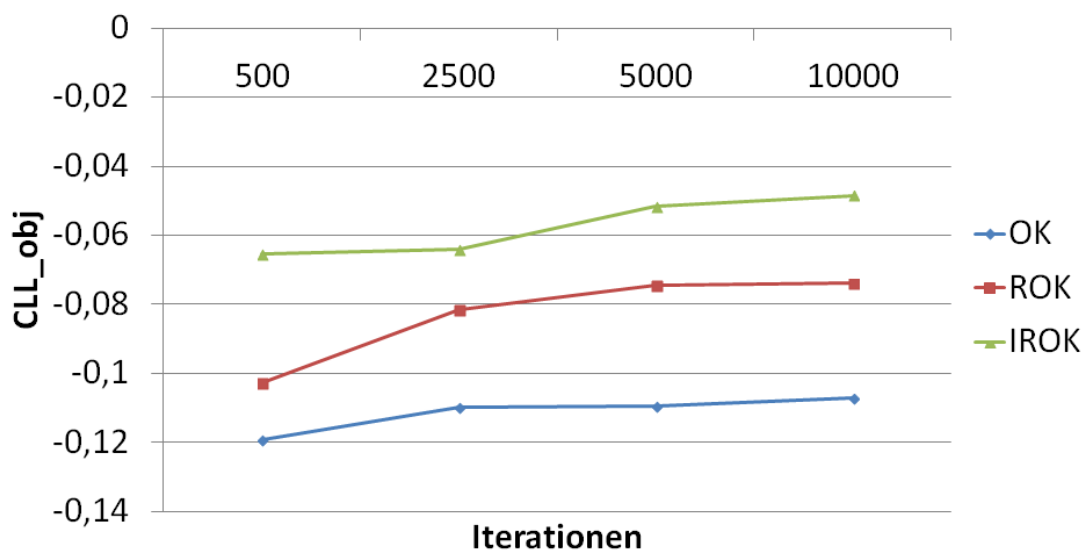


Abbildung 7.8: Mittelwert der mittleren CLL der acht angefragten Objektklassen CLL_{obj} für Sequenz 1.

Abbildung 7.10 zeigt die Mehrklassen-Genauigkeiten ACC_{obj4} und ACC_{obj8} . Für dieses Gütemaß ist eine deutlich geringere Neigung zur Konvergenz für höhere Iterationszahlen festzustellen. Auf der tiefsten Taxonomieebene mit 8 Objektklassen liegen alle drei Modelle mit 0,94 gleich auf. Betrachtet man allerdings die Objektklassen eine Abstraktionsebene höher, so erreichen die ACC_{obj4} Werte der relationalen Modelle mit 0,97 (ROK) und 0,96 (IROK) bessere Ergebnisse als OK mit 0,94.

Aufgrund der geringen Objektklassenanzahl wird für Sequenz 1 nicht die Mehrklassen-AUC sondern lediglich die AUC_{car} berechnet. Dabei wird die Klasse *car* als positiv und die restlichen als negativ betrachtet. Die AUC_{car} für 10000 Iterationen liegt für die relationalen Modelle mit 0,91 (IROK) und 0,90 (ROK) deutlich über dem MLN OK, das einen Wert von 0,78 erreicht.

Nach den Ergebnissen der Objektklassen zeigen die Kurven in Abbildung 7.11 und 7.12 die Werte CLL_{rel} und $CLL_{std_{rel}}$ für die bedingte Loglikelihood der Re-

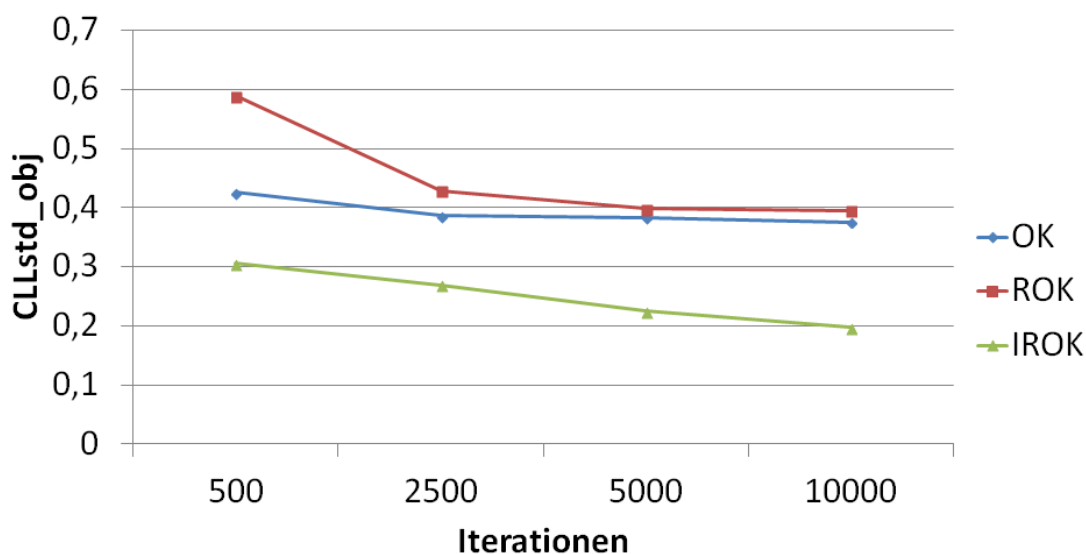


Abbildung 7.9: Mittelwert der Standardabweichung der CLL der acht angefragten Objektklassen $CLLstd_{obj}$ für Sequenz 1.

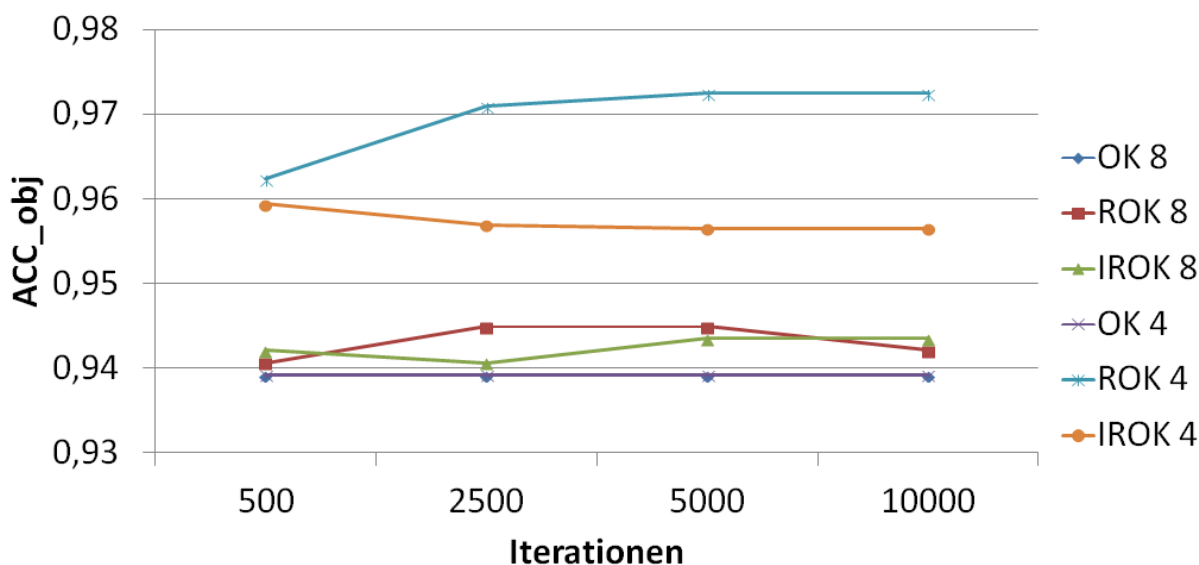


Abbildung 7.10: Mehrklassen-Genauigkeit berechnet für die acht ACC_{obj8} und die vier ACC_{obj4} angefragten Objektklassen für Sequenz 1.

lationen. Wiederum kann eine Konvergenz der Werte für mehr Iterationsschritte festgestellt werden, die Werte für IROK und DR liegen für 10000 Iterationen mit $-0,1$ und $-0,08$ für CLL_{rel} und $0,59$ und $0,68$ für $CLLstd_{rel}$ in ähnlichen Größenordnungen, während das MLN IR mit $-0,04$ und $0,35$ deutlich bessere Resultate für beide Kennzahlen aufweist.

Die Mehrklassen-Genauigkeit ACC_{rel} zeigt für die Relationen ebenfalls Konvergenz für erhöhte Iterationsschrittzahl. Die Ergebnisse für die verschiedenen Inferenziterationen sind in Abbildung 7.13 dargestellt. Bei 10000 Iterationen er-

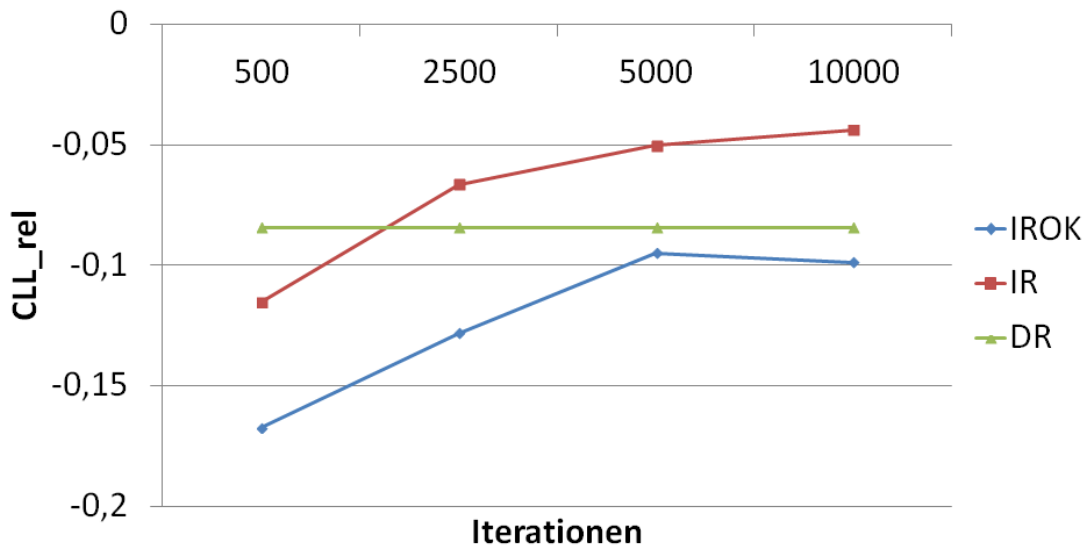


Abbildung 7.11: Mittelwert der mittleren CLL der elf angefragten Objektrelationen CLL_{rel} für Sequenz 1.

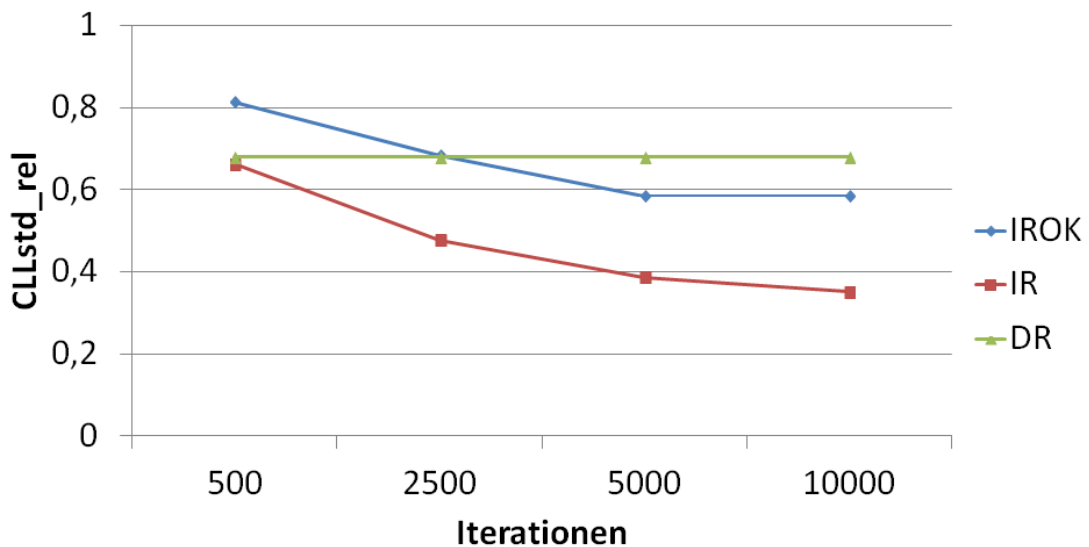


Abbildung 7.12: Mittelwert der Standardabweichungen der CLL der elf angefragten Objektrelationen $CLLstd_{rel}$ für Sequenz 1.

reichen die Modelle IR und DR für dieses Gütemaß mit 0,88 und 0,87 nahezu identische Ergebnisse. Das Klassifikationsergebnis von MLN IROK fällt mit 0,77 schlechter aus. Der Verlauf der Mehrklassen-AUC AUC_{rel} ist in Abbildung 7.14 zu sehen. Das Ergebnis ist ähnlich der CLL_{rel} , so dass die Modelle IROK und DR mit 0,92 und 0,93 praktisch identische Werte aufweisen und hinter MLN IR mit einem Wert von 0,98 zurückbleiben.

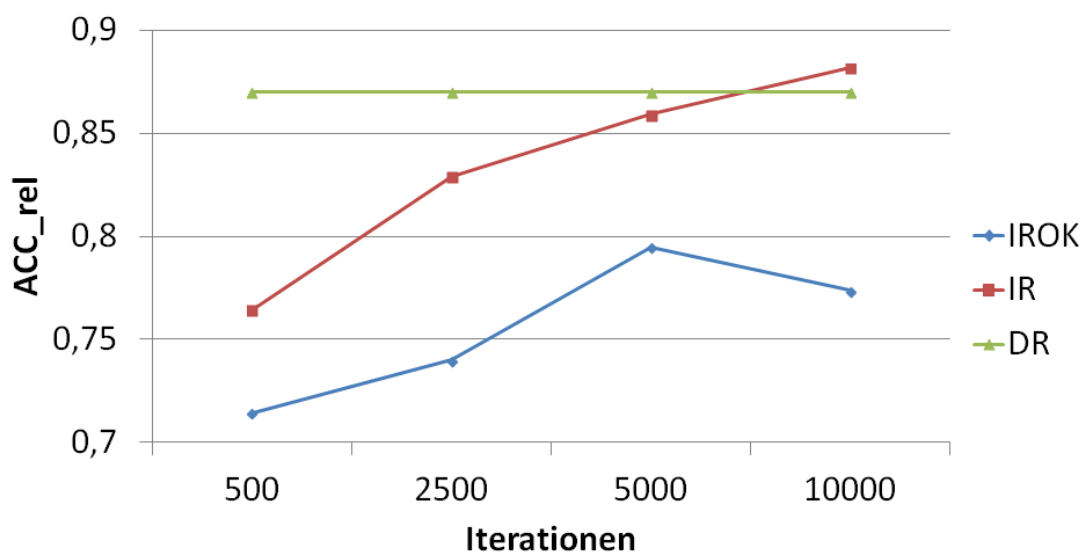


Abbildung 7.13: Mehrklassen-Genauigkeit ACC_{rel} der in Sequenz 1 vorkommenden Objektrelationen.

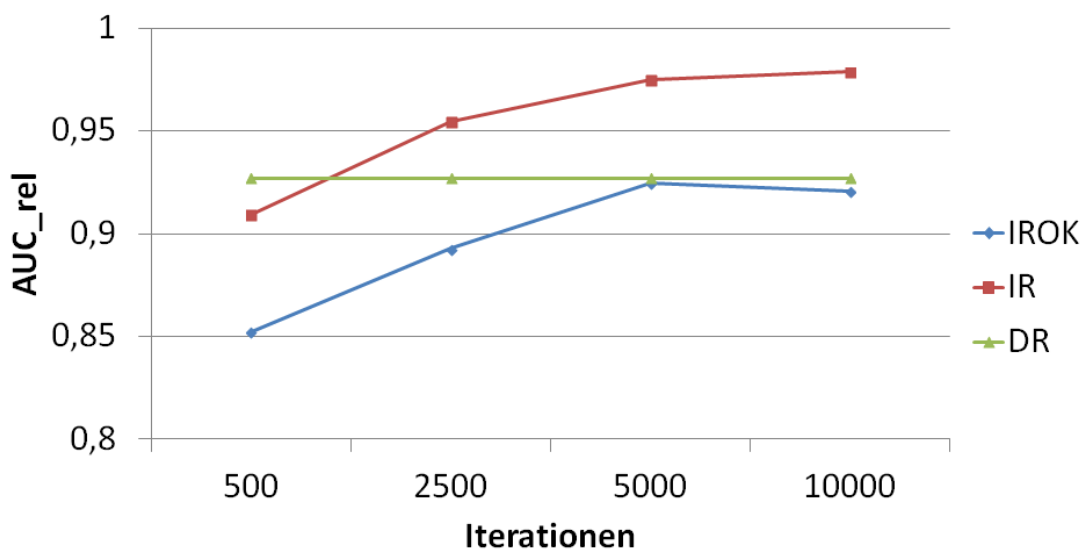


Abbildung 7.14: Mehrklassen-AUC AUC_{rel} der in Sequenz 1 vorkommenden Objektrelationen.

Sequenz 2 wurde ebenfalls bei der Anfahrt an eine Kreuzung gewonnen. Im Gegensatz zu Sequenz 1 liegen durch eine Abbiegespur allerdings mehr Relationen vom Typ *approachCrossing*, *leaveCrossing* und *cross* vor. Zusätzlich ist die Anzahl der Klassen dadurch erhöht, dass kurzzeitig auch stehende Verkehrsteilnehmer mit Fahrrädern auftreten.

In den Abbildungen 7.15 und 7.16 sind wiederum der Verlauf von CLL_{obj} und $CLL_{std_{obj}}$ für die drei Modelle dargestellt. Für die betrachtete Sequenz liegt nur für das MLN ROK eine merkliche Konvergenz in Abhängigkeit von den Iterations-

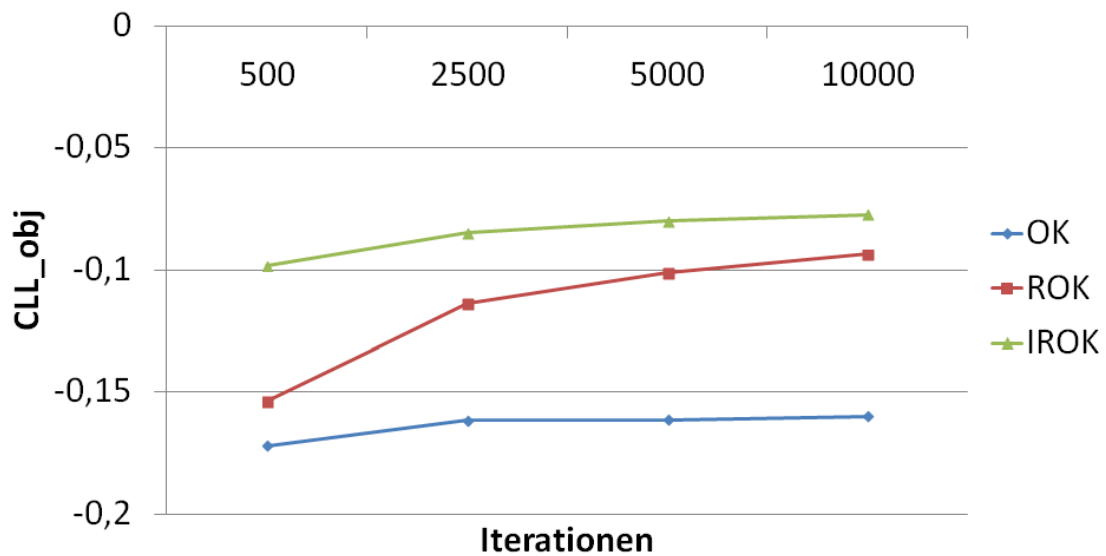


Abbildung 7.15: Mittelwert der mittleren CLL der acht angefragten Objektklassen CLL_{obj} für Sequenz 2.

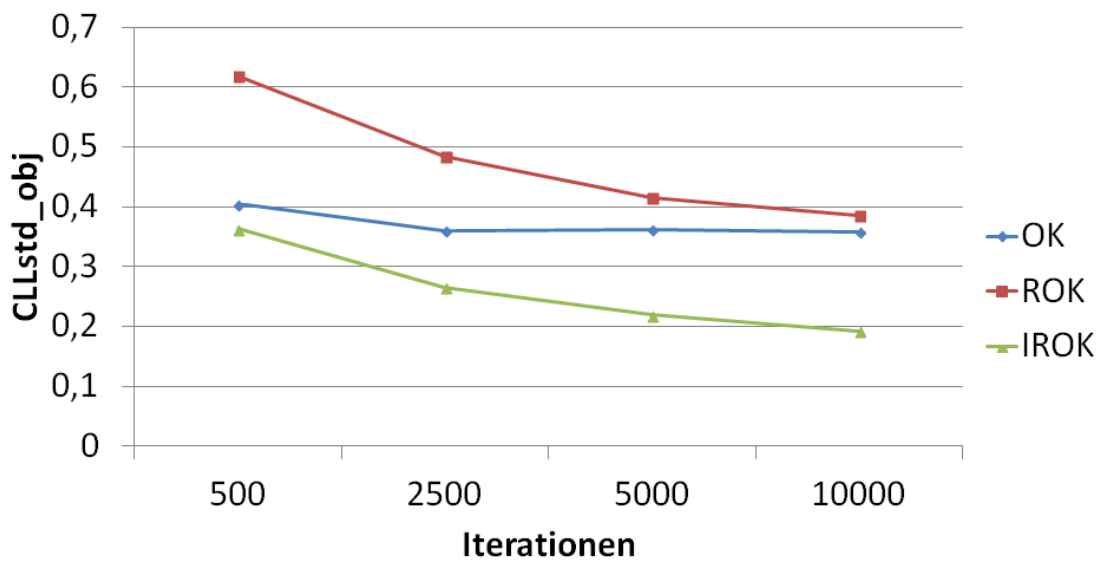


Abbildung 7.16: Mittelwert der Standardabweichung der CLL der acht angefragten Objektklassen $CLLstd_{obj}$ für Sequenz 2.

schritten vor, während diese für MLN IROK vernachlässigbar ist. Die Ergebnisse der relationalen Modelle liegen für 10000 Iterationen mit -0,08 (IROK) und -0,09 (ROK) in benachbarten Größen, während für das MLN OK eine CLL_{obj} von -0,16 vorliegt. Der geringste Mittelwert der Standardabweichungen der CLL weist das Modell IROK mit 0,19 auf, die $CLLstd_{obj}$ ist für MLN ROK mit 0,39 und für MLN OK mit 0,36 ähnlich aber nahezu doppelt so hoch verglichen mit MLN IROK.

Auch für die Werte der Gütekriterien ACC_{obj4} und ACC_{obj8} ist in der zweiten Sequenz nur für das Modell ROK 8 eine merkliche Abhängigkeit von der Itera-

tionsdauer festzustellen, was nochmals detailliert in Abbildung 7.17 zu sehen ist. Für 10000 Iterationen und unter Berücksichtigung aller möglichen Objektklassen, liegen die Werte für MLN IROK 8 und MLN ROK 8 mit jeweils 0,83 merklich über dem Modell ohne relationale Beziehungen MLN OK 8, welches den Wert 0,77 hat. Dieses Ergebnis ist auch auf der höheren Taxonomieebene unverändert, während eine Verbesserung der relationalen Modelle auf den sehr hohen Wert 0,98 erfolgt. Abbildung 7.18 veranschaulicht exemplarisch die Wahrheitstabellen für die vier angefragten Objektklassen von MLN OK, ROK und IROK für Sequenz 2 bei 10000 MC-SAT Iterationen.

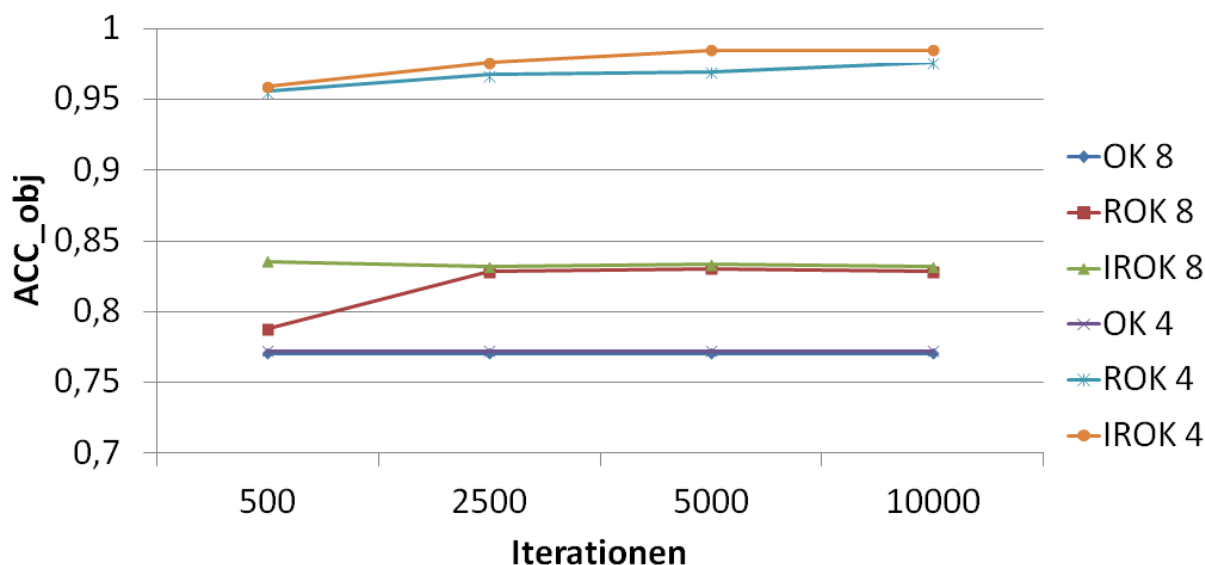


Abbildung 7.17: Mehrklassen-Genauigkeit berechnet für die acht ACC_{obj8} und die vier ACC_{obj4} angefragten Objektklassen für Sequenz 2.

	<i>v</i>	<i>u</i>	<i>o</i>	<i>i</i>
<i>v</i>	447	16	109	0
<i>u</i>	0	0	7	0
<i>o</i>	1	0	4	0
<i>i</i>	0	0	0	0

(a) MLN OK

	<i>v</i>	<i>u</i>	<i>o</i>	<i>i</i>
<i>v</i>	570	1	1	0
<i>u</i>	7	0	0	0
<i>o</i>	5	0	0	0
<i>i</i>	0	0	0	0

(b) MLN ROK

	<i>v</i>	<i>u</i>	<i>o</i>	<i>i</i>
<i>v</i>	568	2	2	0
<i>u</i>	0	6	1	0
<i>o</i>	4	0	1	0
<i>i</i>	0	0	0	0

(c) MLN IROK

Abbildung 7.18: Wahrheitstabellen für die vier angefragten Objektklassen für Sequenz 2 bei 10000 MC-SAT Iterationen. Verwendete Abkürzungen für die Klassen: *v* - *motorizedTP*, *u* - *unmotorizedTP*, *o* - *infrastructObject*, *i* - *inAirIrrelObject*.

Die geringe Objektklassenanzahl der zweiten Sequenz ermöglicht wiederum keine volle Berechnung der Mehrklassen-AUC. Die für diesen Fall aussagekräftigere

AUC_{car} liefert für 10000 Iterationen die Werte 0,97 (MLN IROK), 0,93 (MLN ROK) und 0,94 (MLN OK).

Der Mittelwert der Objektrelationen CLL_{rel} zeigt ein äquivalentes Verhalten zu Sequenz 1, bei der bereits eine Konvergenz und Verbesserung der Ergebnisse mit steigender Iterationszahl festgestellt werden kann. Die MLN IROK und MLN IR erreichen dabei mit -0,31 und -0,3 bzw. 1,18 und 1,13 für $CLL_{std_{rel}}$ bessere Ergebnisse als das Modell DR mit -0,36 bzw. 1,34. Der Sachverhalt ist in den Abbildungen 7.19 und 7.20 verdeutlicht, auffallend ist dabei der höhere Wert für $CLL_{std_{rel}}$ verglichen mit den Sequenzen eins und drei. Dieser Sachverhalt ist auf eine erhöhte Anzahl von Sensorfehlern und damit einhergehenden Unsicherheiten in den Relationen *leaveCrossing / approachCrossing* zurückzuführen.

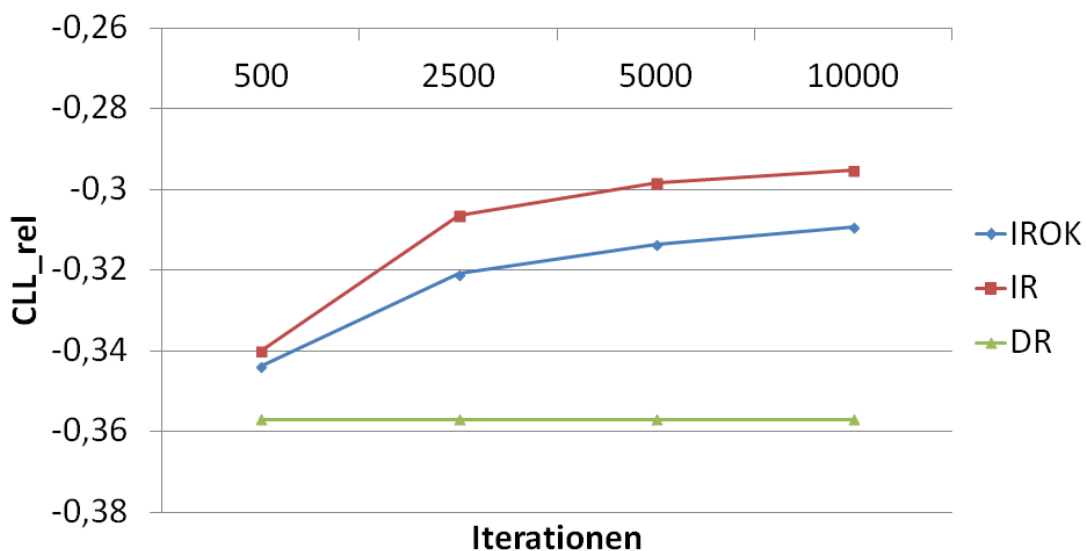


Abbildung 7.19: Mittelwert der mittleren CLL der elf angefragten Objektrelationen CLL_{rel} für Sequenz 2.

Die angesprochenen Unzulänglichkeiten der Sensordaten führen ebenfalls zu niedrigen Werten für die Mehrklassen-Genauigkeit ACC_{rel} . In Abbildung 7.21 sind die Kurven dargestellt und zeigen für alle Modelle und 10000 Iterationen ähnliche, aber geringe Werte von 48%, 52% und 54% für IROK, IR und DR.

Deutlich besser sind die Werte der Mehrklassen-AUC (AUC_{rel}), die den höchsten Wert im MLN IR mit 0,72 erreicht. Die Werte für die Modelle IROK und DR liegen mit 0,65 und 0,69 etwas niedriger, was in Abbildung 7.22 zu sehen ist.

Sequenz 3 wurde bei der Fahrt auf einer innerstädtischen Umgehungsstraße aufgezeichnet. Im Gegensatz zu den vorherigen Sequenzen treten daher vermehrt Relationen der Klassen *approachOncoming*, *leaveOncoming* und *flankOncoming*

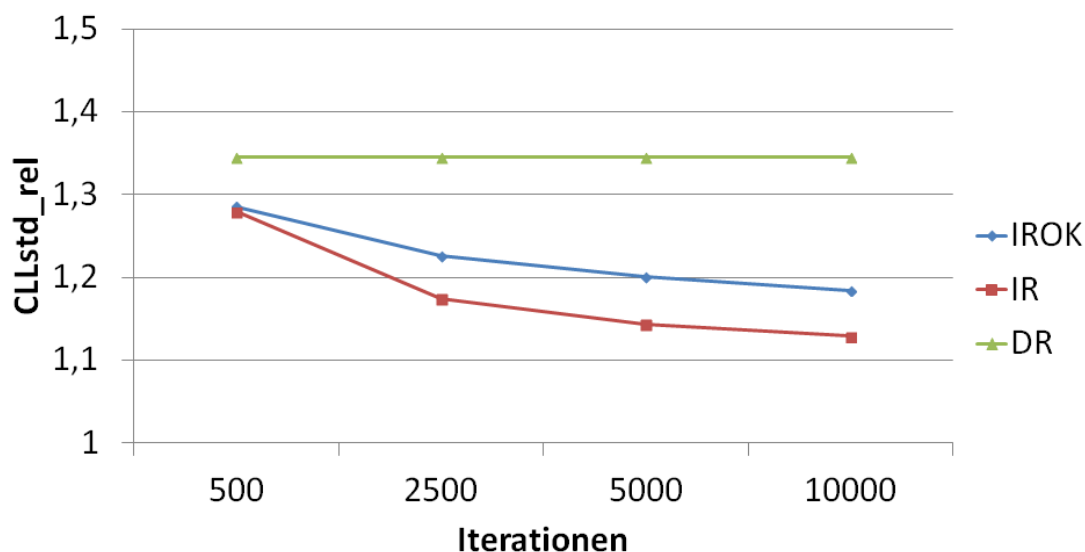


Abbildung 7.20: Mittelwert der Standardabweichungen der CLL der elf angefragten Objektrelationen CLL_{std_rel} für Sequenz 2.

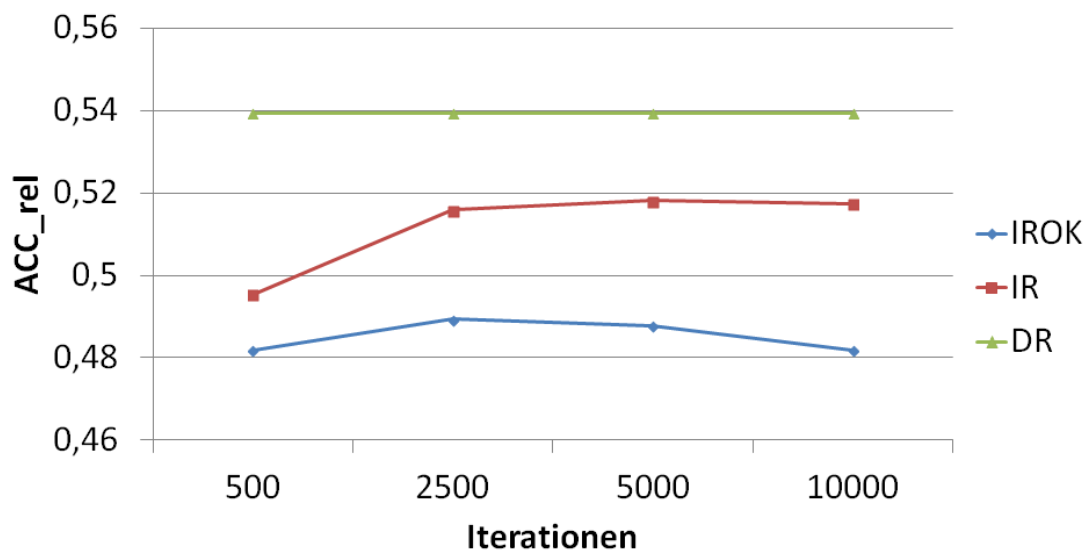


Abbildung 7.21: Mehrklassen-Genauigkeit ACC_{rel} der in Sequenz 2 vorkommenden Objektrelationen.

auf. Obwohl die Anzahl der Einzelbilder geringer als in den beiden anderen Sequenzen ist, liegt die mittlere Objektanzahl in diesen höher und erreicht ein Maximum von sechs im Bild gleichzeitig auftretenden Objekten.

Die conditional loglikelihood bezogen auf die Objekte ist in den Abbildungen 7.23 und 7.24 dargestellt. Auch für die dritte Sequenz kann die Konvergenz für größere Iterationszahlen des Inferenzalgorithmus festgehalten werden. Die aufgeführten Werte beziehen sich daher immer auf 10000 Iterationen. Für Sequenz 3 können keine signifikanten Unterschiede in den drei Modellen für CLL_{obj} beobachtet wer-

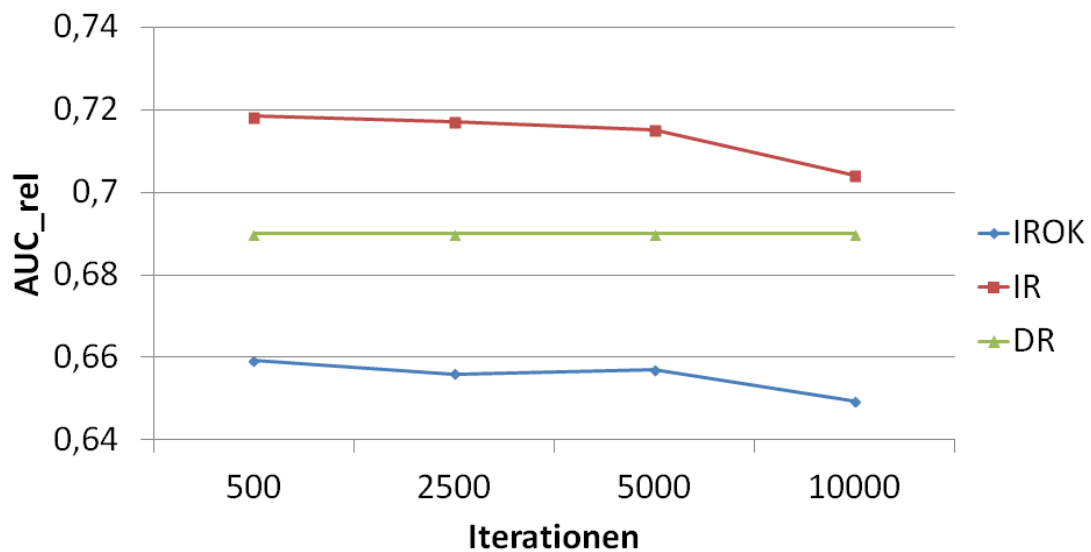


Abbildung 7.22: Mehrklassen-AUC AUC_{rel} der in Sequenz 2 vorkommenden Objektrelationen.

den und sind mit $-0,16$ (IROK), $-0,17$ (ROK) und $-0,17$ (OK) nahezu identisch. Die entsprechenden Werte für $CLL_{std_{obj}}$ sind $0,39$, $0,45$ und $0,34$, so dass in diesem Fall das Modell ohne relationalen Anteile eine leicht geringere Streuung zeigt.

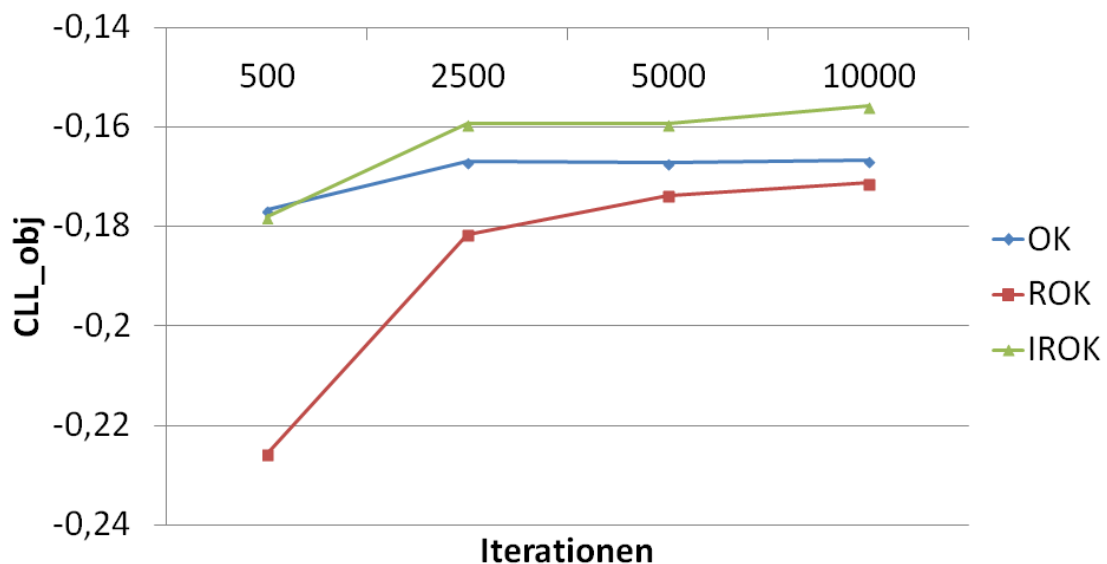


Abbildung 7.23: Mittelwert der mittleren CLL der acht angefragten Objektklassen CLL_{obj} für Sequenz 3.

Dieser leichte Vorteil des Modells OK bestätigt sich für Sequenz 3 auch in dem Qualitätsmaß AUC_{car} . Die erzielten Ergebnisse sind von ähnlich hoher Güte und erreichen $0,84$ und $0,86$ für MLN ROK und MLN IROK, sowie $0,93$ für MLN OK. Die in Abbildung 7.25 präsentierte Mehrklassen-Genauigkeit ACC_{obj} zeigt wie-

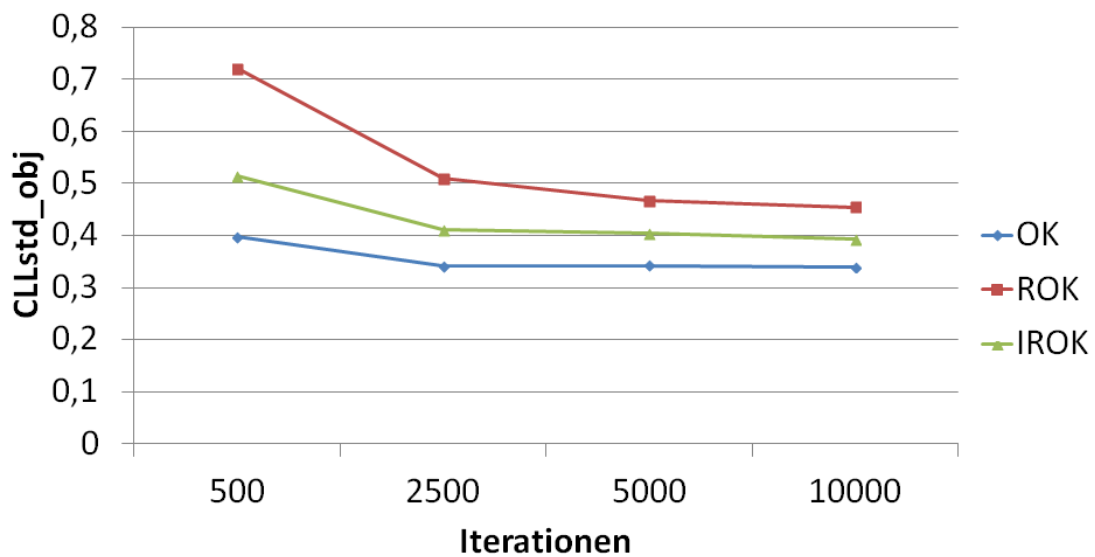


Abbildung 7.24: Mittelwert der Standardabweichung der CLL der acht angefragten Objektklassen $CLLstd_{obj}$ für Sequenz 3.

derum einen deutlich geringeren Konvergenzeinfluss als die anderen beiden Güte- maße. Wie schon in den Sequenzen eins und zwei, liegen die Werte für die relationalen Modelle über denen des Modells OK. Während auf der Taxonomieebene mit vier Klassen die ACC_{obj4} Werte mit 97% (IROK und ROK) allgemein und im Vergleich zu MLN OK ($ACC_{obj4}(OK) = 0,77$) sehr hoch liegen, sind auf der tiefsten Taxonomieebene mit $ACC_{obj8}(IROK) = 0,76$, $ACC_{obj8}(ROK) = 0,76$ und $ACC_{obj8}(OK) = 0,65$ leicht geringere Werte zu finden.

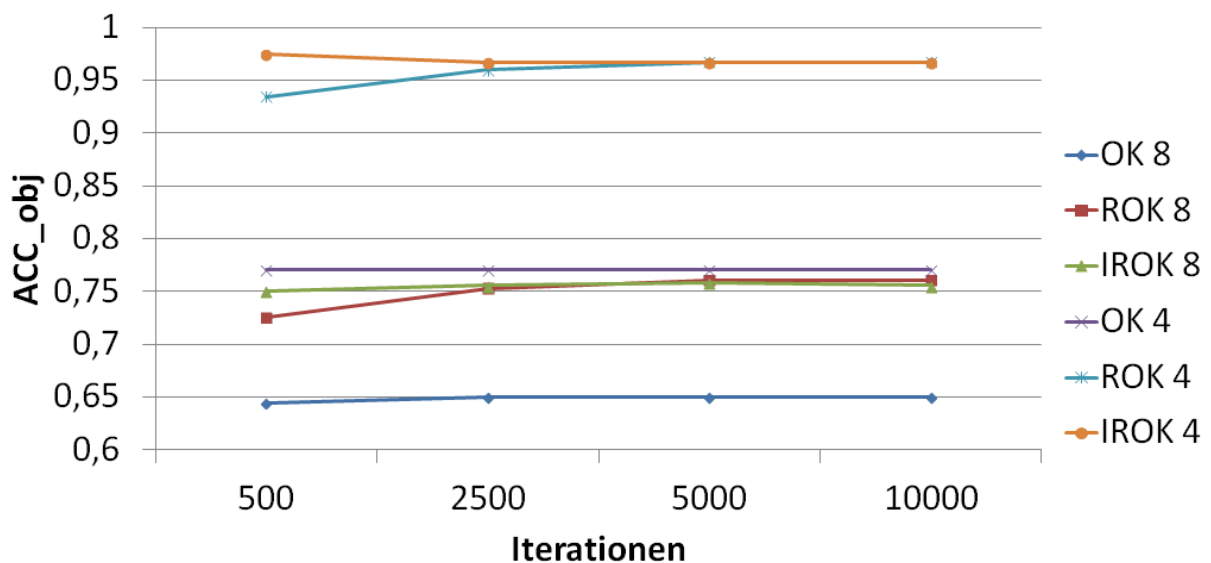


Abbildung 7.25: Mehrklassen-Genauigkeit berechnet für die acht ACC_{obj8} und die vier ACC_{obj4} angefragten Objektklassen für Sequenz 3.

Der detaillierte Verlauf der relationalen bedingten Loglikelihood ist in Abbildung 7.26 und 7.27 dargestellt. Das MLN IROK und MLN IR weisen hierbei mit $-0,05/0,29$ und $-0,02/0,17$ für CLL_{rel} / $CLLstd_{rel}$ bessere Ergebnisse als das Modell DR mit $-0,07/0,65$ auf.

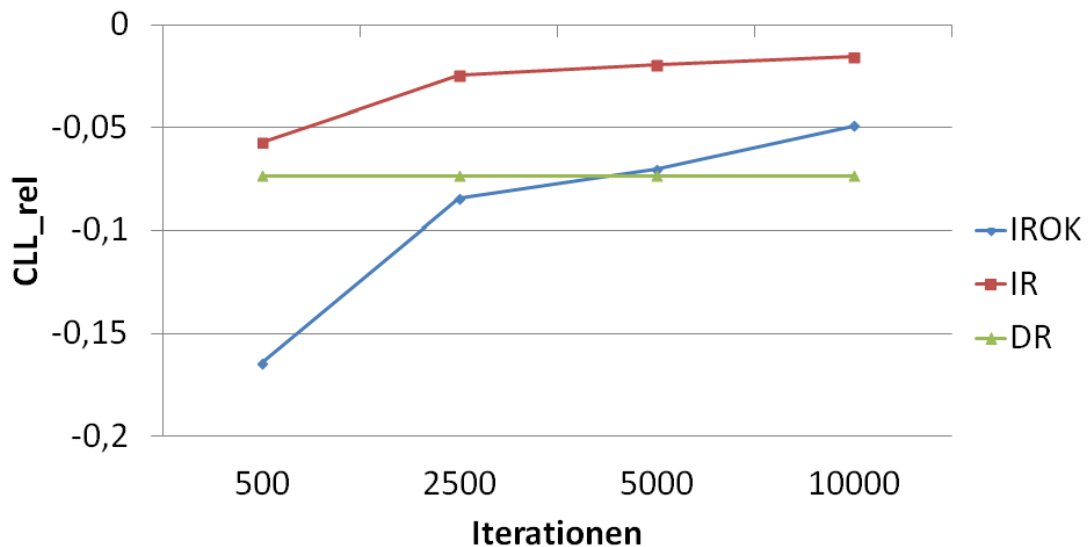


Abbildung 7.26: Mittelwert der mittleren CLL der elf angefragten Objektrelationen CLL_{rel} für Sequenz 3.

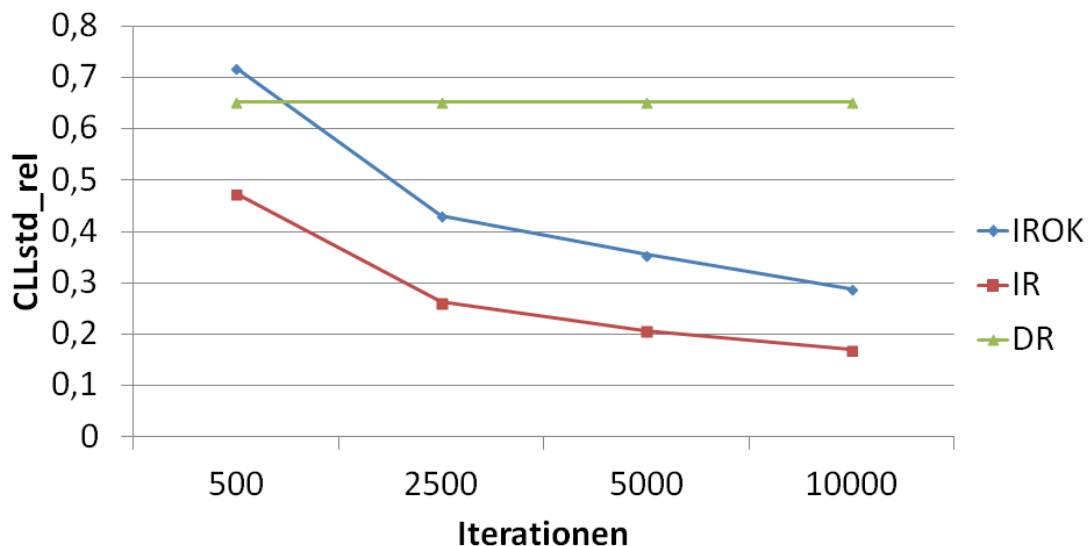


Abbildung 7.27: Mittelwert der Standardabweichungen der CLL der elf angefragten Objektrelationen $CLLstd_{rel}$ für Sequenz 3.

Die Kurve in Abbildung 7.28 stellt mit ACC_{rel} die mittlere Mehrklassen-Genauigkeit bezüglich der Relationen für die dritte Sequenz dar. Es werden Werte von 97% und 94% für die Modelle IR und DR erreicht, gegenüber denen das MLN

IROK mit gleichwohl hohen 90% etwas abfällt. Abbildung 7.29 veranschaulicht exemplarisch die Wahrheitstabellen für die inferierten Objektrelationen von den Modellen DR und IR für Sequenz 3 bei 5000 MC-SAT Iterationen.

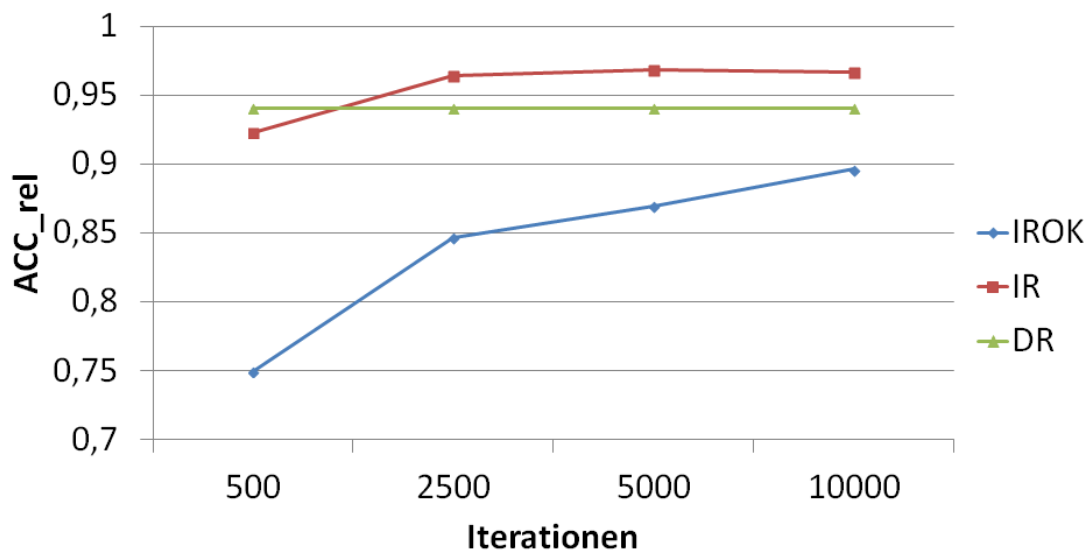


Abbildung 7.28: Mehrklassen-Genauigkeit ACC_{rel} der in Sequenz 3 vorkommenden Objektrelationen.

Zuletzt ist in Abbildung 7.30 die Mehrklassen-AUC für die Objektrelationen abgebildet. Diese zeigt für MLN IROK mit 99,6%, für MLN IR mit 99,7% und für das Modell DR mit 97% ausgezeichnete Werte.

Schließlich fasst Tabelle 7.9 exemplarisch die Dauer der Inferenz für eines der Bilder aus Sequenz 1 mit drei Objekten. Zu beachten ist, dass die Anzahl der Anfragevariablen bei der Inferenz mit MLN IROK deutlich größer ist, als bei den anderen zwei Modelle. Die Anzahl der Grundatome und -klauseln der einzelnen MLNs bei der Inferenz ist als Vergleichsindikator für die Komplexität der Modelle veranschaulicht.

Tabelle 7.9: Dauer der Inferenz in [sec] für eines der Bilder aus Sequenz 1 mit drei Objekten.

MC-SAT Iterationen	MLN OK	MLN ROK	MLN IROK
500	0,24	0,41	0,77
2500	0,92	1,50	2,48
5000	1,78	2,98	4,55
10000	3,48	5,61	8,09
Grundatome	30	166	232
Grundklauseln	132	806	2642

	<i>fo</i>	<i>fl</i>	<i>mAF</i>	<i>mT</i>	<i>mP</i>	<i>flOn</i>	<i>aOn</i>	<i>lOn</i>	<i>cr</i>	<i>aCr</i>	<i>lCr</i>
<i>fo</i>	463	0	0	0	0	0	0	0	0	0	20
<i>fl</i>	0	2	0	0	0	0	0	0	0	0	0
<i>mAF</i>	0	0	0	0	0	0	0	0	0	0	0
<i>mT</i>	0	0	0	0	0	0	0	0	0	0	0
<i>mP</i>	0	0	0	0	0	0	0	0	0	0	0
<i>flOn</i>	0	0	0	0	0	22	0	2	0	0	0
<i>aOn</i>	0	0	0	0	0	6	434	0	0	11	11
<i>lOn</i>	0	0	0	0	0	10	0	482	0	0	16
<i>cr</i>	0	0	0	0	0	0	0	0	5	5	0
<i>aCr</i>	0	0	0	0	0	0	0	0	7	9	2
<i>lCr</i>	0	0	0	0	0	0	0	0	0	0	15

(a) DR

	<i>fo</i>	<i>fl</i>	<i>mAF</i>	<i>mT</i>	<i>mP</i>	<i>flOn</i>	<i>aOn</i>	<i>lOn</i>	<i>cr</i>	<i>aCr</i>	<i>lCr</i>
<i>fo</i>	483	0	0	0	0	0	0	0	0	0	0
<i>fl</i>	0	2	0	0	0	0	0	0	0	0	0
<i>mAF</i>	0	0	0	0	0	0	0	0	0	0	0
<i>mT</i>	0	0	0	0	0	0	0	0	0	0	0
<i>mP</i>	0	0	0	0	0	0	0	0	0	0	0
<i>flOn</i>	0	0	0	0	0	22	0	2	0	0	0
<i>aOn</i>	0	0	0	0	0	4	458	0	0	0	0
<i>lOn</i>	0	0	0	0	0	20	8	480	0	0	0
<i>cr</i>	0	0	0	0	0	0	0	0	8	2	0
<i>aCr</i>	0	0	0	0	0	0	0	0	12	6	0
<i>lCr</i>	0	0	0	0	0	0	0	0	0	0	15

(b) IR

Abbildung 7.29: Wahrheitstabellen der inferierten Objektrelationen von Modell DR und IR für Sequenz 3 bei 5000 MC-SAT Iterationen. Abkürzungen der Objektrelationen gemäß Tabelle 3.1.

Zusammenfassung Die Objektklassifikation in Verkehrsszenen mit Markov-Logik-Netzen liefert für alle Modelle hervorragende Ergebnisse. Die oben beschriebenen Resultate zeigen allerdings auch die Überlegenheit der mit relationaler Information erweiterten Markov-Logik-Netzen IROK und ROK gegenüber dem einfachen Klassifikationsmodell OK auf. Dies kann anhand aller Gütemaße abgeleitet werden, so dass neben den Werten und der Standardabweichung der CLL auch sehr hohe Genauigkeiten mit im Mittel über 70% bereits für die unterste Taxonomieebene erreicht werden. Betrachtet man die für Fahrerassistenzsysteme

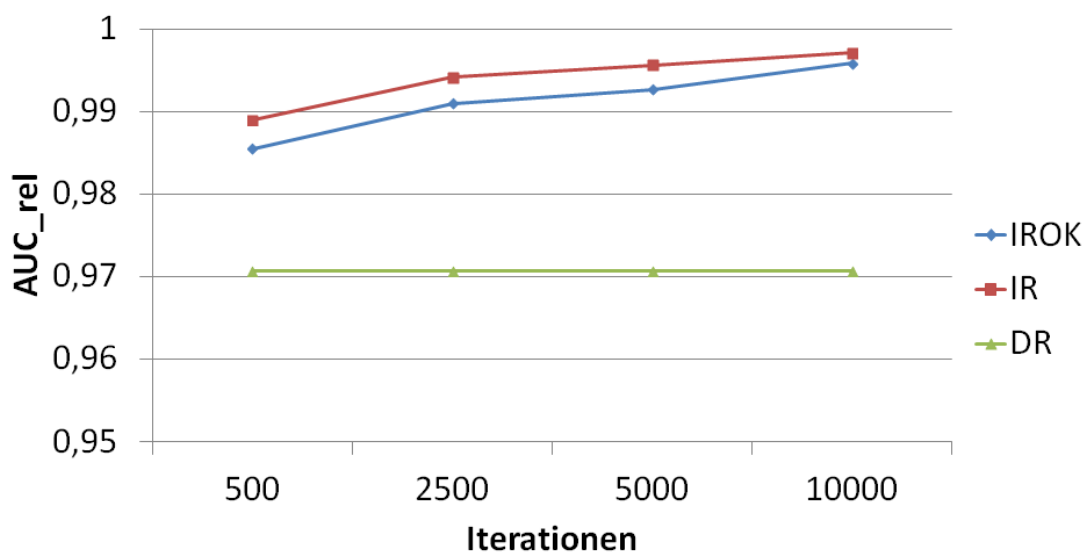


Abbildung 7.30: Mehrklassen-AUC AUC_{rel} der in Sequenz 3 vorkommenden Objektrelationen.

wichtige Ebene eine Stufe höher, liegen die Werte der relationalen Modelle mit über 95% in einem ausgezeichneten Bereich, während in dem simpleren Modell OK keine signifikante Verbesserung festgestellt werden kann. Betrachtet man nur die relationalen Modelle ist ein leichter Vorteil des Modells IROK festzustellen, der allerdings mit einem höheren Modellierungs- und Berechnungsaufwand erkauft wird. Liegt der Schwerpunkt der Objektklassifikation auf hoher Effizienz, kann daher dem Modell ROK der Vorzug gegeben werden, obwohl im Allgemeinen das Modell IROK die bestmögliche Klassifikation erreicht.

Etwas differenzierter müssen die vorliegenden Ergebnisse für die Objektrelationen betrachtet werden. Obwohl wiederum für alle Modelle sehr gute Ergebnisse erzielt werden, bestätigen die erste und dritte Sequenz den Vorteil der Modelle, die gegenüber dem Modell DR, welches deterministische Relationen verwendet, die Relationen mit den Daten inferieren. Sie weisen eine geringere Standardabweichung der conditional loglikelihood auf und erreichen auch in der Mehrklassen-Genauigkeit AUC_{rel} hervorragende Werte, bei der das Modell DR, wenn auch auf hohem Niveau, stagniert. Der Anstieg der Klassifikationsgüte für die komplexeren Modelle mit höherer Iterationszahl indiziert, dass möglicherweise auftretende, komplexere Verkehrsszenen mit mehr Teilnehmern oder -klassen besser mit den relationalen Modellen abgebildet werden können. Obwohl das Modell IR gegenüber dem Modell IROK Vorteile in den Kennzahlen aufweist, kann letzteres ohne Einschränkung direkt für Objekte und Relationen eingesetzt werden, so dass es das umfassendste Modell mit gleichzeitig hoher Güte darstellt. Die vergleichsweise schlechten Resultate aller Modelle für die relationalen Beziehungen in der zweiten Sequenz sind auf die Schwierigkeiten bei der Modellierung von Crossing-

Relationen, gegeben die Sensorwerte zurückzuführen. Die hohe Schwankung der Werte des Modells DR, sichtbar in der $CLLstd_{rel}$ und die sinkende Genauigkeit in AUC_{rel} und ACC_{rel} aller Modelle verdeutlichen diesen Einfluss, da für die gegebene Bildfolge mehrere Ausfälle in den aus den Sensorwerten geschätzten Geschwindigkeiten vorliegen. Trotzdem kann festgehalten werden, dass die schwächere Leistung im Bezug auf die Relationen keinen Einfluss auf die Güte der Objektklassifikation hat.

Zusammenfassend lässt sich sagen, dass das vorgestellte MLN mit dem Modell IROK die sinnvollste Wahl für Inferenz einer Situationsbeschreibung von Verkehrsszenen darstellt. Obwohl eine Unterteilung in zwei Modelle, jeweils für Objekte (IROK) und Relationen (IR) möglich wäre, würde dies nur eine geringe Verbesserung der Gesamtergebnisse zur Folge haben, aber einen doppelten Inferenzaufwand bedeuten. Das Modell IROK kann bei der Inferenzanfrage alle Objekte und deren Relationen inferieren und bietet mit der immanenten Taxonomie die Möglichkeit, Ergebnisse für die jeweilige Szene auf unterschiedliche Abstraktionsebenen auszugeben. Da der Inferenzaufwand auf aktueller Hardware vergleichsweise aufwandsintensiv ist, können einfachere Modelle, wie beispielsweise das vorgestellte Modell ROK (evtl. auch OK und DR) für die Inferenz genutzt werden. Da weniger Informationen inferiert werden, skaliert der Berechnungsaufwand stark (s. Tabelle 7.9), so dass dieser Gewinn den mangelnden Informationsgehalt und die teilweise schlechteren Klassifikationsergebnisse rechtfertigt.

8 Zusammenfassung und Ausblick

Die vorliegende Arbeit präsentiert ein auf Markov-Logik-Netzen basierendes Modell, mit welchem eine mit Wahrscheinlichkeiten versehene Situationsbeschreibung einer variablen Verkehrsszene probabilistisch-logisch inferiert wird. Dabei werden Klasse und Paarmodelle aller interessierenden segmentierten Objekthypothesen der Szene in einem einzigen Inferenzschritt geschätzt.

Der verwendete probabilistisch-logische Formalismus Markov-Logik-Netze verknüpft Prädikatenlogik erster Stufe und Markov-Netze miteinander. Für gegebene Evidenz- und Anfragevariablen kann ein MLN als ein Template zur Erstellung eines belegten Markov-Netzes betrachtet werden, auf welchem klassische probabilistische Inferenzalgorithmen angewendet werden können. Auf diese Weise gelingt es diesem Formalismus die aussagenlogische Ausdrucksfähigkeit der probabilistischen graphischen Modelle zu erweitern, was ein enormer Vorteil für die Inferenz von Beschreibungen variabler Verkehrssituationen darstellt. In der vorliegenden Arbeit werden Markov-Logik-Netze auf Anwendbarkeit als Wissensrepräsentationssprache im Bereich der kognitiven Automobile untersucht.

Zur Festlegung des im Rahmen der Arbeit verwendeten Vokabulars zur Beschreibung einer Verkehrssituation wird eine Ontologie vorgestellt. Diese repräsentiert explizit Wissen über Objekte, Objektattribute und Objektrelationen. Dabei sind die Objektkonzepte in einer taxonomischen Struktur eingebettet, was eine spätere Objektklassifikation auf verschiedenen Detaillierungsebenen erlaubt. Auf Basis der Ontologie werden mehrere Modelle zur Schätzung der darin definierten Objektklassen und Objektrelationen entwickelt.

Außer dem Modell „Direkte Objektrelationen“, welches mit Hilfe einer Definitionsmatrix deterministisch spezifiziert wird, bestehen alle anderen verwendeten Modelle aus prädikatenlogischen Formeln, die in Axiome und Regeln unterteilt sind. Die Gewichte zu den Regeln werden mit Hilfe eines aufgrund einer Vergleichsuntersuchung ausgewählten diskriminativen Parameterlernverfahrens geschätzt. Während der Trainingsphase der einzelnen Modelle werden anhand der Trainingsdaten gleichzeitig für alle in der Ontologie eingeführten Objektklassen beziehungsweise Objektrelationen Definitionen gelernt.

Durch Kombination der Axiome und gewichteten Regeln der Modelle werden mehrere Markov-Logik-Netze erstellt, die zur Inferenz der Objektklassen und Objektrelationen in der Verkehrsszene eingesetzt werden. Als Eingangsdaten dienen

dabei sensoruell gemessene Werte für die Größe der Bounding-Box, Bewegung und Position im Raum für jede segmentierte 3D-Objekthypothese einer Verkehrsszene. Diese werden weiterverarbeitet, qualitativ abstrahiert und in Form von Evidenzdaten für die Inferenz aufbereitet. Gegeben ein MLN, Evidenzdaten und eine Menge von Anfragevariablen, werden mittels Inferenz anhand des belegten Markov-Netzes bedingte Wahrscheinlichkeiten für alle Grundatome der Anfrageprädikate geschätzt. Diese Wahrscheinlichkeiten liefern somit den geschätzten Wahrheitsgrad der durch das jeweilige Grundatom gemachte Aussage.

Das MLN OK modelliert die Objektklassen in Abhängigkeit von den einfachen Objektattributen. Das MLN ROK setzt darüber hinaus die Objektklassen mit den Objektrelationen der Ontologie in Beziehung. Das Modell IR schätzt die Objektrelationen anhand automatisch gelernter Definitionen, während das Modell DR diese anhand manuell erstellter Objektattributenmatrix eindeutig definiert. Die durch das Modell DR im Vorfeld ausgewertete Objektrelationen werden den Evidenzdaten hinzugefügt und als Eingangsdaten bei der Inferenz der Objektklassen mit dem MLN ROK verwendet. Das Modell MLN ROK wird durch das Modell IR zu MLN IROK erweitert, wodurch mit diesem sowohl die Objektklassen als auch die Objektrelationen in einem Inferenzschritt geschätzt werden können.

Die entwickelte MLN-Modelle zur Klassifikation von Objektklassen und/oder Objektrelationen einer Verkehrsszene werden anhand exemplarisch verwendeter Videosequenzen evaluiert. Die Klassifikationsergebnisse aller Modelle weisen hervorragende Werte auf. Die erzielten Inferenzergebnisse zeigen allerdings auch die Überlegenheit der mit relationaler Information erweiterten Markov-Logik-Netzen IROK und ROK gegenüber dem einfachen Klassifikationsmodell OK. Dabei liefert das MLN IROK die besten Resultate mit einer Klassifikationsgenauigkeit von bis zu 98% auf realen Daten.

Der vorgestellte Ansatz ist darüber hinaus bei verschiedenen Sensorarchitekturen einsetzbar und kann Situationen mit variabler Anzahl unterschiedlicher Verkehrsteilnehmer sowie auch aus unterschiedlichen Verkehrsraumumgebungen beschreiben. Weiterhin wird bei der Modellierung der relativen Objektattribute und damit auch für die Objektrelation ein egozentrischer Bezugsrahmen ausgewählt, welcher für eine nachfolgende Interpretation der Intentionen von Verkehrsteilnehmern besonders geeignet ist.

Das vorgestellte Verfahren ermöglicht die Inferenz einer ganzheitlichen, mit Wahrscheinlichkeiten versehenen Situationsbeschreibung im Verkehrsraum aus unsicherheitsbehafteten Sensordaten, welche als Grundlage einer künftigen zentralisierten Architektur für Fahrerassistenzsysteme dienen kann.

Im Rahmen dieser Arbeit wurde die Eignung probabilistisch-logischer Methoden, im Speziellen der Markov-Logik, zur Inferenz von Situationsbeschreibungen aus

sensoriell erfassten Umfelddaten untersucht. Insbesondere wurde deutlich, wie die formalisierte Einbringung von Vorwissen den Schritt von isolierter Objektklassifikation hin zur Szeneninterpretation ermöglicht.

Trotz der vielversprechenden Ergebnisse mit experimentell gewonnenen Daten sind noch Verbesserungen der beschriebenen Ansätze und Modelle vorstellbar.

- Die Wissensmodellierung kann durch zusätzliche Ontologiebegriffe vertieft werden, so dass Situationen detaillierter beschrieben werden können. Denkbar wäre eine Erweiterung durch Begriffe der Straßeninfrastruktur oder auch durch eine Taxonomie der Relationen, in welche weitere spezifischere Objektrelationen definiert werden.
- Die vorgestellten Modelle nutzen die in zeitlichen Prozessen enthaltene Information derzeit nur indirekt durch die Sensordatenvorverarbeitung, beispielsweise der zeitlichen Verfolgung mittels dynamischer Zustandsschätzung. Eine Vorgehensweise ist zu entwickeln, die die explizite Berücksichtigung von mehreren Zeitschritten bei der Inferenz der Situationsbeschreibung ermöglicht.
- Die Komplexität der untersuchten Modelle erlaubt zum gegenwärtigen Zeitpunkt keine Inferenz in Echtzeit. Forschungsarbeiten zu dem Thema der Algorithmenintegration in neuer Automobilhardware und im Bereich der effizienten Inferenzalgorithmen könnten allerdings einen Einsatz in mittelfristiger Zeit ermöglichen.
- Für die Evaluierung der vorgestellten Modelle stand nur eine begrenzte Anzahl von ausgewerteten und annotierten Sequenzen zur Verfügung. Durch eine ausführliche statistische Untersuchung mit experimentell gewonnenen Daten könnten wertvolle Einsichten gewonnen werden, wie die vorgestellten Modelle zu erweitern sind.

Literaturverzeichnis

- [Abu10] T. Abudawood und P. A. Flach: *Learning Multi-class Theories in ILP*. In: *ILP*, S. 6–13, 2010.
- [Are04] M. Arens, A. Ottlik und H. H. Nagel: *Using Behavioral Knowledge for Situated Prediction of Movements*. In: *KI*, S. 141–155, Springer-Verlag, 2004.
- [Baa03] *The Description Logic Handbook: Theory, Implementation, and Applications*. In: *Description Logic Handbook*, F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi und P. F. Patel-Schneider (Hrsg.), Cambridge University Press, 2003.
- [Bac08] A. Bachmann und T. Dang: „Improving Motion-Based Object Detection by Incorporating Object-Specific Knowledge“. *International Journal of Intelligent Information and Database Systems (IJIDS), Special Issue on: Information Processing in Intelligent Vehicles and Road Applications 2* (2), S. 258–276, 2008.
- [Bac09] A. Bachmann und I. Lulcheva: *Combining Low-Level Segmentation with Relational Classification*. In: *International Conference on Computer Vision; IEEE Workshop on Visual Surveillance*, S. 1216–1221, 2009.
- [Ben08] B. Bennett, D. R. Magee, A. G. Cohn und D. C. Hogg: „Enhanced tracking and recognition of moving objects by reasoning about spatio-temporal continuity“. *Image Vision Comput.* **26** (1), S. 67–81, Jan. 2008.
- [Ber05] E. J. Bernstein und Y. Amit: *Part-Based Statistical Models for Object Classification and Detection*. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, S. 734–740, IEEE Computer Society, Washington, DC, USA, 2005.
- [Bes75] J. Besag: „Statistical Analysis of Non-Lattice Data“. *Journal of the Royal Statistical Society. Series D (The Statistician)* **24** (3), S. pp. 179–195, 1975.
- [Bib09] M. Biba: *Integrating Logic and Statistics: Novel Algorithms in Markov Logic Networks*. VDM Verlag Dr. Müller, 2009.

- [Bre11] R. Brehar, C. Fortuna, S. Bota, D. Mladenic und S. Nedevschi: *Spatio-temporal reasoning for traffic scene understanding*. In: *Proceedings of the IEEE Intelligent Computer Communication and Processing (ICCP)*, Aug 25-27 2011.
- [Bro09] F. Bromberg, D. Margaritis und V. Honavar: „Efficient Markov network structure discovery using independence tests“. *J. Artif. Int. Res.* **35** (1), S. 449–484, Juli 2009.
- [Car04] P. Carbonetto, N. de Freitas und K. Barnard: *A Statistical Model for General Contextual Object Recognition*. In: *ECCV*, S. 350–362, 2004.
- [Che10] A. Chechetka, D. Dash und M. Philipose: *Relational Learning for Collective Classification of Entities in Images*. In: *Statistical Relational Artificial Intelligence*, 2010.
- [Cle97] E. Clementini und P. Di Felice: „A Global Framework for Qualitative Shape Description“. *Geoinformatica* **1** (1), S. 11–27, 1997.
- [Coh01] A. G. Cohn und S. M. Hazarika: „Qualitative Spatial Representation and Reasoning: An Overview“. *Fundam. Inform.* **46** (1-2), S. 1–29, 2001.
- [Coh02] A. G. Cohn, D. R. Magee, A. Galata, D. C. Hogg und S. M. Hazarika: *Towards an Architecture for Cognitive Vision using Qualitative Spatio-Temporal Representations and Abduction*. In: *In Spatial Cognition III*, S. 232–248, Springer-Verlag, 2002.
- [Coh06] A. G. Cohn, D. C. Hogg, B. Bennett, V. E. Devin, A. Galata, D. R. Magee, C. J. Needham und P. E. Santos: *Cognitive Vision: Integrating Symbolic Qualitative Representations with Computer Vision*. In: *Cognitive Vision Systems*, S. 221–246, 2006.
- [Col02] M. Collins: *Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms*. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10, EMNLP '02*, S. 1–8, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002.
- [Dan93] S. Dance und T. Caelli: *On the Symbolic Interpretation of Traffic Scenes*. In: *ACCV93: Proceedings of the Asian Conference on Computer Vision*, S. 798–801, 1993.
- [Dic98] E. D. Dickmanns: „Vehicles Capable of Dynamic Vision: A New Breed of Technical Beings?“. *Artif. Intell.* **103** (1-2), S. 49–76, 1998.

- [Div09] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros und M. Hebert: *An Empirical Study of Context in Object Detection*. In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2009.
- [Dom06] P. Domingos: *Learning, Logic, and Probability: A Unified View*. In: *PRICAI*, S. 1, 2006.
- [Dom09] P. Domingos und D. Lowd: *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.
- [DP97] S. Della Pietra, V. Della Pietra und J. Lafferty: „Inducing Features of Random Fields“. *IEEE Trans. Pattern Anal. Mach. Intell.* **19** (4), S. 380–393, Apr. 1997.
- [Dze01] S. E. Dzeroski und S. Dzeroski: *Relational Data Mining*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [Elk12] C. Elkan: *Maximum Likelihood, Logistic Regression and Stochastic Gradient Training*, 2012. Course material.
- [EM08] N. Eric Maillot und M. Thonnat: „Ontology based complex object recognition“. *Image Vision Comput.* **26** (1), S. 102–113, Jan. 2008.
- [Faw06] T. Fawcett: „An introduction to ROC analysis“. *Pattern Recognition Letters* **27** (8), S. 861–874, 2006.
- [Fer08] C. Fernández, P. Baiget, X. Roca und J. González: „Interpretation of complex situations in a semantic-based surveillance framework“. *Image Commun.* **23** (7), S. 554–569, 2008.
- [Fri99] N. Friedman, L. Getoor, D. Koller und A. Pfeffer: *Learning Probabilistic Relational Models*. In: *IJCAI*, S. 1300–1309, 1999.
- [Fuc08] S. Fuchs: *A Comprehensive Knowledge Base for Context-aware Tactical Driver Assistance Systems*. Smart system technologies. Shaker, 2008.
- [Geh05] J. D. Gehrke, A. D. Lattner und O. Herzog: *Qualitative Mapping of Sensory Data for Intelligent Vehicles*. In: *Workshop on Agents in Real-Time and Dynamic Environments at the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, U. Visser, G. Lakemeyer, G. Vachtsevanos und M. Veloso (Hrsg.), S. 51–60, University of Edinburgh, Edinburgh, UK, 2005.

- [Gen87] M. R. Genesereth und N. J. Nilsson: *Logical foundations of artificial intelligence*. Morgan Kaufmann, San Francisco, 1987.
- [Geo06] B. Georis, M. Mazière, F. Brémond und M. Thonnat: *Evaluation and Knowledge Representation Formalisms to Improve Video Understanding*. In: *Proceedings of the International Conference on Computer Vision Systems (ICVS'06)*, New-York, NY, USA, January 2006.
- [Ger08] R. Gerber und H. H. Nagel: „Representation of occurrences for road vehicle traffic“. *Artif. Intell.* **172** (4-5), S. 351–391, 2008.
- [Get07a] L. Getoor und B. Taskar: *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.
- [Get07b] L. H. Getoor (Hrsg.): *Introduction to statistical relational learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. [u.a.], 2007.
- [GP03] A. Gomez-Perez, O. Corcho-Garcia und M. Fernandez-Lopez: *Ontological Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
- [Gör03] G. H. Görz (Hrsg.): *Handbuch der künstlichen Intelligenz*. Oldenbourg, München, 4., korr. Aufl.. Aufl., 2003.
- [Gru93] T. Gruber: „A translation approach to portable ontology specifications“. *Knowledge Acquisition* **5** (2), S. 199–220, 1993.
- [Gru02] M. Gruninger und J. Lee: „Ontology: Applications and Design“. *Commun. ACM* **45** (2), S. 39–41, 2002.
- [Gua98] N. Guarino: *Formal Ontology and Information Systems*. In: *Proc. of the 1st International Conference on Formal Ontologies in Information Systems and FOIS'98*, N. Guarino (Hrsg.), S. 3–15, IOS Press, Trento and Italy, June 1998.
- [Haa98] M. Haag: *Bildfolgenauswertung zur Erkennung der Absichten von Straßenverkehrsteilnehmern*, Bd. 193 d. Reihe *Dissertationen zur Künstlichen Intelligenz (DISKI)*. infix, Sankt Augustin, 1998.
- [Hae09] J. Haering und U. Wilhelm: *Situation-Interpretation as a Key Enabler for Cost-Effective and Low-Risk Driver Assistance Systems with High Collision Mitigation Capabilities*. In: *Proceedings of International Technical Conference on the Enhanced Safety of Vehicles*, Bd. 2009, S. –, 2009.

- [Hül11] M. Hülsen, J. M. Zöllner und C. Weiss: *Traffic Intersection Situation Description Ontology for Advanced Driver Assistance*. In: *2011 IEEE Intelligent Vehicles Symposium*, Nr. Iv, S. 991–997, IEEE, 2011.
- [Hot08] L. Hotz, B. Neumann und K. Terzic: *High-Level Expectations for Low-Level Image Processing*. In: *KI*, S. 87–94, Springer-Verlag, 2008.
- [How00] R. J. Howarth und H. Buxton: „Conceptual descriptions from monitoring and watching image sequences“. *Image Vision Comput.* **18** (2), S. 105–135, 2000.
- [Hub04] D. F. Huber, A. Kapuria, R. Donamukkala und M. Hebert: *Parts-Based 3D Object Classification*. In: *CVPR* (2), S. 82–89, 2004.
- [Hum09] B. Hummel: *Description Logic for Scene Understanding at the example of Urban Road Intersections*. Nr. 13. Universitätsverlag Karlsruhe, Schriftenreihe Mess- und Regelungstechnik, Dissertation, Universität Karlsruhe (TH), Apr. 2009.
- [Jay03] E. Jaynes und G. Bretthorst: *Probability Theory: The Logic of Science*. Cambridge University Press Cambridge:, 2003.
- [Kam08] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebel, F. v. Hundelshausen, O. Pink, C. Frese und C. Stiller: „Team AnnieWAY’s autonomous system for the 2007 DARPA Urban Challenge“. *J. Field Robot.* **25** (9), S. 615–639, Sep. 2008.
- [Kau96] H. Kautz, B. Selman und Y. Jiang: *A General Stochastic Approach to Solving Problems with Hard and Soft Constraints*. In: *The Satisfiability Problem: Theory and Applications*, S. 573–586, American Mathematical Society, 1996.
- [Kem10] A. Kembhavi, T. Yeh und L. S. Davis: *Why did the person cross the road (there)? scene understanding using probabilistic logic models and common sense reasoning*. In: *Proceedings of the 11th European conference on Computer vision: Part II, ECCV’10*, S. 693–706, Springer-Verlag, Berlin, Heidelberg, 2010.
- [Ker00] K. Kersting und L. D. Raedt: *Bayesian Logic Programs*. In: *Proceedings of the Work-in-Progress Track at the 10th International Conference on Inductive Logic Programming*, J. Cussens und A. Frisch (Hrsg.), S. 138–155, 2000.

- [Kok06] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, D. Lowd und P. Domingos: *The Alchemy System for Statistical Relational AI*. Techn. Ber., Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2006. See <http://alchemy.cs.washington.edu>.
- [Kum05] S. Kumar und M. Hebert: *A Hierarchical Field Framework for Unified Context-Based Classification*. In: *Tenth IEEE International Conference on Computer Vision (ICCV '05)*, IEEE (Hrsg.), Bd. 2, S. 1284 – 1291, October 2005.
- [Lac03] N. Lachiche und P. Flach: *Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves*. In: *Proc. 20th International Conference on Machine Learning (ICML'03)*, S. 416–423, AAAI Press, January 2003.
- [Laf01] J. Lafferty, A. McCallum und F. Pereira: *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In: *Proceedings of the 18th International Conference on Machine Learning*, S. 282–289, Morgan Kaufmann, San Fransisco, 2001.
- [Lai07] A. Laika und W. Stechele: *A review of different object recognition methods for the application in driver assistance systems*. In: *Proceedings of the Eight International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS '07*, S. 10–, IEEE Computer Society, Washington, DC, USA, 2007.
- [Lee06] S.-i. Lee, V. Ganapathi und D. Koller: *Efficient structure learning of Markov networks using L1 regularization*. In: *In NIPS*, 2006.
- [Len11] P. Lenz, J. Ziegler, A. Geiger und M. Roser: *Sparse scene flow segmentation for moving object detection in urban environments*. In: *Intelligent Vehicles Symposium*, S. 926–932, 2011.
- [Li07] L.-J. Li und L. Fei-Fei: *What, where and who? Classifying event by scene and object recognition..* In: *Proc. of IEEE Intern. Conf. in Computer Vision (ICCV).*, 2007.
- [Lip08] M. Lippi und P. Frasconi: *Markov Logic Improves Protein β -Partners Prediction*. In: *6th International Workshop on Mining and Learning with Graphs*, 2008.
- [Liu89] D. C. Liu und J. Nocedal: „On the limited memory BFGS method for large scale optimization“. *Math. Programming* **45** (3, (Ser. B)), S. 503–528, 1989.

- [Low07] D. Lowd und P. Domingos: *Efficient Weight Learning for Markov Logic Networks*. In: *PKDD*, S. 200–211, 2007.
- [Lul08] I. Lulcheva, B. Hummel und A. Bachmann: *Probabilistisch-logische Objektklassifikation für Verkehrsszenen*. In: *5. Workshop Fahrerassistenzsysteme*, 2008.
- [Mie04a] A. Miene: *Räumlich-zeitliche Analyse von dynamischen Szenen*. Dissertation, 2004.
- [Mie04b] A. Miene, A. D. Lattner, U. Visser und O. Herzog: *Dynamic-Preserving Qualitative Motion Description for Intelligent Vehicles*. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '04)*, S. 642–646, June 14-17 2004.
- [Mug94] S. Muggleton und L. D. Raedt: „Inductive Logic Programming: Theory and Methods“. *Journal of Logic Programming* **19/20**, S. 629–679, 1994.
- [Mug02] S. Muggleton: *Learning Structure and Parameters of Stochastic Logic Programs*. In: *ILP*, S. 198–206, 2002.
- [Nag04] H.-H. Nagel: „Steps Towards a Cognitive Vision System“. *AI Magazine* **25**, S. 31–50, 2004.
- [Neu03] B. Neumann und T. Weiss: *Navigating through Logic-Based Scene Models for High-Level Scene Interpretations*. In: *ICVS*, S. 212–222, 2003.
- [Neu08] B. Neumann und R. Möller: „On scene interpretation with description logics“. *Image Vision Comput.* **26** (1), S. 82–101, 2008.
- [Nie11] D. Nienhüser, T. Gumpp und J. M. Zöllner: *Relevance estimation of traffic elements using Markov logic networks*. In: *Intelligent Transportation Systems*, S. 1659 – 1664, 2011.
- [Pea06] J. Pearl: *Probabilistic reasoning in intelligent systems : networks of plausible inference*. The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, San Francisco, Calif., 10. [pr.]. Aufl., 2006.
- [Pel03] M. Pellkofer: *Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung*. Dissertation, 2003.
- [Poo93] D. Poole: „Probabilistic Horn Abduction and Bayesian Networks“. *Artificial Intelligence* **64** (1), S. 81–129, 1993.

- [Poo97] D. Poole: „The Independent Choice Logic for Modelling Multiple Agents Under Uncertainty“. *Artificial Intelligence* **94** (1-2), S. 7–56, 1997.
- [Poo06] H. Poon und P. Domingos: *Sound and Efficient Inference with Probabilistic and Deterministic Dependencies*. In: *AAAI*, 2006.
- [Pro04] R. Provine, C. Schlenoff, S. Balakirsky, S. Smith und M. Uschold: „Ontology-based methods for enhancing autonomous vehicle path planning“. *Robotics and Autonomous Systems* **49** (1-2), S. 123–133, 2004.
- [Rab89] L. R. Rabiner: *A tutorial on hidden Markov models and selected applications in speech recognition*. In: *Proceedings of IEEE*, Bd. 77, S. 257–286, IEEE, 1989.
- [Rae08] L. D. Raedt (Hrsg.): *Logical and Relational Learning*. Cognitive Technologies. Springer-Verlag, Berlin, Heidelberg, 2008.
- [Ric06a] M. Richardson und P. Domingos: „Markov logic networks“. *Machine Learning* **62** (1-2), S. 107–136, 2006.
- [Ric06b] M. Richardson und P. Domingos: „Markov logic networks“. *Machine Learning* **62** (1-2), S. 107–136, 2006.
- [Rie05] S. Riedel und E. Klein: *Genic Interaction Extraction with Semantic and Syntactic Chains*. In: *Learning in Logic Workshop, ICML*, 2005.
- [Rie08] S. Riedel und M.-R. Ivan: *Collective Semantic Role Labelling with Markov Logic*. In: *Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*, 2008.
- [Rus03] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik und D. D. Edwards: *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
- [Rus08] B. C. Russell, A. Torralba, K. P. Murphy und W. T. Freeman: „LabelMe: A Database and Web-Based Tool for Image Annotation“. *Int. J. Comput. Vision* **77** (1-3), S. 157–173, Mai 2008.
- [Sat95] T. Sato: *A Statistical Learning Method for Logic Programs with Distribution Semantics*. In: *International Conference on Logic Programming*, S. 715–729, 1995.
- [Sch07] P. H. Schmitt: *Formale Systeme*. Vorlesungsskript, KIT, 2007.

- [Sch08] S. Schoenmackers, O. Etzioni und D. S. Weld: *Scaling Textual Inference to the Web*. In: *Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*, 2008.
- [Sch10] J. Schneider: *Modellierung und Erkennung von Fahrsituationen und Fahrmanövern für sicherheitsrelevante Fahrerassistenzsysteme*. Univ.-Verlag TU Chemnitz, 2010.
- [Sel95] B. Selman, H. Kautz und B. Cohen: *Local Search Strategies for Satisfiability Testing*. In: *DIMACS SERIES IN DISCRETE MATHEMATICS AND THEORETICAL COMPUTER SCIENCE*, S. 521–532, 1995.
- [Sin05] P. Singla und P. Domingos: *Discriminative Training of Markov Logic Networks*. In: *AAAI*, S. 868–873, 2005.
- [Sin06] P. Singla und P. Domingos: *Entity Resolution with Markov Logic*. In: *ICDM*, S. 572–582, 2006.
- [Sin08] P. Singla, H. Kautz, J. Luo und A. Gallaher: *Discovery of Social Relationships in Consumer Photo Collections using Markov Logic*. In: *CVPR Workshop on Semantic Learning and Applications in Multimedia*, 2008.
- [Sin11] P. Singla und R. J. Mooney: *Abductive Markov Logic for Plan Recognition*. In: *AAAI*, 2011.
- [Sok09] M. Sokolova und G. Lapalme: „A systematic analysis of performance measures for classification tasks“. *Inf. Process. Manage.* **45** (4), S. 427–437, 2009.
- [Son11] Z. Song, Q. Chen, Z. Huang, Y. Hua und S. Yan: *Contextualizing object detection and classification*. In: *CVPR*, S. 1585–1592, 2011.
- [Sou11] C. R. C. Souza und P. E. Santos: *Probabilistic logic reasoning about traffic scenes*. In: *Proceedings of the 12th Annual conference on Towards autonomous robotic systems*, TAROS'11, S. 219–230, Springer-Verlag, Berlin, Heidelberg, 2011.
- [Sta04] S. Staab und R. Studer (Hrsg.): *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.
- [Sti07] C. Stiller, G. Färber und S. Kammel: *Cooperative Cognitive Automobiles*. In: *Proc. IEEE Intelligent Vehicles Symposium*, 2007.
- [Sti08] C. Stiller, S. Kammel, I. Lulcheva und J. Ziegler: „Probabilistische Methoden in der Umfeldwahrnehmung Kognitiver Automobile“. *at - Automatisierungstechnik* **56** (11), S. 563 – 574, Nov. 2008.

- [Sun06] Z. Sun, G. Bebis und R. Miller: „On-Road Vehicle Detection: A Review“. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28** (5), S. 694–711, 2006.
- [Tha88] P. Thayse, André [Hrsg.] ; Gochet (Hrsg.): *From standard logic to logic programming : introducing a logic based approach to artificial intelligence*. Wiley, Chichester [u.a.], 1988.
- [Tra08] S. D. Tran und L. S. Davis: *Event Modeling and Recognition Using Markov Logic Networks*. In: *ECCV* (2), S. 610–623, 2008.
- [Tsa07] N. Tsapatsoulis und S. Petridis: *Classifying Images from Athletics Based on Spatial Relations*. In: *Proceedings of the Second International Workshop on Semantic Media Adaptation and Personalization, SMAP '07*, S. 92–97, IEEE Computer Society, Washington, DC, USA, 2007.
- [Vac09] S. Vacek: *Videogestützte Umfelderkennung zur Interpretation von Verkehrssituationen für kognitive Automobile*. Univ.-Verlag Karlsruhe, 2009.
- [Ver06] D. Vernon: *The Space of Cognitive Vision..* In: *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, H. I. Christensen und H. H. Nagel (Hrsg.), Bd. 3948 d. Reihe LNCS, S. 7–26, Springer, 2006.
- [Wei04] W. Wei, J. Erenrich und B. Selman: *Towards Efficient Sampling: Exploiting Random Walk Strategies*. In: *AAAI*, S. 670–676, 2004.