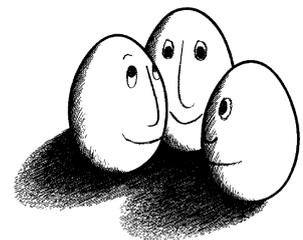


Diplomarbeit

**Verkehrsvorhersage unter  
Verwendung von Methoden  
des maschinellen Lernens**

Markus Kokott  
14. August 2012

Gutachter:  
Prof. Dr. Katharina Morik  
Dipl.-Inf. Christian Bockermann



Technische Universität Dortmund  
Fakultät für Informatik  
Lehrstuhl für Künstliche Intelligenz (LS VIII)  
<http://www-ai.cs.tu-dortmund.de>



„Statistics is the grammar of science“  
-- Karl Pearson

## Danksagung

Mein Dank geht an Prof. Dr. Katharina Morik und Dipl.-Inf. Christian Bockermann für die Betreuung meiner Arbeit, den Mitarbeitern des LS 8, weil sie für Fragen immer ein offenes Ohr hatten, meinen Korrektur-Lesern, die sich durch den Text gekämpft haben, meiner Freundin für ihre Geduld und Unterstützung sowie meinen Eltern, weil man Eltern nie genug danken kann.



# Zusammenfassung

Diese Arbeit beschäftigt sich mit der Vorhersage von Zustandsübergängen nach der Drei-Phasen-Verkehrstheorie. Zunächst werden Motifs als charakteristische Elemente in Verkehrsdaten von Induktionsschleifen identifiziert. Mithilfe solcher Motifs wird eine einfach zu erweiternde Repräsentation entwickelt, die sich als Eingabe für viele Verfahren des maschinellen Lernens eignet. Eine Evaluation der vorgestellten Repräsentation erfolgt mit einer Zahl von Lernverfahren exemplarisch anhand von Messdaten, die auf den Autobahnen im Ruhrgebiet gesammelt wurden.

## Abstract

This work deals with the forecasting of phase transitions in respect to the three-phase traffic theory. At first, motifs are identified to find characteristical elements in traffic data. Then, such motifs are used to develop a representation that is easily extensible and suitable for a number of machine learning algorithms. Exemplarily, this representation gets evaluated with a couple of learning algorithms for real traffic data gathered on German motorways in the Ruhr Area.



# Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	xi
Notation und Symbole	xiii
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>7</b>
2.1 Mathematik . . . . .	7
2.1.1 Zeitreihen . . . . .	7
2.1.2 Quantile . . . . .	10
2.1.3 Euklidischer Abstand . . . . .	11
2.1.4 Standardisierung von Zufallsvariablen . . . . .	12
2.1.5 Stichprobenziehung . . . . .	12
2.2 Informatik . . . . .	13
2.2.1 Datenstrom . . . . .	13
2.2.2 Fensterung . . . . .	14
2.2.3 Maschinelles Lernen . . . . .	15
2.2.4 Bestimmung der Vorhersagegüte eines Modells . . . . .	16
2.2.5 Kreuzvalidierung . . . . .	18
2.2.6 Parameteroptimierung . . . . .	19
2.3 Verkehrsdynamik . . . . .	20
2.3.1 Makroskopische Verkehrsgrößen . . . . .	21
2.3.2 Verkehrsphasen . . . . .	22
2.3.3 Zustandsübergänge . . . . .	24
<b>3 Datenbasis</b>	<b>27</b>
3.1 Das Sensornetzwerk . . . . .	27
3.2 Erfasste Größen . . . . .	28
3.3 Metadaten des Datensatzes . . . . .	29
3.4 Abgeleitete Größen . . . . .	30
3.5 Bestimmung der Verkehrsphase . . . . .	31
3.6 Festlegung der Ausgaben $y$ des Datensatz . . . . .	32
<b>4 Motifs</b>	<b>33</b>
4.1 Ursprung . . . . .	35

4.2	Algorithmus zur Extraktion von Motifs . . . . .	36
4.2.1	Vorüberlegungen . . . . .	36
4.2.2	Vorverarbeitung . . . . .	37
4.2.3	Finden von Übereinstimmungen . . . . .	38
4.2.4	Extraktion von Motifs . . . . .	42
4.2.5	Parameter für die Motif-Extraktion . . . . .	42
<b>5</b>	<b>Repräsentation des Datenstroms</b>	<b>45</b>
5.1	Wortvektor-Analogie . . . . .	46
5.2	TF*IDF . . . . .	48
5.2.1	Parameter für die Wortvektor-Repräsentation . . . . .	49
<b>6</b>	<b>Experimente</b>	<b>51</b>
6.1	Datensätze . . . . .	51
6.2	Aufbau des Experiments . . . . .	56
6.3	Vergleich von Lernalgorithmen . . . . .	56
6.3.1	Verwendete Lernalgorithmen . . . . .	57
6.3.2	Resultate . . . . .	67
6.4	Einfluss des Vorhersagehorizonts . . . . .	71
6.5	Klassifikationsgüte in den einzelnen Clustern . . . . .	73
6.6	Verwendung von Modellen zur Vorhersage für andere Zählschleifen . .	74
6.7	Evaluation mit Daten anderer Zeiträume . . . . .	75
6.8	Anreicherung der Wortvektoren mit zusätzlichen Informationen . . .	76
6.9	Vergleich mit anderen Ansätzen . . . . .	77
<b>7</b>	<b>Fazit und Ausblick</b>	<b>81</b>
	<b>Literaturverzeichnis</b>	<b>85</b>

# Abbildungsverzeichnis

1.1	Allgemeine Lernaufgabe . . . . .	4
1.2	Übersicht der Arbeit . . . . .	4
2.1	Beispiele für Zeitreihen . . . . .	8
	(a) Verbraucherpreisindex . . . . .	8
	(b) Umsatz Einzelhandel . . . . .	8
	(c) Anzahl Sonnenflecken . . . . .	8
2.2	Phasen des maschinellen Lernens . . . . .	16
2.3	Staufronten eines Verkehrsmusters . . . . .	23
2.4	Entstehung einer Schockwelle . . . . .	26
3.1	Positionen der Zählschleifen . . . . .	28
3.2	Bestimmung der Verkehrsphase . . . . .	31
4.1	Beispiel für ein Motif . . . . .	34
4.2	Übersicht: Extraktion von Motifs . . . . .	36
4.3	Fensterungen zur Extraktion von Motifs . . . . .	38
4.4	Beispiel: Übereinstimmungen von Teilfolgen . . . . .	39
4.5	Prinzip des Ordnungsstrahls . . . . .	41
5.1	Übersicht: Repräsentation . . . . .	45
5.2	Erzeugung von Wortvektoren . . . . .	47
6.1	Histogramm für die Anzahl an Fahrzeugen pro Minute des Tages . . . . .	52
6.2	Ermittlung der Anzahl von Clustern unter den Zählschleifen . . . . .	54
6.3	Grundsätzlicher Aufbau des Basisexperiments . . . . .	57
6.4	Klassifikation mit $k$ nächsten Nachbarn . . . . .	58
6.5	Klassifikationsgüte von $k$ nächste Nachbarn in Abhängigkeit von $k$ . . . . .	59
6.6	Klassifikation mit einem Baum . . . . .	60
6.7	Klassifikationsgüte von Random Forests in Abhängigkeit von $I$ und $K$ . . . . .	61
6.8	Trennende Hyperebene im $\mathbb{R}^2$ . . . . .	62
6.9	Verlauf der Klassifikationsgüte des SVM Modells während der Parameteroptimierung . . . . .	64
6.10	Beispiel für ein neuronales Netz . . . . .	65
6.11	Verlauf der Klassifikationsgüte bei einem neuronalen Netz während der Parameteroptimierung . . . . .	66
6.12	Vergleich der Klassifikationsgüte von den verwendeten Lernverfahren . . . . .	68

## *Abbildungsverzeichnis*

6.13 Vorhersagegüte bzgl. des Vorhersagehorizonts . . . . .	71
6.14 Klassifikationsgüte für die Datensätze aus unterschiedlichen Clustern	73
6.15 Klassifikationsgüte für Beispiele aus späteren Monaten . . . . .	76
6.16 Klassifikationsgüte für drei Ansätze zur Vorhersage . . . . .	78

# Tabellenverzeichnis

2.1	Beispiel für eine Konfusionsmatrix . . . . .	18
3.1	Attribute des Datensatzes . . . . .	29
4.1	Parameter für die Extraktion von Motifs . . . . .	43
5.1	Parameter für die Repräsentation durch Wortvektoren . . . . .	50
6.1	Eigenschaften der unterschiedlichen Gruppen von Zählschleifen . . . . .	54
6.2	Eigenschaften der in den Experimenten verwendeten Zählschleifen . . . . .	55
6.3	Übersicht über die Datensätze . . . . .	56
6.4	Vergleich von Lernalgorithmen: Konfusionsmatrizen . . . . .	70
	(a) Naive Bayes . . . . .	70
	(b) $k$ nächst Nachbarn . . . . .	70
	(c) Random Forests . . . . .	70
	(d) Support Vector Machines . . . . .	70
	(e) Konfusionsmatrix für ein neuronales Netz . . . . .	70
6.5	Konfusionsmatrix für einen Vorhersagehorizont von 25 Minuten . . . . .	72
6.6	Klassifikationsgüte von Modellen für Datensätze fremder Zählschleifen . . . . .	75
6.7	Vergleich von Sensitivität und Genauigkeit unterschiedlicher Verfahren . . . . .	78
6.8	Konfusionsmatrizen für die Vorhersage von freiem und gestautem Verkehr anhand von Tageszeit und Wochentag . . . . .	79
	(a) Zählschleife Cara . . . . .	79
	(b) Zählschleife DoWe . . . . .	79
	(c) Zählschleife DuWe . . . . .	79
	(d) Zählschleife EsAl . . . . .	79
	(e) Zählschleife MüWi . . . . .	79
	(f) Zählschleife WiSt . . . . .	79



# Notation und Symbole

$\mathbb{C}$	Menge der komplexen Zahlen.
$\mathbb{N}$	Menge der natürlichen Zahlen.
$\mathbb{R}$	Menge der reellen Zahlen.
$C$	Teilfolge einer Werte- bzw. Zeitreihe.
$F$	(aktives) Fenster für eine Werte- bzw. Zeitreihe.
$T$	Zeitreihe.
$W$	Wertereihe.
$\mathbf{X}$	Matrix mit Beispiel-Vektoren $\langle \mathbf{x}_0, \dots, \mathbf{x}_m \rangle$ .
$\mathbf{Y}$	Vektor mit Ausgaben $\langle y_0, \dots, y_m \rangle^T$ zu $\mathbf{X}$ .
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	Parametervektoren; dient als Parametrisierung für ein Modell.
$c_i$	Element $i$ einer Teilfolge $C$ .
$\bar{c}_i$	Mittelwert einer konsekutiven Folge von Elementen $\{c_{i-h}, \dots, c_i\}$ .
$\mathbf{x}$	Beispiel-Vektor $\langle x_0, \dots, x_d \rangle$ .
$x_i$	Attribut $i$ eines Beispiels $\mathbf{x}$ .
$\mathbf{x}_i$	Beispielvektor $i$ aus $\mathbf{X}$ .
$y$	Ausgabe zu $\mathbf{x}$ .
$y_i$	Ausgabe zu $\mathbf{x}_i$ .
$\hat{y}$	Vorhersage für $y$ .
$\hat{y}_i$	Vorhersage für $y_i$ .
$Q$	Verkehrsfluss.
$Q(x, t_i)$	Verkehrsfluss an der Stelle $x$ im Zeitintervall $[t_i - \Delta t; t_i]$ .
$Q_{zs}(t_i)$	Verkehrsfluss an der Zählschleife $zs$ im Zeitintervall $[t_i - \Delta t; t_i]$ .
$\rho$	Verkehrsdichte.
$\rho(x, t_i)$	Verkehrsdichte zum Zeitpunkt $t$ an einem Querschnitt $x$ .
$\rho_{zs}(t_i)$	Verkehrsdichte zum Zeitpunkt $t$ an der Zählschleife $zs$ .
$v$	Geschwindigkeit.
$v(x, t_i)$	Durchschnittsgeschwindigkeit im Zeitintervall $[t_i - \Delta t; t_i]$ an der Stelle $x$ .
$v_{zs}(t_i)$	Durchschnittsgeschwindigkeit im Zeitintervall $[t_i - \Delta t; t_i]$ an der Zählschleife $zs$ .

$\Delta$	Kennzeichnet ein Intervall.
$\phi$	Relativer Anteil (z. B. Parameter für eine Quantil-Abfrage); $\phi \in \mathbb{R} \wedge \phi \in [0; 1]$ .
$d$	Abstand zweier Vektoren bzw. Beispiele.
$h$	Horizont; Intervallgröße für einen Zeitraum.
$l$	Länge.
$R$	Radius um einen Punkt.
$t$	Zeitpunkt.
$w$	Breite eines Fensters $F$ .
$dist_{euklid}(\mathbf{x}_i, \mathbf{x}_j)$	Euklidische Distanz zwischen $\mathbf{x}_i$ und $\mathbf{x}_j$ .
$sim(C, C')$	Ähnlichkeit zweier Teilfolgen $C$ und $C'$ .
$\varepsilon(\hat{f}(\mathbf{x}))$	Fehler von $\hat{f}(\mathbf{x})$ .
$\hat{f}(\mathbf{x})$	Funktion zur Bestimmung von $\hat{y}$ für $\mathbf{x}$ .
$L(y, \hat{f}(\mathbf{x}))$	Verlustfunktion zum Vergleich von $y$ und $\hat{y}$ .
$sign(f(\mathbf{x}))$	Vorzeichen von $f(\mathbf{x})$
$df(j)$	<i>document frequency</i> ; Dokumenthäufigkeit von Wort $j$ .
$idf(j)$	<i>inverse document frequency</i> ; inverse Dokumenthäufigkeit von Wort $j$ .
$tf_i(j)$	<i>term frequency</i> ; Vorkommenshäufigkeit von Wort $j$ in Dokument $i$ .
$tf_i idf(j)$	TF*IDF-Maß des Wortes $j$ bezüglich des Dokuments $i$ .
$\omega(C)$	Wortvektorrepräsentation einer Teilfolge $C$ .

# Einleitung

Die hohe Bevölkerungsdichte sowie die wirtschaftlich gute Lage des Ruhrgebiets sorgen für eine hohe Verkehrsnachfrage auf den Autobahnen dieser Region. Dabei ist ein monotones Wachstum des Verkehrsaufkommens zu beobachten. Seit 1980 hat der Personenverkehr auf den Autobahnen NRWs um 50% zugenommen. Im selben Zeitraum hat sich der Güterverkehr auf den Straßen sogar annähernd verdoppelt [35]. In den letzten fünf Jahren ist der Fahrzeugbestand der Bundesrepublik Deutschland um 1,75 Millionen PKW angewachsen<sup>1</sup>. Trotz des Bevölkerungsrückgangs in Deutschland wird auch für die Zukunft ein weiterer Anstieg des Fahrzeugbestands prognostiziert, da z. B. der Freizeit-Verkehr und die Stadtflucht zunehmen [11].

Ohne geeignete Gegenmaßnahmen führt dieser stetige Zuwachs zu immer häufigeren und längeren Staus auf den Autobahnen. In der Rangliste für staubedingte Reisezeitverlängerungen belegt das Ruhrgebiet bereits heute mit 23,2% den dritten Platz in Europa – direkt hinter den Metropolen London (25%) und Paris (31%) [35]. Dies hat unmittelbare Folgen:

- **Größere Umweltbelastungen** durch erhöhte CO<sub>2</sub>-Emissionen [7].
- **Steigende Unfallgefahr** durch die erhöhte Verkehrsdichte [87].
- **Gesundheitliche Schäden** der Verkehrsteilnehmer (z.B. durch erhöhten Stresspegel [37]).
- **Gesamtwirtschaftliche Schäden** in Milliardenhöhe [35].

Eine offensichtliche Gegenmaßnahme ist die bauliche Schaffung von zusätzlichen Kapazitäten auf den Autobahnen. Hier ist jedoch zu beachten, dass Baumaßnahmen langwierige und kostenintensive Prozesse sind und das Ruhrgebiet bereits ein sehr dichtes Autobahnnetz besitzt. Außerdem kann die Erhöhung der Verkehrskapazität zu einer überproportionalen Erhöhung der Verkehrsnachfrage und somit zu weiteren Staus führen [26].

Die Alternative hierzu ist die Verkehrsbeeinflussung. Neben der festen Blechschilderung erlaubt die immer größere Verbreitung von Telematikanlagen auf den Autobahnen eine dynamische Beeinflussung des Verkehrs. Die wichtigsten Maßnahmen hierfür sind [35]:

---

<sup>1</sup>Den jährlichen Pressemitteilungen des Kraftfahrt-Bundesamts zum Fahrzeugbestand in Deutschland entnommen:

[http://www.kba.de/cln\\_030/nn\\_189178/DE/Presse/PressemitteilungenStatistiken/pressemitteilungenstatistiken\\_\\_node.html](http://www.kba.de/cln_030/nn_189178/DE/Presse/PressemitteilungenStatistiken/pressemitteilungenstatistiken__node.html)

## 1 Einleitung

- **Netzbeeinflussung:** Verkehrsteilnehmer werden auf weniger frequentierte Strecken umgeleitet,
- **Streckenbeeinflussung:** Errichtung von Wechselverkehrszeichen für Überholverbote, Geschwindigkeitsbegrenzungen usw.,
- **temporäre Seitenstreifenfreigabe:** zur Entlastung kann der für Pannen vorgesehene Sicherheitsbereich am Rand der Autobahn für den normalen Verkehr geöffnet werden,
- **Zufussreglung:** mit Ampeln an den Auffahrten wird der Zustrom an Fahrzeugen eingedämmt,
- **variable Fahrstreifenzuteilung:** um das Auffahren zu erleichtern wird die rechte Spur kurz vor einer Auffahrt für den Durchgangsverkehr gesperrt.

Um oben genannte Maßnahmen sinnvoll und rechtzeitig zu nutzen, ist eine akkurate Vorhersage für die kurzfristige Entwicklung der Verkehrssituation erforderlich [17]. Die Informatik bietet insbesondere im Bereich des maschinellen Lernens Verfahren die zur Lösung ähnlicher Probleme in vielen Anwendungsdomänen erfolgreich eingesetzt werden. Deshalb beschäftigt sich die vorliegende Arbeit mit der Untersuchung und Bewertung von Verfahren des maschinellen Lernens für die kurzfristige Verkehrsprognose.

Verkehrsvorhersagemodelle werden grob in zwei Kategorien eingeteilt: In der Kategorie der *mikroskopischen Modelle* wird das Verhalten einzelner Verkehrsteilnehmer betrachtet. Modelle dieser Kategorie sind für das Verständnis von Verkehrssystemen gut geeignet. Mit dem *Nagel-Schreckenberg-Modell* wurde z.B. das Phänomen des Phantomstaus (oder auch „*Stau aus dem Nichts*“) auf eine Reihe kollektiver Bremsmanöver zurückgeführt [77]. Die dieser Arbeit zugrunde liegende Datenbasis erlaubt jedoch keine Betrachtung des Verkehrs mit dieser Auflösung, weshalb mikroskopische Modelle nicht weiter betrachtet werden.

Wichtiger für diese Arbeit sind *makroskopische Verkehrsmodelle*. Anstatt die Daten einzelner Fahrzeuge auszuwerten wird die Annahme getroffen, dass das Verhalten einzelner Verkehrsteilnehmer von gemeinsamen Rahmenbedingungen (wie dem Wetter, der Verkehrsdichte, den Sichtverhältnissen usw.) abhängt [45]. Aus diesem Grund werden gemeinsame Daten über Gruppen erhoben. Meistens werden Fahrzeuge, die den selben Streckenabschnitt kurz nacheinander durchfahren zusammengefasst. Darüber hinaus werden Fahrzeuge häufig anhand ihrer Länge in unterschiedliche Fahrzeugklassen (z.B. PKW und LKW) eingeteilt.

Viele Modelle nutzen die Periodizität von Verkehrsdaten aus. Bereits 1979 haben Ahmed und Cook *autoregressive integrated moving average* Modelle (*ARIMA*) verwendet, um einzelne Verkehrsgrößen wie den Fluss zu bestimmen [2]. Da die Vorhersage für typische Verkehrssituationen (d.h. es treten keine unerwarteten Ereignisse wie Unfälle auf) gute Ergebnisse erzielt, ist die Verwendung von *ARIMA*-Modellen

bis heute weit verbreitet [16]. Es wurden z.B. Clustering-Verfahren mit ARIMA kombiniert, um Messstationen mit ähnlichen Verläufen zusammenzufassen (z.B. mit *k-means-Clustering* im *ATHENA-Modell* [61] oder mithilfe von *Kohonenkarten* bei der *KARIMA-Methode* [98]). Darüber hinaus existieren auch *saisonale ARIMA-Modelle*, um wiederkehrende Abhängigkeiten der Verkehrsgrößen (z.B. Ferienzeiten) zu erfassen [101]. Für die kurzfristige Verkehrsvorhersage sind ARIMA-Modelle i.d.R. jedoch zu unflexibel gegenüber unerwarteten Ereignissen.

Für die Regression einzelner Verkehrsgrößen (wie dem Verkehrsfluss, der Verkehrsdichte oder der durchschnittlichen Geschwindigkeit) wurden u. a. bereits *nächste Nachbarn* [24, 89], *Support Vector Machines* [25, 16], *Markov-Ketten* [104, 91] oder *Neuronale Netze* [51] untersucht. Wie Vlahogianni et. al in [97] jedoch feststellen, sind bislang verfügbare Modelle für mehr als eine Verkehrsgröße komplexer und weniger gut geeignet als spezialisierte Modelle für die Vorhersage nur einer Verkehrsgröße.

Mit *JamBayes* wurde für den Großraum Seattle ein *Bayesisches Netz* zur Vorhersage der Verkehrsphase entwickelt [46]. Hier zeigt sich die große Komplexität der Aufgabe – neben den offensichtlichen Einflussgrößen wie dem Verkehrsfluss oder der durchschnittlichen Geschwindigkeit auf einem Autobahnabschnitt, spielen auch diverse weniger gut erfassbare Größen eine Rolle. So werden von diesem Modell Wetter- und Sichtverhältnis genauso betrachtet wie die Ansetzung von Großereignissen.

Bei der Betrachtung von *JamBayes* zeigt sich, dass es nicht ausreicht, ein Modell für die Vorhersage einzelner Verkehrsgrößen zu trainieren. Das Verkehrssystem unterliegt hochkomplexen Zusammenhängen und Messdaten müssen im Gesamtkontext betrachtet werden. Um dieses Problem zu lösen, wird die Entwicklung einer erweiterbaren Repräsentation als Eingabe für eine Vielzahl von Lernalgorithmen einen Schwerpunkt dieser Arbeit bilden. Die grundlegende Idee ist dabei, dass ähnliche Randbedingungen auch ähnliche Auswirkungen auf die überwachten Daten haben.

Diese Arbeit verfolgt darüber hinaus das Ziel, intuitiv verständliche Vorhersagen für den Zustand des Verkehrssystems im Ganzen zu treffen. Deshalb wird von Regressionsmodellen für die einzelnen Verkehrsgrößen abgesehen und stattdessen die Klassifikation gewählt. Die Klassen lehnen sich hierbei an die im natürlichen Sprachgebrauch verwendeten Klassen von freiem und gestautem Verkehr an. Wie in Kapitel 3 gezeigt wird, verwendet diese Arbeit jedoch noch eine dritte Klasse, die den gestauten Verkehr weiter aufteilt.

An dieser Stelle soll eine allgemeine Definition der Lernaufgabe erfolgen. Eine schematische Darstellung dieser Lernaufgabe erfolgt in Abbildung 1.1.

**DEFINITION 1.1: Allgemeine Lernaufgabe für die Verkehrsvorhersage**

*Gegeben ist ein zeitlich geordneter Strom von Messdaten. Es wird ein Fenster verwaltet, das die letzten  $\Delta t_w$  Messungen bis zum aktuellen Zeitpunkt  $t_a$  enthält. Das Modell soll nun zum Zeitpunkt  $t_a$  eine Prognose der Verkehrssituation zum Zeitpunkt  $t_a + \Delta t_h$  für einen gegebenen Vorhersagehorizont  $\Delta t_h$  liefern.*

## 1 Einleitung

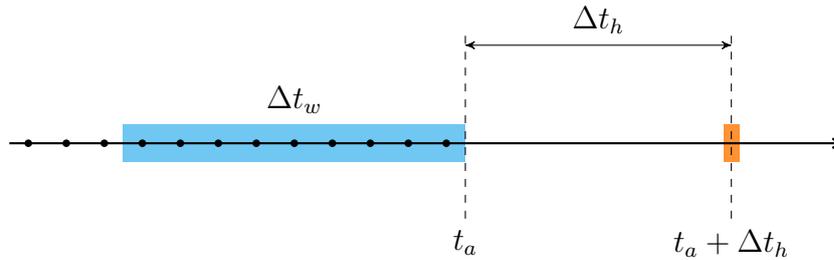


Abbildung 1.1: Schematische Darstellung der Lernaufgabe: Zum aktuellen Zeitpunkt  $t_a$  werden die Messdaten der letzten  $\Delta t_w$  Zeitpunkte verwendet, um eine Vorhersage für den Status des Verkehrssystems zum Zeitpunkt  $t_a + \Delta t_h$  (mit festem Vorhersagehorizont  $\Delta t_h$ ) zu treffen.

Um diese Lernaufgabe zu lösen, wird eine Repräsentation mithilfe von *Motifs* (siehe Kapitel 4) gesucht, welche die Ausgangsdaten geeignet in eine Eingabe für diverse Lernverfahren transformieren. Eine weitere Anforderung an die Repräsentation ist die Möglichkeit, eine beliebige Anzahl von Verkehrsgrößen miteinander verknüpfen zu können, um ein gemeinsames Modell zu ermöglichen.

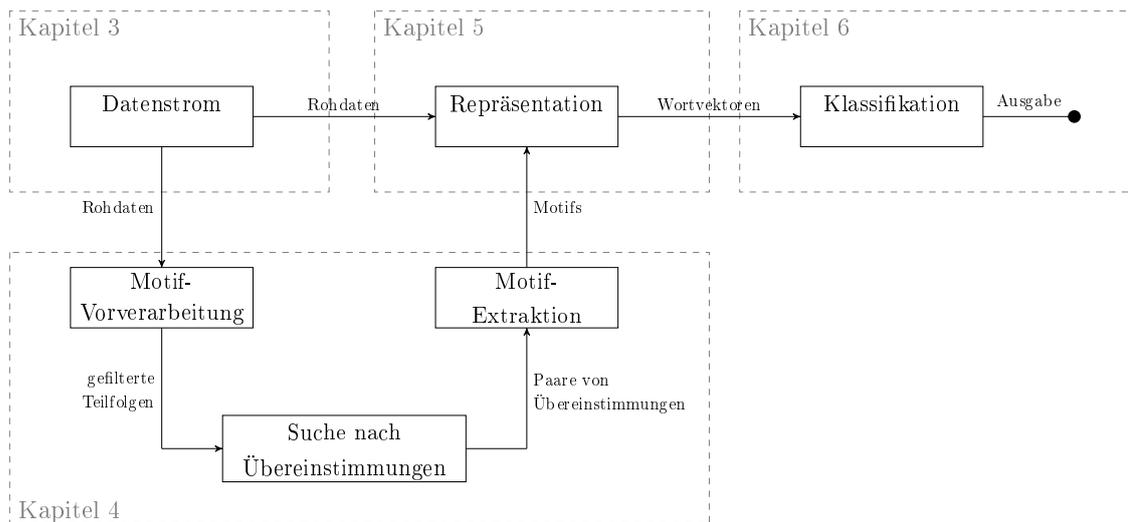


Abbildung 1.2: Schematische Darstellung des Vorhersagemodells. Die Gliederung der weiteren Arbeit folgt im Wesentlichen diesem Schaubild.

Die weitere Arbeit ist wie folgt gegliedert: Zunächst werden in Kapitel 2 wichtige Grundlagen aus den Bereichen Mathematik, Informatik und Verkehrsdynamik erläutert, um die weiteren Kapitel aus Abbildung 1.2 einzuleiten. In Kapitel 3 wird die verwendete Datenbasis vorgestellt. Aus diesem Strom von Messdaten werden in Kapitel 4 Motifs gewonnen, die in Kapitel 5 als Grundlage für eine Wortvektorrepräsentation der Ausgangsdaten verwendet werden. Eine Evaluation folgt im Anschluss in Kapitel 6. Es wird eine Reihe von Lernalgorithmen vorgestellt und auf ihre Eignung zur Vorhersage von Zustandsübergängen in Verkehrssystemen hin untersucht.

Die hier vorgestellte Repräsentation von Verkehrsdaten wird nach unterschiedlichen Aspekten evaluiert, bevor ein Vergleich mit einem „Expertensystem“ die Reihe von Experimente abrundet. Die Arbeit wird abschließend in Kapitel 7 noch ein Mal zusammengefasst und es werden mögliche, zukünftig zu untersuchende Aspekte diskutiert.



# Grundlagen

Dieses Kapitel widmet sich einigen Grundlagen aus den Bereichen Mathematik, Informatik und Verkehrsdynamik. An dieser Stelle werden kurz Begriffe und Verfahren vorgestellt, die für den weiteren Teil dieser Arbeit nützlich sind.

## 2.1 Mathematik

Die vorliegende Arbeit bedient sich einiger mathematischer Werkzeuge. Sie entstammen insbesondere der Statistik.

### 2.1.1 Zeitreihen

In Kapitel 3 werden die Ausgangsdaten dieser Diplomarbeit vorgestellt. Es handelt sich hierbei um *Zeitreihen* diverser Verkehrsgrößen. Mit dem Begriff „Zeitreihe“ werden geordnete Folgen von (Mess-)Werten bezeichnet. Gegenüber anderen Messdaten besitzen Daten aus Zeitreihen immer zwei Dimensionen. Die erste evtl. implizite Dimension ist die *Indextdimension*. Die Indexwerte der Reihe sind monoton steigende Elemente aus dem Bereich der Zeit. Die zweite Dimension ist meist ein Vektor und trägt den Namen *Wertedimension*. Ein solcher Vektor kann numerische, nominale oder auch beide Arten von Attributen enthalten.

#### DEFINITION 2.1: **Zeitreihe**

Eine **Zeitreihe** ist eine mathematische Folge von Elementen, die in ihrer zeitlichen Indextdimension streng monoton steigt. Die Werte der Indextdimension sowie mögliche nominale Werte können durch eine geeignete Abbildung auf die Menge der natürlichen Zahlen abgebildet werden. So ist eine Zeitreihe o. B. d. A. definiert durch:

$$x : \mathbb{N} \rightarrow \mathbb{N}^l \times \mathbb{R}^m \times \mathbb{C}^n.$$

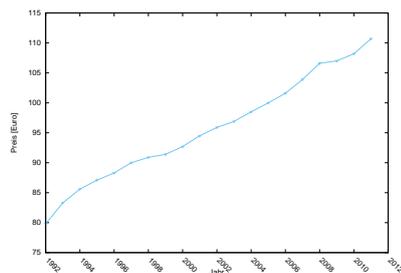
In Abbildung 2.1 sind Zeitreihen von drei realen Prozessen dargestellt. Auch wenn sich die Verläufe von Zeitreihen stark unterscheiden können, setzt sich eine Zeitreihe nach dem *klassischem Komponentenmodell* immer aus vier Komponenten zusammen [85].

<sup>1</sup>Der *GENESIS*-Datenbank des *statistischen Bundesamts Deutschland* entnommen:

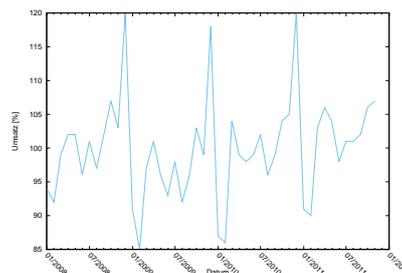
<https://www-genesis.destatis.de/genesis/online>

<sup>2</sup>Daten vom *Solar Influences Data Analysis Center*: <http://sidc.oma.be/sunspot-data/>

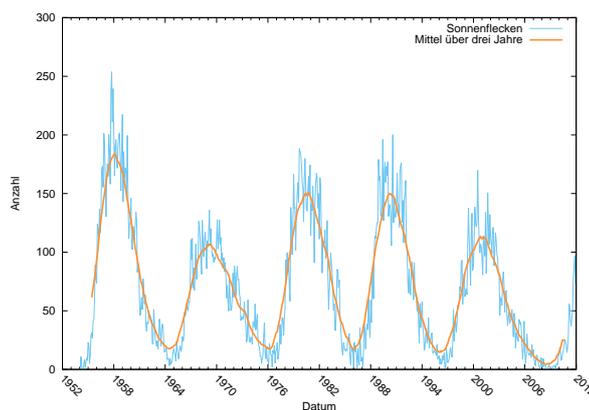
## 2 Grundlagen



(a) Verbraucherpreisindex



(b) Umsatz Einzelhandel



(c) Anzahl Sonnenflecken

Abbildung 2.1: Beispiele für Zeitreihen: Abbildung (a) zeigt den Verlauf des Verbraucherpreisindex für die Bundesrepublik Deutschland<sup>1</sup> zwischen 1992 und 2011. In Abbildung (b) ist der monatliche Umsatz des Einzelhandels in Deutschland<sup>1</sup> für die Jahre 2008 bis 2011 zu sehen. In Abbildung (c) ist die Anzahl der beobachteten Sonnenflecken<sup>2</sup> für die Jahre 1952 bis 2011 dargestellt.

### Trend

Unterliegen die Werte einer Zeitreihe einem *Trend*, so ist ein monotoner An- bzw. Abstieg der Werte zu beobachten. Es handelt sich hier um die langfristige „Richtung“ in welche die Werte driften. Ein Trend kann von einem anderen Trend abgelöst werden, wenn bspw. der allgemeine Anstieg stoppt und ein Abstieg beginnt.

#### BEISPIEL 2.1: Verbraucherpreisindex der Bundesrepublik Deutschland

Um die Entwicklung der Inflation nachvollziehen zu können, berechnet das statistische Bundesamt Deutschland regelmäßig den Verbraucherpreisindex (VPI). Das statistische Bundesamt verwaltet einen „Warenkorb“, der für die aktuelle Gesellschaft

*typische Konsumgüter enthält. Hinzukommen allgemeine Lebenshaltungskosten wie Miete, Kraftstoffe usw. Mit dem Preis für diesen Warenkorb kann so die Teuerungsrate bestimmt werden. Wie in Abbildung 2.1(a) zu sehen ist, nimmt diese Teuerungsrate in den Jahren 1992 bis 2011 relativ konstant zu. Es kann bei dieser Zeitreihe also eindeutig ein Trend identifiziert werden.*

## Saison

Bei der *Saisonkomponente* handelt es sich um wiederkehrende Regelmäßigkeiten im Verlauf der Zeitreihe. Dies sind meist Randbedingungen, die von außen auf das System einwirken. Randbedingungen sind z.B. Feiertage, der Beginn der Sommerferien oder Sportereignisse, die regelmäßig zu hohem Verkehrsaufkommen führen. Die Saisonkomponente kann allerdings auch wörtlich interpretiert werden. Bei vielen Zeitreihen haben die Jahreszeiten, bedingt durch Gewohnheiten wie dem Konsum zu Weihnachten oder den Wetterbedingungen, Einfluss auf den Verlauf.

### BEISPIEL 2.2: Umsatz des deutschen Einzelhandels

*Die Zeitreihe in Abbildung 2.1(b) stellt die Umsatzzahlen des Einzelhandels in Deutschland für die Jahre 2008 bis 2011 dar. Auffallend sind die wiederkehrenden Spitzen in den Verkaufszahlen vor jedem Jahreswechsel im Dezember sowie der starke Einbruch für die Folgemonate Januar und Februar. Dieses Muster ist auf saisonale Einflüsse zurückzuführen, da Konsumenten durch das Weihnachtsfest gewöhnlich deutlich mehr Geld umsetzen und ihren Konsum in den Folgemonaten zügeln.*

## Zyklen

Die Werte vieler Systeme haben einen langjährigen, alternierenden Verlauf. Diese Komponente wird *Zyklus* oder auch *Konjunktur* genannt. Es sind wiederkehrende Muster zu erkennen, die jedoch eine variable Periode und Amplitude besitzen können.

### BEISPIEL 2.3: Sonnenflecken (1)

*Eine der längsten kontinuierlich erfassten Zeitreihen ist die Erhebung der Wolf'schen Relativzahl. Seit über 150 Jahren wird die Zahl der monatlich beobachteten Sonnenflecken erfasst. In Abbildung 2.1(c) ist ein Ausschnitt dieser Zeitreihe zu sehen. Für die Anzahl der Sonnenflecken wurde ein ungefähr elf Jahre langer Zyklus entdeckt [42]. Es handelt sich hierbei um einen Zyklus und keine saisonale Einflüsse, da die*

*Ursache nicht auf ein wiederkehrendes Ereignis zurückzuführen ist, sondern dem System selbst innewohnt. Außerdem variiert sowohl die Periode des Zyklus als auch die Amplitude der Ausschläge der Kurve teilweise sehr stark.*

### Rauschen

Wenn eine Zeitreihe von Trend, Saison und Zyklen befreit wurde, bleibt nur noch eine weitere Komponente über: das *Rauschen*, auch *Restkomponente* genannt. In der Regel ist diese Komponente in allen Zeitreihen zu realen Abläufen zu finden. Es handelt sich dabei um zufällige Abweichungen mit meist niedrigen Amplituden.

#### BEISPIEL 2.4: Sonnenflecken (2)

*In Abbildung 2.1(c) ist neben dem exakten Verlauf der monatlich beobachteten Sonnenflecken noch das Mittel der Werte über drei Jahre eingetragen. Es ist leicht zu sehen, dass die Zahl von Sonnenflecken im Mittel einen an eine Sinuskurve erinnernden Verlauf hat. Die Abweichungen zwischen dem Mittelwert und der exakten Zahl von Sonnenflecken ist also auf ein Rauschen zurückzuführen.*

### 2.1.2 Quantile

Bei Quantilen handelt es sich um Rangstatistiken einer Verteilung [30]. Quantile werden aber auch häufig als Schätzer für Stichproben verwendet. In dieser Arbeit werden Quantile im Wesentlichen verwendet, um Schwellwerte in Abhängigkeit der unterliegenden Verteilung zu bestimmen.

#### DEFINITION 2.2: $\phi$ -Quantil

*Gegeben sei eine Wertemenge  $\mathbf{W}$ . Sind die  $n$  Elemente von  $\mathbf{W}$  aufsteigend sortiert, so ist das  $\phi$ -Quantil das Element  $w_i$ , welches die ersten  $(\phi \cdot 100)\%$  von allen anderen Werten der Verteilung trennt. Links des  $\phi$ -Quantils liegen dabei ausschließlich kleinere Werte. Für diskrete Mengen wird der Index  $i$  ausgewählt für den gilt:*

$$(i - 1) < \phi \cdot n \leq i \wedge (i + 1) > \phi \cdot n.$$

*Für das Element  $w_i$  gilt dann:*

$$w_j < w_i \quad , \forall j < i$$

*und*

$$w_k \geq w_i \quad , \forall k > i.$$

Zur exakten Bestimmung von  $\phi$ -Quantilen einer Beispielmenge werden die  $n$  Beispiele in aufsteigender Reihenfolge in einer Liste angeordnet. Das  $\phi$ -Quantil ist dann das Element der Liste mit Position  $\lceil \phi \cdot n \rceil$ . Natürlich ist es für große Datenmengen auch möglich approximative Rangstatistiken zu verwenden (siehe z. B. [4, 38]).

Eine besondere Bedeutung haben das 0.25-, das 0.75- sowie das 0.5-Quantil. Diese Elemente werden auch *unteres* (0.25) und *oberes Quartil* (0.75) bzw. *Median* (0.5) genannt. Bei Betrachtung des Medians (siehe Beispiel 2.1.2) zeigt sich ein Vorteil von Quantilen gegenüber anderen Rangstatistiken: Quantile sind robust gegenüber Ausreißern, d. h. wenige übermäßig große wie kleine Werte haben nur geringen Einfluss auf ein Quantil.

#### BEISPIEL 2.5: Bestimmung des 0,5-Quantils (Median)

Gegeben sei eine Wertereihe  $\mathbf{W}$  mit zehn Werten:

$$[0; 1; 2; 2; 2; 3; 3; 5; 6; 1042].$$

Der Median ist das  $\phi$ -Quantil für  $\phi = 0,5$  und berechnet sich durch

$$z(0,5) = w_{\lceil 0,5 \cdot 10 \rceil} = w_5 = 2.$$

Der Median ist damit deutlich robuster gegenüber dem übermäßig hohen Wert 1042 als es der Mittelwert für diesen Fall ist:

$$\bar{x}(\mathbf{W}) = \frac{\sum_{i=0}^{10} (w_i)}{10} = 106,6.$$

### 2.1.3 Euklidischer Abstand

Für die Vorhersage des Verkehrszustands werden im späteren Verlauf Vektoren aus den Daten gewonnen. Zum Vergleich zweier Vektoren werden *Metriken* verwendet. Besonders häufig wird der *euklidische Abstand* eingesetzt. Der Abstand zweier Vektoren ist nach dieser Metrik durch die Summe der Quadrate der Differenzen der einzelnen Dimensionen bestimmt:

$$dist_{euklid}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \quad (2.1)$$

Auch wenn es eine Vielzahl weiterer Metriken gibt, wird zur Bestimmung des Abstands bzw. der Ähnlichkeit von Beispielen in dieser Arbeit ausschließlich der euklidische Abstand verwendet.

### 2.1.4 Standardisierung von Zufallsvariablen

Durch eine *Standardisierung* (oder auch *z-Transformation*) werden die Werte einer Beispielmenge so transformiert, dass der Erwartungswert der Menge null und ihre Standardabweichung eins beträgt. Für die zu standardisierende Beispielmenge wird sowohl das arithmetische Mittel  $\bar{x}$  als auch die empirische Standardabweichung  $s_x$  benötigt. Die Standardisierung erfolgt dann mithilfe von (2.2) [10]:

$$x'_i = \frac{x_i - \bar{x}}{s_x}. \quad (2.2)$$

Werden Beispielmengen standardisiert betrachtet, ist es möglich die Mengen zu vergleichen auch wenn sie unterschiedlichen Wertebereichen entstammen. Außerdem kann beim Clustering von Attributen unterschiedlicher Wertebereichen der Einfluss der Attribute auf das Ergebnis ausgeglichen werden. In Kapitel 6 werden Gruppen von Zählschleifen identifiziert und es werden relative Anteile aus dem Bereich  $[0; 1]$  zusammen mit der Höchstgeschwindigkeit verwendet. Werte sind hier üblicher deutlich höher und sind fast ausschließlich für das Ergebnis des Clusterings ausschlaggebend, wenn die Werte nicht standardisiert werden.

### 2.1.5 Stichprobenziehung

Bei vielen realen Prozessen ist die Grundgesamtheit aller Beispiele weder erfassbar noch bekannt. Um trotzdem statistische Informationen zu erlangen, wird eine *Stichprobe* verwendet. Es handelt sich hierbei um eine Untermenge der Grundgesamtheit. Eine Stichprobe kann auf unterschiedliche Arten *gezogen* werden und die Auswahl des Verfahrens für eine solche Ziehung wirkt sich auf die Güte aller darauf aufbauenden Analysen und Experimente aus. Dementsprechend wurden viele unterschiedliche Stichprobenverfahren entwickelt. Eine Übersicht der gängigsten Verfahren findet sich z. B. in [90].

Für die Experimente in Kapitel 6 wird die *Stratifikation* verwendet, um Stichproben zu ziehen. Das Verfahren teilt die Beispielmenge in *Schichten* auf. Die Summe der Beispiele aller Schichten ergibt genau die Grundgesamtheit, d. h. die Aufteilung erfolgt disjunkt [19]. Im hier behandelten Fall entsprechen die Schichten den Klassen der Beispielmenge. Durch Stratifikation können nun Stichproben beliebiger Größe aus der Beispielmenge gezogen werden.

Stratifizierte Beispielmengen haben die selbe Schichtenstruktur wie die Grundgesamtheit, d. h. der Anteil an Beispielen der Klasse  $i$  ist in der Stichprobe annähernd identisch zum Anteil von Beispielen dieser Klasse bzgl. aller Beispiele in der Grundgesamtheit [90].

Stratifikation wird in dieser Arbeit verwendet, um die Beispielmenge in Trainings- und Testmenge aufzuteilen. Diese Aufteilung soll die Anteile der einzelnen Klassen konservieren, da ein sehr großes Ungleichgewicht in der Schichtenstruktur herrscht (vgl. Kapitel 6). Eine rein zufällige Aufteilung birgt die Gefahr, dass nicht alle Klassen

in der Trainings- bzw. Testmenge enthalten sind und somit die Klassifikationsgüte sinkt [102].

## 2.2 Informatik

In diesem Abschnitt werden insbesondere Terminologien und Verfahren des maschinellen Lernens besprochen.

### 2.2.1 Datenstrom

Daten können in Bezug auf ihren Umfang in zwei Kategorien aufgeteilt werden:

- endliche *Datenmengen*,
- ungebundene *Datenströme*.

Bei erst genannten handelt es sich um klassische Datenquellen wie Textdateien oder relationale Datenbanken. Einige grundlegende Eigenschaften zur Charakterisierung von Datenmengen sind:

- das Volumen der Daten nimmt nicht oder nur in geringem Umfang zu,
- zeitliche Informationen zu den einzelnen Daten können vorhanden sein, müssen es jedoch nicht,
- ein wahlfreier Zugriff auf die einzelnen Element der Datenmenge ist technisch möglich.

Mit einem Datenstrom hingegen wird eine sequentielle Folge von Ereignissen bezeichnet. Die Quellen für solche Ereignisse können hierbei unterschiedlichster Form sein: von Aktienkursen über Webserver-Logs bis hin zu Sensornetzen. Aufgrund der zunehmenden Verbreitung von Sensornetzen wird der Verarbeitung von Datenströmen ein immer größerer Stellenwert beigemessen [6].

Der Entwurf von Algorithmen zur Verarbeitung von Datenströmen ist jedoch komplexer als es für Datenmengen der Fall ist. Dies ist mit den Eigenschaften von Datenströmen zu begründen. Sie umfassen u. a.:

- die einzelnen Elemente eines Datenstroms können i. d. R. nur ein einziges Mal betrachtet werden,
- ein wahlfreier Zugriff ist ausgeschlossen,
- durch den ungebundenen Charakter ist das Verlangsamen von Datenströmen auf Dauer nicht möglich,

- jedes Ereignis trägt eine implizite zeitliche Information, da die Ereignisse in einem Datenstrom zeitlich geordnet auftreten.

Algorithmen die zur Verarbeitung eines Datenstroms geeignet sind werden als *Online-Algorithmen* bezeichnet. Solche Algorithmen sind zumeist probabilistisch oder heuristisch. Mit der Einführung von Fehlerschranken für das Ergebnis wird es ermöglicht, Datenströme ohne Speicherung oder Verlangsamung zu verarbeiten [34].

### 2.2.2 Fensterung

Viele Online-Algorithmen verwenden eine *Fensterung*, um den ungebundenen Datenstrom zu handhaben. Hierbei wird ein Fenster mit fester *Breite*  $w$  über den Strom von Beispielen geschoben, um eine sich ständig ändernde Teilfolge des Datenstroms mit konstanter Länge betrachten zu können [80]. Die *Schrittweite* der Fensterung beträgt  $s$  Elemente, d. h. bei jedem Verschieben des Fensters werden  $s$  neue Elemente aufgenommen, wofür die ältesten  $s$  Elemente entfernt werden. Die zeitliche Ordnung des Datenstroms wird hierbei konserviert, d. h. das älteste Element des Fensters befindet sich immer an Position  $0$ , das aktuellste Element an Position  $w - 1$ .

#### DEFINITION 2.3: Fensterung

Gegeben ein Datenstrom  $\mathbf{X}$ , der o. B. d. A. ein Element pro Zeitschritt  $\Delta t$  erzeugt. Eine **Fensterung**  $F$  mit Breite  $w$  und Schrittweite  $s$  wird zu jedem Zeitpunkt  $t_i$  aktualisiert, wenn gilt:

$$t_i \bmod s = 0.$$

Das Fenster  $F$  besteht für das Zeitintervall  $[t_i, t_{i+(s-1)}]$  aus den Elementen:

$$F = \langle x_{i-(s-1)}, x_{i+1-(s-1)}, \dots, x_i \rangle.$$

Auf die Elemente eines Fensters können beliebige Funktionen, wie z. B. Durchschnitts- und Quantilberechnungen oder die Extraktion von Extremwerten angewendet werden [71]. In dieser Arbeit werden Fensterungen mit Schrittweite  $s = 1$  für drei Aufgaben verwendet:

- Reduzierung des Rauschen der Ausgangsdaten durch Mittelwertbildung,
- Erzeugung von Teilfolgen,
- Beschränkung auf die Elemente der letzten  $\Delta t_w$  Zeitintervalle.

### 2.2.3 Maschinelles Lernen

Das allgemeine Ziel des *maschinellen Lernens* ist es, in einer Datenmenge Wissen zu entdecken und für *Aussagen* über Datentupel in der Zukunft zu nutzen. Ein System kann als *lernend* bezeichnet werden, wenn „*es sein Verhalten so ändert, dass es in Zukunft eine bessere Leistung aufweist*“ [102], d. h. Aussagen nach dem Lernen eine höhere Qualität besitzen als im Vorfeld.

Im maschinellen Lernen werden hierfür automatisierte Verfahren verwendet, die in Datenmengen und -strömen nach Strukturen zur Verbesserung der Aussage suchen. Die verwendeten Daten werden als *Beispielmenge* bezeichnet und bestehen aus einzelnen *Beispielen*, die für jedes *Attribut* eine *Ausprägung* besitzen. Für ein konkretes Problem wird eine *Lernaufgabe* definiert und anschließend algorithmisch gelöst. Ziel des *Lernalgorithmus* ist es, ein *Modell* zur Vorhersage einer Ausgabe abhängig von der Merkmalsausprägung der Beispiele zu finden. Es wird also generell eine Funktion  $\hat{f}(\mathbf{x}_i)$  gesucht, die eine Vorhersage  $\hat{y}_i$  trifft:

$$\hat{f}(\mathbf{x}_i) = \hat{y}_i \quad (2.3)$$

Hierbei soll die Vorhersage  $\hat{y}_i$  möglichst gut die wahren Werte  $y_i$  approximieren.

Die Ausgabe  $\hat{y}$  kann sowohl quantitativer als auch qualitativer Natur sein. Dies gilt ebenfalls für die Merkmale der Eingabe. Abhängig von der Ausgabe ist die Lernaufgabe eine *Regression* (quantitative Ausgaben) oder eine *Klassifikation* (qualitative Ausgaben).

#### BEISPIEL 2.6: Regression

*In der Meteorologie werden Modelle verwendet, um das Wetter in den nächsten Stunden bis Tagen vorauszusagen. Häufige Ausgaben sind die Temperatur und die zu erwartende Regenmenge in einer bestimmten Region. Es handelt sich hierbei um Werte aus dem Bereich der reellen Zahlen. Somit liegt hier eine Regression vor.*

#### BEISPIEL 2.7: Klassifikation

*Medizinische Proben oder Aufnahmen dienen häufig dem Zweck eine bestimmte Krankheit erkennen oder ausschließen zu können. Lernalgorithmen in diesem Bereich sollen also eine binäre Ausgabe der Form „positiv“/„negativ“ erzeugen. Dies ist offensichtlich eine quantitative Ausgabe, wodurch hier eine Klassifikation vorliegt.*

Eine Lernaufgabe ist im Allgemeinen in die drei Phasen aus Abbildung 2.2 unterteilt. Zu Beginn wird in der *Trainingsphase* das Modell trainiert. Dafür wird ein Teil der verfügbaren Beispielmenge verwendet. Diese *Trainingsmenge* besteht idealerweise aus zufällig aus der unterliegenden Verteilung gezogenen Beispielen. Für diese

## 2 Grundlagen

Beispiele wird nun ein Modell gesucht, das „gut zu den Daten passt“. Für eine gegebene Modellklasse bedeutet dies, dass die *Parameter* des Modells optimiert werden müssen. Wird z. B. eine lineare Regression im  $\mathbb{R}^2$  durchgeführt, ist die Modellklasse durch die Funktion

$$\hat{f}(\mathbf{X}) = \beta_0 + \beta_1 \cdot \mathbf{X} \quad (2.4)$$

gegeben und das Training besteht daraus  $\beta_0$  und  $\beta_1$  zu optimieren.

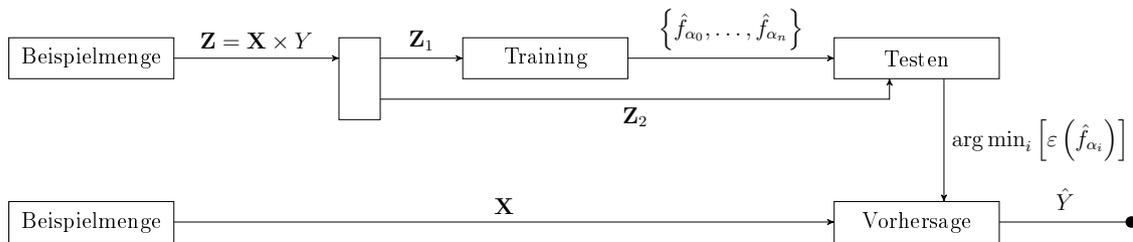


Abbildung 2.2: Die drei Phasen des maschinellen Lernens.

In der anschließenden *Testphase* wird das gelernte Modell *evaluiert*. Es wird ein Teil der Beispiele als *Testmenge* ausgewählt. Oft ist dies der Teil der Beispielmenge, der nicht in der Trainingsmenge enthalten ist. Unter der Annahme, dass die Beispiele beider Mengen sowohl unabhängig als auch identisch verteilt sind, wird in dieser Phase die *erwartete Vorhersagegüte* des Modells bestimmt.

Üblicherweise wird in der Trainingsphase eine Menge von Modellen  $\{\hat{f}_{\alpha_0}, \dots, \hat{f}_{\alpha_n}\}$  bestimmt und das für die Vorhersage der Ausgabe unabhängiger Beispiele am besten geeignete  $\hat{f}_{\alpha_i}$  in der Testphase ausgewählt [41].

Anschließend kann die Phase der eigentlichen Anwendung des Modells beginnen. Im Gegensatz zur Trainings- und Testmenge werden nun Vorhersagen  $\hat{y}_j$  für Beispiele  $\mathbf{x}_j$  ohne wahre Ausgabe  $y_j$  mit dem Modell  $\hat{f}_{\alpha_i}$  vorhergesagt.

Neben dem oben beschriebenen *überwachten Lernen*, bei dem die Ausgabe für die Trainings- und Evaluationsphase bekannt ist gibt es noch das *unüberwachte Lernen*. Hier sind keine wahren Ausgaben  $y$  bekannt. Häufig werden Verfahren des unüberwachten Lernens verwendet, um Strukturen in der Datenbasis zu finden. So können ähnliche Beispiele zu *Clustern* zusammengefasst werden wie z. B. in Abschnitt 6.1 zur Gruppierung von Sensoren.

### 2.2.4 Bestimmung der Vorhersagegüte eines Modells

Für die Bestimmung der Vorhersagegüte eines Modells wird der Fehler betrachtet, den ein Modell begeht. Hierfür wird eine Verlustfunktion  $L(y, \hat{f}(\mathbf{x}) = \hat{y})$  verwendet. Sie bestimmt die Abweichung zwischen den einzelnen Vorhersagen  $\hat{y}_i$  und den wahren Werten  $y_i$ . Der Vorhersagefehler eines Modells  $\hat{f}$  bzgl. einer Beispielmenge  $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$

mit  $n$  Beispielen ist somit

$$\begin{aligned}\varepsilon(\hat{f}(\mathbf{X})) &= \sum_{i=1}^n L(y_i, \hat{f}(\mathbf{x}_i)) \\ &= \sum_{i=1}^n L(y_i, \hat{y}_i).\end{aligned}\tag{2.5}$$

Es existieren viele Verlustfunktionen zur Bestimmung des Fehlers eines Modells. Die am häufigsten verwendeten Funktionen sind:

- *quadratischer Fehler*:  $L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$ ,
- *absoluter Fehler*:  $L(y_i, \hat{y}_i) = |y_i - \hat{y}_i|$ ,
- *0-1-Loss*:  $L(y_i, \hat{y}_i) = \begin{cases} 0, & \text{falls } y_i = \hat{y}_i \\ 1, & \text{sonst.} \end{cases}$

Wenn für ein Klassifikationsproblem keine Kosten für die fehlerhafte Klassifikation der einzelnen Klassen gegeben sind, wird meist der 0-1-Loss verwendet. Auch für diese Arbeit ist dies der Fall.

Die Beispielmenge wird in zwei Teile aufgeteilt – der Trainings- ( $Z_1$ ) und Testmenge ( $Z_2$ ). Dementsprechend kann zwischen dem *Trainingsfehler* für falsche Aussagen  $\hat{y}_i$  aus der Trainingsmenge und dem *Testfehler* für falsche Aussagen  $\hat{y}_j$  aus der Testmenge unterschieden werden. Ersterer kann prinzipiell durch Erhöhung der Komplexität des Modells an null angenähert werden. Eine zu hohe Komplexität ist jedoch aus zwei Gründen zu vermeiden [41]:

- Es tritt der „*Fluch der hohen Dimension*“ auf. So ist ein Raum mit höherer Dimension spärlicher besetzt und es werden mehr Beispiele benötigt, wodurch die Laufzeit steigt.
- Es tritt das Phänomen der *Überanpassung* an die Daten der Trainingsmenge auf, wodurch die Vorhersagegüte für *unabhängige* Beispiele sinkt.

Wenn ein Modell überangepasst ist, *generalisiert* es schlecht. Da in der Praxis aber insbesondere die Vorhersage für in der Trainingsphase nicht betrachtete Beispiele von großem Interesse ist, muss die Modellkomplexität gefunden werden, für die der zu erwartende Vorhersagefehler  $\varepsilon(\hat{f}(\mathbf{X}))$  minimal ist. Hierzu wird der Fehler der Testmenge  $Z_2$  als Schätzer verwendet [52]:

$$\varepsilon(\hat{f}_{Z_2}(\mathbf{X})) \sim \varepsilon(\hat{f}(\mathbf{X})).\tag{2.6}$$

## Konfusionsmatrix

Die nachfolgenden Kapitel beschäftigen sich mit der Klassifikation zur Vorhersage von Verkehrszuständen. An dieser Stelle wird deshalb kurz ein nützliches Werkzeug zur Bestimmung und Darstellung der Güte eines Klassifikationsmodells vorgestellt. Es handelt sich hierbei um die *Konfusionsmatrix*. In solch einer Matrix werden die Vorhersagen  $\hat{y}$  aller Klassen den wahren Klassen  $y$  gegenüber gestellt. Hierdurch ist die Bestimmung der *Sensitivität* (oder auch *Recall*) sowie der *Genauigkeit* (auch *Precision* genannt) für die einzelnen Klassen möglich [41].

Die Sensitivität bezeichnet die Anzahl der korrekt klassifizierten Beispiele in Relation zu allen Beispielen, die tatsächlich dieser Klasse angehören. Bei der Genauigkeit handelt es sich um die Rate der korrekt klassifizierten Beispiele gegenüber aller dieser Klasse zugeordneten Beispielen. Zusätzlich zu der klassenübergreifenden *Klassifikationsgüte* lässt sich also mit der Konfusionsmatrix auf einfache Art und Weise eine Bewertung der Vorhersage bzgl. einzelner Klassen erstellen. In Tabelle 2.1 ist ein Beispiel für eine Konfusionsmatrix gegeben.

Tabelle 2.1: Konfusionsmatrix für die Klassifikation des Iris-Datensatz. Es wurde eine Support Vector Machine trainiert. Die klassenübergreifende Vorhersagegüte beträgt 91%.

		Y			Genauigkeit
		iris-setosa	iris-versicolor	iris-virginica	
$\hat{f}(\mathbf{X})$	iris-setosa	31	0	0	100%
	iris-versicolor	0	33	9	78,57%
	iris-virginica	0	0	27	100%
Sensitivität		100%	100%	75%	

### 2.2.5 Kreuzvalidierung

Eine *Kreuzvalidierung* kann eingesetzt werden, um den Testfehler eines Lernverfahren abzuschätzen. Besonders nützlich ist dieses Verfahren, wenn nicht ausreichend Beispiele für eine Aufteilung in eine Trainings- und Testmenge verfügbar sind. Die Beispielmenge wird für eine  $K$ -fache Kreuzvalidierung in  $K$  disjunkte Partitionen von ungefähr gleichem Umfang aufgeteilt. Mit jeweils  $(K - 1)$  Partitionen wird das Training durchgeführt, bevor die nicht betrachtete Partition zum Testen des Modells verwendet wird, um das Modell zu evaluieren. Dies wird  $K$  mal wiederholt. In jedem Durchlauf wird die zurückgehaltene Partition gewechselt, sodass am Ende jede der  $K$  Partitionen als Testmenge verwendet wurde [39]. Typische Werte für  $K$  sind fünf, zehn und Anzahl der Beispiele  $N$ . Letztere Variante wird auch *leave-one-out* Kreuzvalidierung genannt [41].

Der zu erwartende Fehler einer  $K$ -fachen Kreuzvalidierung mit einer Trainings-

menge von  $N$  Beispielen ist dann durch

$$\varepsilon\left(\hat{f}(\mathbf{x})\right) \hat{=} CV\left(\hat{f}(\mathbf{x})\right) = \frac{1}{N} \cdot \sum_{i=1}^N L\left(y_i, \hat{f}^{-i}(\mathbf{x}_i)\right) \quad (2.7)$$

gegeben. Dabei bezeichnet  $\hat{f}^{-i}(\mathbf{x})$  das Modell, welches ohne die Partition mit Beispiel  $\mathbf{x}_i$  erzeugt wurde [27]. In dieser Arbeit wird eine fünffache Kreuzvalidierung verwendet, um Modelle zu erzeugen und ihre Klassifikationsgüte abzuschätzen. Die Aufteilung in diese fünf Partitionen erfolgt hierbei stratifiziert.

### 2.2.6 Parameteroptimierung

Für die Minimierung des zu erwartenden Fehlers und somit für die Maximierung der Vorhersagegüte ist nicht nur die Auswahl der Modellklasse entscheidend. Jede Modellklasse besitzt eine, sich in Zahl und Art unterscheidende Menge von Parametern. Eine lineare Regression im  $\mathbb{R}^2$  ist z. B. durch

$$f(x) = \beta_0 + \beta_1 \cdot x \quad (2.8)$$

gegeben [41]. Eine Parameteroptimierung sucht für die Parameter  $\beta_0$  und  $\beta_1$  eine optimale Belegung mit Werten aus  $\mathbb{R}$ , sodass der Fehler bzgl. einer gegebenen Beispielmenge  $(\mathbf{X}, Y)$  minimal ist. In dieser Arbeit werden in Kapitel 6 zwei unterschiedliche Verfahren zur Parameteroptimierung eingesetzt, die im Folgendem beschrieben werden.

#### Rasteroptimierung

Bei einer *Rasteroptimierung* wird für jeden zu optimierenden Parameter eine Liste von möglichen Werten bzw. ein Intervall vorgeben [63]. Die Schrittweite innerhalb eines Intervalls muss hier kein konstanter Wert sein. Für  $k$  nächste Nachbarn in Kapitel 6 wird z. B. eine logarithmische Schrittweite verwendet.

Unabhängig von der Schrittweite innerhalb eines Intervalls oder der Definition einer Liste von Werten wird bei einer Rasteroptimierung für jede mögliche Kombination der Werte als Parametrisierung ein Modell erzeugt und die Klassifikationsgüte der einzelnen Modelle bestimmt. Bei einer großen Zahl von Parametern und/oder vielen möglichen Werten verlängert sich die Trainingsphase erheblich [33]. Deshalb wird dieses Optimierungsverfahren in Kapitel 6 nur dann verwendet, wenn die realistische Wertemenge für die Parameter eines Modells verhältnismäßig klein ist und nur aus ganzzahligen Werten besteht.

#### Optimierung mit evolutionären Algorithmen

Der Suchraum vieler Optimierungsprobleme ist zu groß, um in einer akzeptablen Zeit algorithmisch nach dem globalen Optimum durchsucht zu werden. In der Informatik wird zur Reduzierung von Laufzeiten oft Randomisierung eingesetzt. Dies hat jedoch

zur Folge, dass ein Erreichen des globalen Optimums nicht garantiert werden kann. Meist ist das Resultat jedoch „ausreichend“ gut und der Zeitgewinn wiegt den Verlust an Genauigkeit auf.

Für die Parameteroptimierung von Lernalgorithmen mit Werten aus  $\mathbb{R}$  werden in Kapitel 6 *evolutionäre Algorithmen* verwendet. Es handelt sich dabei um eine randomisierte Suche nach den optimalen Parametern für ein Modell, welche sich an der biologischen Evolution orientiert [99].

In der Biologie sorgt das Prinzip der *natürlichen Auslese* („*Survival of the Fittest*“) dafür, dass sich in einem bestimmten Lebensraum die am besten angepassten Spezies gegenüber anderen Arten durchsetzt. Die Anpassung an die Umwelt erfolgt im Wesentlichen durch *Mutationen* und *Kreuzungen* des Erbguts [22].

Auch evolutionäre Algorithmen bedienen sich der natürlichen Selektion. In jeder Iteration der Optimierung (*Generation*) wird eine Reihe von *Individuen* erzeugt. Jedes Individuum beschreibt eine mögliche Lösung des Optimierungsproblems. Für jedes dieser Individuen wird die *Fitness* bestimmt, um zu bewerten wie gut die Lösung für das behandelte Problem ist. Hierfür wird eine *Fitnessfunktion* verwendet [99]. In dieser Arbeit wird ein Klassifikationsproblem behandelt, wodurch sich die Klassifikationsgüte als Fitnessfunktion anbietet und im Folgenden verwendet wird.

Die Menge aller Individuen einer Generation ist die *Population*. Nach Bewertung der Fitness werden zufällige Mutationen für die einzelnen Individuen durchgeführt. Bei einer Mutation werden dabei nur wenige Parameter eines Individuums geringfügig geändert. Um die Population der nächsten Generation zu erzeugen, werden nun Kreuzungen aus je zwei Individuen erzeugt. Durch eine solche „Paarung“ erhält das resultierende Kind „Erbgut“ beider Eltern. Bei Kreuzungen haben Individuen mit einer großen Fitness eine höhere Wahrscheinlichkeit beachtet zu werden als Individuen mit einer schlechten Güte. Durch dieses Verfahren strebt die Lösung des Problems einem lokalen Optimum entgegen [99].

### 2.3 Verkehrsdynamik

Bei der *Verkehrsdynamik* handelt es sich um ein Teilgebiet der Physik. Untersuchungsgegenstand dieser Disziplin sind *Verkehrssysteme* bestehend aus der für den Verkehr verfügbaren Infrastruktur (z. B. Straßen) und den „Partikeln“, die den Verkehr bilden (z. B. PKW). Die Menge der Verkehrsteilnehmer ist dabei nicht auf motorisierte *Fahrer-Fahrzeug-Einheiten* beschränkt. Es existieren beispielsweise Arbeiten zum Verhalten von Fußgängern [31] oder auch Fahrradfahrern [28]. In diesem Abschnitt beschränken sich die erläuterten Grundlagen jedoch nur auf den Anwendungsfall des motorisierten Verkehrs auf Autobahnen.

Allgemeines Ziel der Verkehrsdynamik ist die Beschreibung eines Verkehrssystems durch ein Modell. Üblicherweise werden Modelle anhand ihres Detaillierungsgrades unterschieden. In hoch aufgelösten *mikroskopischen Modellen* werden einzelne Fahrer-Fahrzeug-Einheiten betrachtet. Jeder Verkehrsteilnehmer wird zu jedem Zeitpunkt

durch individuelle Größen wie Position, Geschwindigkeit, Beschleunigung, dem Verhalten des Fahrers oder den Fahrzeugeigenschaften beschrieben. Je nach Aufwand der Datenerhebung können auch *submikroskopische* Zustandsgrößen des Fahrzeugs wie den gesetzten Blinkern oder der Stellung des Gaspedals in solche Modelle einfließen [45].

Mikroskopische Modelle sind besonders gut geeignet, um Phänomene zu entschlüsseln, die auf die Interaktion einzelner Verkehrsteilnehmer zurückzuführen sind. So wurde z. B. mit dem *Nagel-Schreckenberg-Modell* [77] die Entstehung von Stau auf eine Reihe kollektiver Bremsmanöver zurückgeführt [60]. Darüber hinaus ist die mikroskopische Betrachtung von Verkehr notwendig für die Entwicklung von Fahrassistenzsystemen sowie der Simulation von Maßnahmen zur Verkehrsbeeinflussung [94].

Im Folgenden ist die Sicht auf das Verkehrssystem jedoch *makroskopisch*, da keines der oben genannten Szenarien in dieser Arbeit betrachtet wird und mikroskopische Modelle i. d. R. zu komplex für die Verkehrsvorhersage in Echtzeit sind. In makroskopischen Modellen ist die Betrachtung individueller Fahrer-Fahrzeug-Einheiten nicht erforderlich. Es liegt vielmehr die Annahme zugrunde, dass sich das Verhalten der einzelnen Verkehrsteilnehmer im Mittel dem gemeinsamen Verhalten anpasst, da die vorherrschenden Rahmenbedingungen (z. B. Wetterbedingungen, Sichtverhältnisse, Verkehrsdichte usw.) eine Anpassung aller Fahrer erfordern. Es wird also ein *aggregiertes* Verhalten der Summe aller Fahrer-Fahrzeug-Einheiten verwendet [45]. Makroskopische Modelle werden insbesondere verwendet, wenn

- das mikroskopische Verhalten einzelner Fahrer-Fahrzeug-Einheiten vernachlässigt werden kann,
- die Anwendung unter Echtzeitbedingungen erfolgt,
- die Eingangsdaten heterogen sind [94].

### 2.3.1 Makroskopische Verkehrsgrößen

Bei der makroskopischen Betrachtung von Verkehrssystemen finden Verfahren der Hydrodynamik Anwendung. Hierbei wird das Verkehrssystem als strömendes Fluid angesehen und durch drei Größen beschrieben [94].

Mit dem *Verkehrsfluss*  $Q(x, t)$  wird die Anzahl der Fahrzeuge, die den Querschnitt  $x$  während eines Zeitintervalls  $\Delta t$  passieren bezeichnet. Als Einheit wird für  $Q(x, t)$  meist „Fahrzeuge pro Stunde  $\left[\frac{\text{Fhz.}}{\text{h}}\right]$ “ gewählt.

Die *Verkehrsdichte*  $\rho(x, t)$  beschreibt hingegen die Anzahl von Fahrzeugen auf einem Streckenabschnitt  $\Delta x$  zum Zeitpunkt  $t$ . Üblicherweise wird diese Größe in „Fahrzeuge pro Kilometer  $\left[\frac{\text{Fhz.}}{\text{km}}\right]$ “ angegeben.

Als letzte Größe wird die *Geschwindigkeit*  $v(x, t)$  verwendet. Es handelt sich hierbei um die durchschnittliche Geschwindigkeit der Fahrzeuge an der Stelle  $x$  zum Zeitpunkt  $t$ . Die Geschwindigkeit wird gewöhnlich in „Kilometer pro Stunde  $\left[\frac{\text{km}}{\text{h}}\right]$ “ beschrieben.

Diese drei Größen sind durch die *Fluss-Dichte-Beziehung* (2.9) miteinander verknüpft.

$$Q(x, t) = \rho(x, t) \cdot v(x, t). \quad (2.9)$$

Es ist somit möglich eine der drei Größen exakt zu bestimmen, wenn die beiden anderen Größen bekannt sind.

In Abschnitt 3.4 wird gezeigt, dass Verkehrsfluss und Verkehrsdichte nicht direkt verfügbar sind. Um die Verkehrsdichte  $\rho(x, t)$  zu approximieren, wird eine weitere makroskopische Verkehrsgröße verwendet. Es handelt sich hierbei um die *Belegung*  $p(x, t)$ , die den relativen Anteil beschreibt, den ein Querschnitt  $x$  im Zeitraum  $\Delta t$  von Fahrzeugen belegt war. Diese Größe ist ohne Einheit und kann wie in Abschnitt 3.4 beschrieben wird zur Näherung der Dichte verwendet werden. Anschließend kann mit (2.9) der Verkehrsfluss bestimmt werden.

### 2.3.2 Verkehrsphasen

Ein Verkehrssystem durchläuft unterschiedliche Phasen. Eine sehr grobe, aber auch intuitive Einteilung bieten die Verkehrsphasen freier und gestauter Verkehr. Der freie Verkehr ist dabei durch eine niedrige Verkehrsdichte, zu vernachlässigenden Interaktionen zwischen Fahrzeugen sowie der im Rahmen gesetzlicher Vorgaben frei wählbaren Geschwindigkeit charakterisiert. Dem gegenüber herrschen im gestauten Verkehr erhöhte Verkehrsdichten, sodass sich die Verkehrsteilnehmer gegenseitig beeinflussen [59].

Der gestaute Verkehr ist dabei ein lokal zusammenhängendes Verkehrsmuster [60]. Ein solches Muster besitzt eine *stromab-* und eine *stromaufwärtige Staufront*, wie in Abbildung 2.3 dargestellt ist. Innerhalb des gestauten Verkehrs liegt eine Verkehrsdichte  $\rho_{Stau}$  sowie eine Geschwindigkeit  $v_{Stau}$  vor. Fahrzeuge, die an der stromabwärtigen Staufront den gestauten Bereich verlassen, beschleunigen auf  $v_1 > v_{Stau}$  in einen Bereich mit geringerer Dichte  $\rho_1$ . Umgekehrt geraten Fahrzeuge, welche die stromaufwärtige Staufront passieren, von einem Bereich mit niedriger Fahrzeugdichte  $\rho_0$  in den gestautem Bereich. Hieraus resultiert eine Entschleunigung auf die deutlich niedrigere Geschwindigkeit  $v_{Stau}$ .

Bei der Betrachtung des Verkehrsflusses an den Staufonten fällt auf, dass es drei mögliche Fälle gibt:

- $Q_{zu} < Q_{ab}$ : Mehr Fahrzeuge verlassen den gestauten Bereich als ihn erreichen. Wenn dieses Verhältnis bestehen bleibt, löst sich der Stau nach einer gewissen Zeit auf.
- $Q_{zu} > Q_{ab}$ : Es kommen mehr Fahrzeuge am gestauten Bereich an als ihn an der stromabwärtigen Staufont verlassen. Die räumliche Ausdehnung des gestauten Bereichs nimmt zu.
- $Q_{zu} = Q_{ab}$ : Es herrscht Gleichgewicht. Der gestaute Bereich wird weder zu noch abnehmen, da für jedes Fahrzeug, das den gestauten Bereich verlässt ein neues hinzukommt.

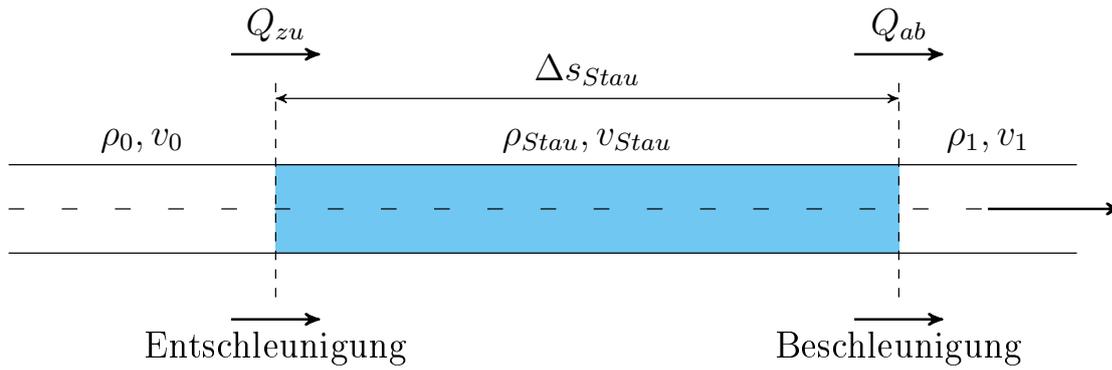


Abbildung 2.3: Ein gestauter Bereich besitzt immer eine stromauf- und eine stromabwärtige Staufront, die den Übergang in Phasen mit geringeren Verkehrsdichten und höheren Geschwindigkeiten markieren.

In der Praxis wird ein Verkehrssystem mit mehr als zwei Phasen beschrieben. Insbesondere die Phase des gestauten Verkehrs lässt sich weiter aufteilen. Das *Highway Capacity Manual (HCM)* [1] des amerikanischen *Transportation Research Board*<sup>3</sup> definiert sechs *Qualitätsstufen* des Verkehrsflusses (*level-of-service – LOS*). Vier der sechs Phasen befinden sich dabei im gestauten Verkehr. Die Abgrenzung der Phasen erfolgt durch Schwellwerte für die Größen Verkehrsdichte, Geschwindigkeit und Verkehrsfluss. Die Bestimmung der Verkehrsphase in Abschnitt 3.5 erfolgt in Anlehnung an die im HCM vorgestellten Qualitätsstufen. Nach der empirisch begründeten *Drei-Phasen-Verkehrstheorie* von Kerner [58] werden in dieser Arbeit jedoch nur drei Phasen unterschieden. Die Phase des gestauten Verkehrs wird hier in zwei Phasen aufgeteilt, sodass ein Verkehrssystem die Zustände

- *freier Fluss* (F),
- *synchroner Fluss* (S),
- *sich bewegender breiter Stau* (J)

besitzt. Wesentliche Unterscheidungsmerkmale sind auch hier die makroskopischen Verkehrsgrößen Verkehrsdichte  $\rho$ , die Geschwindigkeit  $v$  und der Verkehrsfluss  $Q$ . Umgangssprachliche Entsprechungen für die drei Phasen sind freier Verkehr (F), zäh fließender Verkehr (S) und Stau (J). Im Folgenden werden die grundlegenden Eigenschaften der drei Phasen definiert [60, 94].

#### DEFINITION 2.4: Freier Fluss(F)

Ein Autobahnabschnitt befindet sich im **freien Fluss**, wenn die Verkehrsdichte  $\rho$  niedrig ist und die Fahrzeugführer ihre Geschwindigkeit im Rahmen von gesetzlichen wie durch das Fahrzeug bestimmten Grenzen frei wählen können.

<sup>3</sup><http://www.trb.org/>

Im freien Fluss sind Interaktionen zwischen Fahrer-Fahrzeug-Einheiten im Allgemeinen zu vernachlässigen, da Fahrzeuge mit Geschwindigkeiten unter der eigenen Reisegeschwindigkeit überholt werden können.

### DEFINITION 2.5: **Synchroner Fluss (S)**

*Ein Autobahnabschnitt befindet sich im **synchronem Fluss**, wenn die Verkehrsdichte  $\rho$  erhöht ist und die Interaktion zwischen Fahrer-Fahrzeug-Einheiten nicht mehr vernachlässigt werden kann. Das Verkehrsmuster synchroner Fluss hat stromauf- und stromabwärtige Staufrenten, die durch Sprünge in der Verkehrsdichte  $\rho$  sowie der Geschwindigkeit  $v$  charakterisiert sind.*

Aufgrund der hohen Verkehrsdichte und dem weiterhin kontinuierlichen Strom von Fahrzeugen ist der Verkehrsfluss noch immer hoch und kann das Level, das im freien Fluss vorliegt sogar übersteigen. Die hohe Verkehrsdichte führt dazu, dass sich die individuellen Reisegeschwindigkeiten einer kollektiven Geschwindigkeit für das gesamte Verkehrsmuster angleichen. Die stromabwärtige Staufrent ist im synchronen Fluss meist ortsfest [59].

### DEFINITION 2.6: **Sich bewegender breiter Stau (J)**

*Ein Autobahnabschnitt befindet sich im **sich bewegendem breiten Stau**, wenn die Verkehrsdichte  $\rho$  stark erhöht und der Verkehrsfluss  $Q$  sehr niedrig ist. Die Geschwindigkeiten im Stau variieren stark und Verkehrsteilnehmer werden regelmäßig zum Anhalten gezwungen. Die stromabwärtige Staufrent propagiert sich entgegengesetzt zur Fahrtrichtung.*

Aufgrund des wellenförmigen Verlaufs der Geschwindigkeit und der stromaufwärtigen Fortpflanzung dieses Verkehrsmusters wird von einer *Schockwelle* gesprochen. Der nachfolgende Abschnitt erläutert, wie Zustandswechseln und Schockwellen auf Autobahnen entstehen.

### 2.3.3 Zustandsübergänge

Bei Zustandsübergängen in Verkehrssystemen handelt es sich um eine Art von „Zwischenphasen“, da ein Zustandswechsel niemals spontan erfolgt und immer einen räumlich-zeitlichen Verlauf hat [60]. Im Vergleich zu den Phasen selbst sind solche Phasenübergänge jedoch deutlich kürzer.

Eine Transition ist dadurch charakterisiert, dass die Werte der Verkehrsgrößen deutlich stärker als innerhalb einer Phase schwanken [107] und sich fern eines Gleichgewichtszustands befinden [105]. Es existieren wiederkehrende Strukturen, die in Übergängen einen Bruch in der fundamentalen Fluss-Dichte-Beziehung (2.9) beschreiben [50]. Die Hauptintention dieser Arbeit ist es, Bereiche in den Zeitreihen zu finden, die im Ungleichgewicht sind und somit Muster für Transitionen zu gewinnen.

### Ursachen für Zustandsübergänge

Ein Stau entsteht im freien Fluss nicht völlig ohne Ursache. Es müssen vielmehr gewisse Rahmenbedingungen vorliegen. Durch empirische Studien von Verkehrssystemen wurden drei Faktoren identifiziert, die maßgeblich für die Entstehung von Stau verantwortlich sind [94].

Besonders wichtig für die Stauentstehung ist ein *hohes Verkehrsaufkommen*. Nur durch die resultierende erhöhte Verkehrsdichte beeinflussen sich die Verkehrsteilnehmer ausreichend, um „Wechselwirkungsketten“ auszulösen. Gründe für ein hohes Verkehrsaufkommen sind z. B. die Stoßzeiten des Berufsverkehrs.

Ein Stau resultiert in einer Phase mit hohem Verkehrsaufkommen dabei immer aus einer *lokalen Störung*, die sich stromaufwärts fortpflanzt. Lokale Störungen treten im täglichen Verkehr regelmässig auf. Es handelt sich hierbei u. a. um Unachtsamkeiten, plötzliche Spurwechsel oder sich überholende LKW. Solche Ereignisse können in Daten mit makroskopischer Auflösung nicht entdeckt werden. Die lokale Störung verursacht jedoch eine *Kolonneninstabilität* in Form von Dichte- und Geschwindigkeitsschwankungen und verursacht somit makroskopisch messbare Sequenzen [60].

Neben diesen unbedingt notwendigen Randbedingungen erhöhen *Engstellen* die Wahrscheinlichkeit für einen Stau. An einer Engstelle verringert sich die Kapazität des Verkehrssystems woraus oft ein Anstieg der Verkehrsdichte resultiert. Es wird zwischen *permanenten* (Zufahrten, Steigungen, Kurven), *temporären* (Unfälle, Baustellen) und *verhaltensinduzierten Engstellen* (Staus/Unfälle auf der Gegenfahrbahn) unterschieden [94].

### Verkehrszusammenbruch

Der Zustandsübergang zwischen freiem und synchronem Fluss ist für die Verkehrsprognose besonders wichtig. Er wird als *Verkehrszusammenbruch* bezeichnet, da sich das Verhältnis von Verkehrsfluss, Geschwindigkeit und Verkehrsdichte in (2.9) grundlegend ändert [50]. Während dieser  $F \rightarrow S$ -Transition synchronisieren sich die individuellen Geschwindigkeiten und es stellt sich eine kollektive Geschwindigkeit für jede Spur ein. Der Grund hierfür ist die steigende Verkehrsdichte und die fehlenden Überholmöglichkeiten. Mit dem Erreichen des synchronen Flusses steigt die Wahrscheinlichkeit für eine  $S \rightarrow J$ -Transition. Für die Staubildung ist ein Übergang vom freien in den synchronen Fluss zwingend erforderlich [60].

### Entstehung von Schockwellen

Die Transition vom synchronen Fluss zum Stau geht mit der Entstehung von *Schockwellen* einher. Eine lokale Störung im synchronen Fluss löst eine sich gegen die Fahrtrichtung propagierende Schockwelle aus.

Das Phänomen einer Schockwelle im synchronen Fluss wird anhand Abbildung 2.4 näher erläutert. Der Fahrer des ersten Fahrzeugs erreicht mit Geschwindigkeit

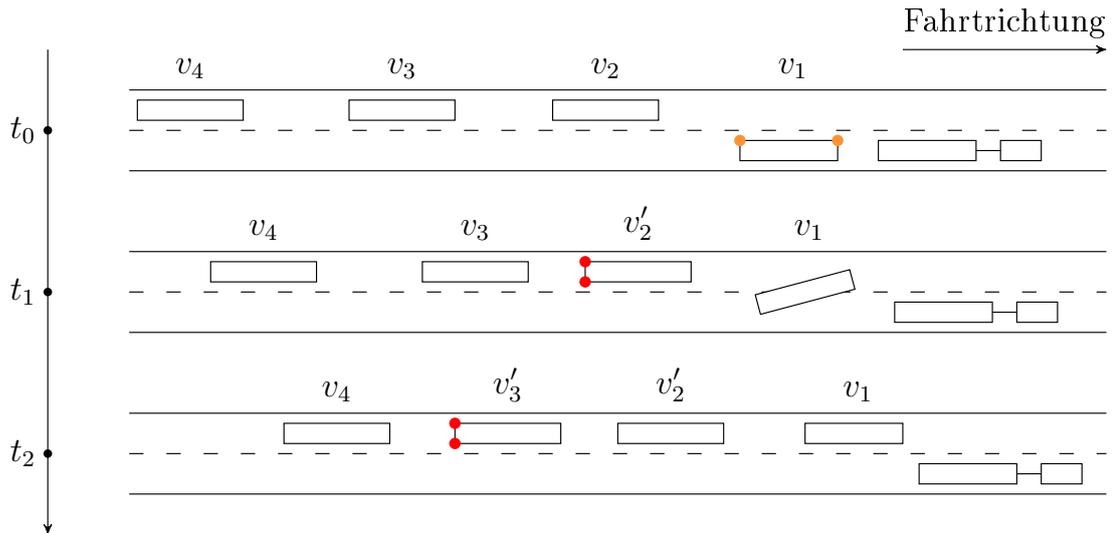


Abbildung 2.4: Entstehung einer Schockwelle im synchronen Fluss.

$v_1$  einen langsamer fahrenden LKW und beginnt mit einem Überholmanöver. Durch das Ausscheren auf die linke Spur wird der Fahrer von Fahrzeug 2 zum Bremsen gezwungen. Der Abstand zwischen Fahrzeug 1 und 2 sinkt aufgrund der menschlichen Reaktionszeit und der beschränkten Bremsfähigkeit von Fahrzeugen. Um den Sicherheitsabstand wiederherzustellen muss also  $v'_2$  kleiner als  $v_1$  sein.

Durch das Bremsmanöver von Fahrzeug 2 wird nun auch der Fahrer von Fahrzeug 3 gezwungen seine Geschwindigkeit  $v_3$  auf  $v'_3$  zu verringern. Aus den selben Gründen wie oben gilt auch hier  $v'_3 < v'_2$ . Diese Schockwelle setzt sich gegen die Fahrtrichtung weiter fort. Wenn die Verkehrsdichte abnimmt und die Kette von Bremsvorgängen abreißt kann die Schockwelle sich auflösen. Wenn die Verkehrsdichte weiterhin erhöht bleibt wird die Phase des Staus erreicht, da irgendwann  $v'_i = 0$  gilt und sich *Stop-and-Go-Verkehr* einstellt.

Empirische Studien belegen, dass Schockwellen im realen Verkehr immer eine Wellenlänge von mindestens einem Kilometer haben und sich mit einer konstanten Geschwindigkeit von ca.  $15 \frac{km}{h}$  gegen die Fahrtrichtung ausbreiten [58]. Die dabei auftretenden Verkehrsmuster von Be- und Entschleunigung sind über mehrere Minuten auf einem Messquerschnitt festzustellen [76].

# Datenbasis

Für den weiteren Verlauf der Arbeit werden Datensätze vom *Lehrstuhl für Physik von Transport und Verkehr*<sup>1</sup> der Universität Duisburg-Essen verwendet. Es handelt sich hierbei um Messwerte von Induktionsschleifen, die in die Fahrbahndecke der Autobahnen im Ruhrgebiet eingelassen sind. Eine Induktionsschleife ist ein ortsfester elektrischer Schwingkreis. Solange die Induktionsschleife unbelegt ist, d.h. sich kein metallischer Gegenstand unmittelbar über dem Schwingkreis befindet, liegt eine konstante Spannung vor. Sobald ein Fahrzeug den Bereich über dem Schwingkreis belegt fällt diese Spannung ab. Dieser Spannungsabfall endet sobald das Fahrzeug den Bereich über der Induktionsschleife verlässt. Eine solche Sensorik erlaubt das Zählen von Fahrzeugen und die Bestimmung der Dauer, die ein Fahrzeug eine Induktionsschleife belegt [94].

Um auch die Geschwindigkeit sowie die Länge von Fahrzeugen erfassen zu können, werden zwei Induktionsschleifen verwendet, die mit kurzem Abstand aufeinander folgen. Durch den kurzen zeitlichen Versatz zwischen dem Spannungsabfall der ersten und der zweiten Induktionsschleife können diese Größen für ein solches Induktionsschleifenpaar (kurz: *Zählschleife*) bestimmt werden. Die Länge von Fahrzeugen wird hierbei bestimmt, um eine detailliertere Datenerhebung durchführen zu können. Durch Festlegung eines Schwellwerts für die Länge von Fahrzeugtypen – im vorliegenden Fall PKW/LKW – können gemessene Größen für diese Gruppen von Fahrzeugen weiter aufgeschlüsselt werden [94]. Der vorliegende Datensatz umfasst Daten für das zweite Halbjahr 2010 (01.07.2010 – 31.12.2010).

## 3.1 Das Sensornetzwerk

Die Datenerhebung findet an 323 Messquerschnitten auf den Autobahnen des Ruhrgebiets statt. Abhängig von der Anzahl an Spuren am jeweiligen Messquerschnitt werden bis zu drei Zählschleifen verwendet, sodass insgesamt 765 Sensoren in die Fahrbahndecke eingelassen sind.

In Abbildung 3.1 ist die räumliche Aufteilung des Sensornetzwerks abgebildet. Es fällt auf, dass die Dichte an Sensoren regional stark divergiert. Besonders viele Zählschleifen finden sich in stark befahrenen Gebieten wie dem Autobahndreieck Essen-Ost oder dem Autobahnkreuz Herne. An weniger stark frequentierten Autobahnabschnitten, wie dem nördlichen Rand des Ruhrgebiets, wurden deutlich weniger

---

<sup>1</sup><http://www.uni-due.de/ptt/>

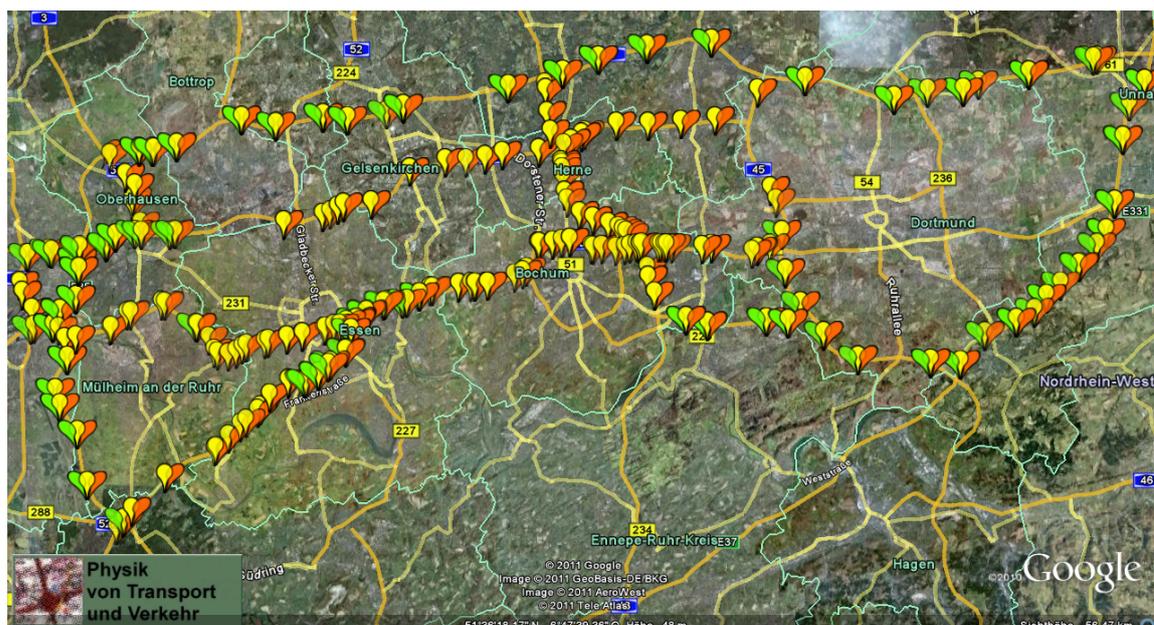


Abbildung 3.1: Positionen der Zähl Schleifen auf den Autobahnen des Ruhrgebiets. Die Farbe der Marker kennzeichnet die durch den Sensor überwachte Spur: rot  $\hat{=}$  rechte Spur, gelb  $\hat{=}$  zweite Spur von rechts, grün  $\hat{=}$  dritte Spur von rechts.

Sensoren installiert.

## 3.2 Erfasste Größen

Die gemessenen Werte werden für jede Zähl Schleife über 60 Sekunden aggregiert und anschließend übermittelt. Der Datensatz beinhaltet 765 Beispiele pro Minute und setzt somit eine in Abschnitt 2.3.1 vorgestellte makroskopische Betrachtung des Verkehrssystems voraus.

Aufgrund der primitiven Natur von Induktionsschleifen enthält jedes Beispiel lediglich fünf direkte Verkehrsgrößen. Wie der Term „Zähl Schleife“ bereits suggeriert wird die Anzahl der Fahrzeuge erhoben. Die Anzahl von LKW pro Minute ( $j_{lkw}$ ) wird dabei getrennt erfasst. Zusammen mit der Anzahl aller Fahrzeuge pro Minute ( $j_{ges}$ ) ist die Zahl der PKW somit implizit durch

$$j_{pkw} = j_{ges} - j_{lkw} \quad (3.1)$$

gegeben.

Auch die Durchschnittsgeschwindigkeit pro Minute wird in Fahrzeugklassen aufgeteilt. Hier wird jedoch ein Wert für PKW ( $v_{pkw}$ ) und LKW ( $v_{lkw}$ ) übermittelt. Eine Durchschnittsgeschwindigkeit für alle Fahrzeugtypen ist weniger aufschlussreich

und somit nicht enthalten. Da die Aufteilung durch einen festen Schwellwert für die maximale Länge eines PKW erfolgt, sind z. B. Motorräder der Fahrzeugklasse PKW zuzurechnen. Ebenso werden Busse zur Klasse der LKW gezählt.

Die letzte direkt erfasste Größe ist der relative Anteil, den die jeweilige Zählschleife im Aggregationsintervall belegt war. Diese Belegung ( $p\_bel$ ) ist eine wichtige Verkehrsgröße (siehe auch Abschnitt 3.4 in diesem Kapitel), die nicht weiter in Fahrzeugklassen aufgeteilt wird.

### 3.3 Metadaten des Datensatzes

Neben den erfassten Messwerten besitzt der Datensatz eine Reihe von Informationen über die Zählschleifen selbst. Es handelt sich hierbei bspw. um Positionsangaben (Breiten- und Längengrad) oder der Fahrstreifenposition (rechts, mittig, links). Darüber hinaus besitzt jedes Beispiel eine boolesche Variable mit dem Index *defekt*. Nicht alle Zählschleifen übermitteln zu jedem Zeitpunkt Werte. Wenn ein Beispiel keine der erfassten Größen enthält ist *defekt* mit 1 bzw. *true* belegt. Über den gesamten Zeitraum für den Daten vorliegen beträgt die Quote der defekten Sensoren 21,47%. In Tabelle 3.1 ist eine Übersicht aller 18 Merkmale eines Beispiels zu finden.

Tabelle 3.1: Attribute der Verkehrsdaten aus Induktionsschleifendetektion.<sup>2</sup>

Spalte	Name	Typ	Beschreibung
1	zeit	timestamp with time zone	Form: „2010-12-24 01:00:00+01“ - GMT 00:00 bis 24:00
2	minute_tag	smallint NOT NULL	GMT 00:00 = minute_tag „0“ bis GMT 23:59 = minute_tag „1439“
3	zs	integer NOT NULL	ID der Zählschleife
4	j_ges	smallint	minutenaggregierte Anzahl Fhz. / min. (PKW + LKW)
5	j_lkw	smallint	minutenaggregierte Anzahl LKW/min.
6	v_pkw	smallint	minutenaggregierte mittlere Geschwindigkeit PKW in km/h
7	v_lkw	smallint	minutenaggregierte mittlere Geschwindigkeit LKW in km/h
8	p_bel	smallint	minutenaggregierte Belegung in %
9	x_pos	numeric	WGS-84 Longitude
10	y_pos	numeric	WGS-84 Latitude
11	dav_name	character varying(50)	ID des Messquerschnitts
12	fahrstreifen	smallint	Fahrstreifen-Position der Zählschleife, „0“ = rechte Spur
13	spuren	smallint	maximale Anzahl von Fahrstreifen an diesem Messpunkt
14	otdf_id	character varying(14)	PTT-ID des Fahrbahnabschnitts <sup>3</sup>
15	station	integer	Abstand (in Metern) zur letzten Anschlussstelle
16	„from“	character varying(100)	Fahrbahnabschnitt von . . .
17	towards	character varying(100)	Fahrbahnabschnitt bis . . .
18	defekt	boolean	„FALSE“ für Zählschleife mit Messwerten ansonsten „TRUE“

<sup>2</sup>Tabelle von Dipl.-Phys. Dominik Wegerle, Physik von Transport und Verkehr, Universität Duisburg-Essen.

<sup>3</sup>die otdf\_id setzt sich aus der Autobahnnummer, der Fahrtrichtung (**NO** oder **SW**), des Fahrbahntyps (z.B. **HF**=Hauptfahrbahn) und der Nummer der nächsten Anschlussstelle zusammen; Bsp.: „**A040-NO-HF-002**“steht für die Hauptfahrbahn der A40 in östlicher Richtung ab der Anschlussstelle „Grenzübergang Straelen(1)“bis zur Anschlussstelle „Straelen(2)“.

### 3.4 Abgeleitete Größen

Weil die Messwerte der Beispiele minutenaggregiert sind, können zur Modellierung verwendete Lernverfahren nicht auf Daten von Einzelfahrzeugen zurückgreifen, sondern sind auf makroskopische Größen angewiesen. Zwei wichtige Größen in diesem Zusammenhang sind der Verkehrsfluss  $Q(x, t)$  und die Verkehrsdichte  $\rho(x, t)$  (vgl. Abschnitt 2.3.1). Eine Bestimmung von  $\rho(x, t)$  sowie  $Q(x, t)$  erfolgt mithilfe der Belegung und der Durchschnittsgeschwindigkeiten.

Der Verkehrsfluss  $Q(x, t)$  beschreibt die Anzahl an Fahrzeugen, die in einem Zeitintervall  $\Delta t$  den Ort  $x$  überqueren. Da die Messstationen ortsfeste Zählschleifen sind, ergibt sich der Verkehrsfluss  $Q_{zs}(t)$  für eine Zählschleife  $zs$  in diesem Fall aus (3.2) [43].

$$Q(x, t) \hat{=} Q_{zs}(t) = j\_ges_{zs}(t) \cdot 60 \text{ [Fhz./h]}. \quad (3.2)$$

Die Verkehrsdichte  $\rho(x, t)$  bezeichnet die Anzahl von Fahrzeugen pro Längeneinheit. Zu ihrer Approximation existieren zwei Ansätze. Beim ersten Ansatz handelt es sich um eine Analogie zu strömenden Fluiden in der Hydromechanik [43]. Aus der Fluss-Dichte-Beziehung (2.9) ergibt sich die Dichte  $\rho_{zs}$  für den vorliegenden Fall durch (3.3):

$$\begin{aligned} \rho(x, t) \hat{=} \rho_{zs}(t) &= \frac{Q_{zs}(t)}{v_{zs}(t)} \\ &= \frac{j\_ges_{zs}(t) \cdot 60}{v_{zs}(t)} \text{ [Fhz./km]}. \end{aligned} \quad (3.3)$$

Für die zweite Variante zur Bestimmung von  $\rho_{zs}(t)$  muss die mittlere Fahrzeuglänge bekannt sein oder festgelegt werden. Bei diesem Ansatz wird die mittlere Zeit, die eine Zählschleife im Intervall  $\Delta t$  belegt ist, durch die mittlere Fahrzeuglänge geteilt [43]. Es resultiert daraus mit (3.4) eine alternative Formel zur Bestimmung der Dichte.

$$\rho_{zs}(t) \hat{=} \rho(x, t) = \frac{p\_bel_{zs}(t)}{100 \cdot \bar{l}} \text{ [Fhz./km]}. \quad (3.4)$$

Wenn die Messdaten wie hier durch Induktionsschleifen gewonnen werden, wird (3.4) bevorzugt. Dies liegt daran, dass für (3.3) eigentlich eine *räumliche* Durchschnittsgeschwindigkeit für einen Abschnitt erforderlich ist. Die Näherung der Dichte kann somit äußerst stark von der wahren Verkehrsdichte abweichen. Auch (3.4) liefert keinesfalls einen exakten Wert der eigentlichen Verkehrsdichte, ist allerdings für *zeitliche* Durchschnittsgeschwindigkeiten definiert und liefert deshalb bessere Näherungen [94]. In dieser Arbeit wird also (3.4) zur Bestimmung der Verkehrsdichte verwendet. Als Länge wird der vom Lehrstuhl für Physik von Verkehr und Transport bereitgestellte Wert  $\bar{l} = 7,5\text{m}$  übernommen.

### 3.5 Bestimmung der Verkehrsphase

Wenn sowohl die Dichte  $\rho_{zs}$  als auch die Geschwindigkeit  $v_{zs}$  für eine Zählschleife  $zs$  zum Zeitpunkt  $t$  bekannt sind kann die aktuelle Phase des Verkehrs an  $zs$  immer bestimmt werden. Die Bestimmung der Phase erfolgt durch die Einteilung des *Fundamentaldiagramms* aus Abbildung 3.2 in drei Bereiche. Die Schwellwerte für diese Einteilung orientieren sich hierbei an den im HCM [1] definierten Werten zur Unterscheidung von level-of-service Stufen im Verkehr.

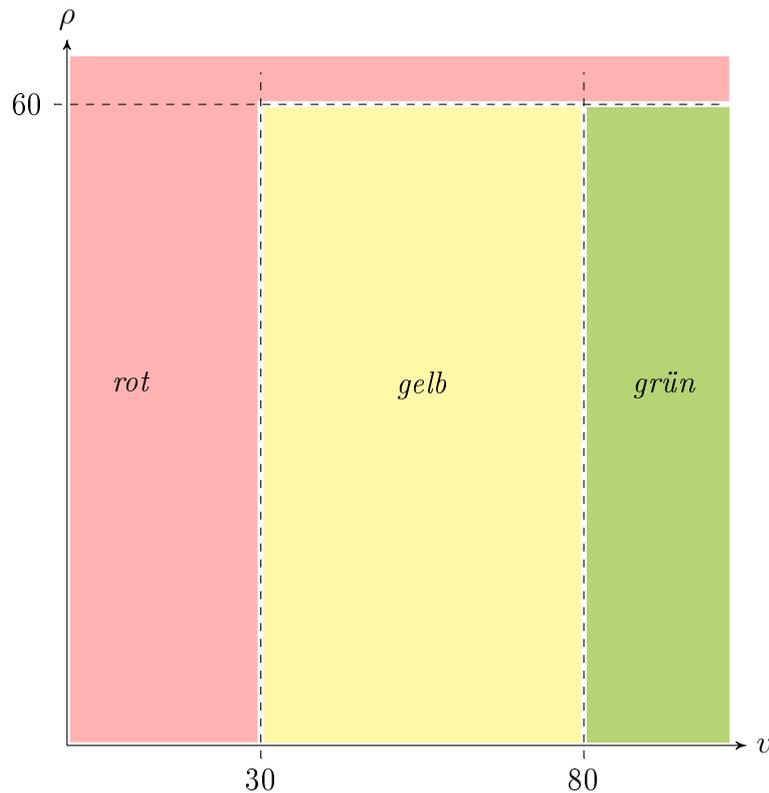


Abbildung 3.2: Die Bestimmung der Verkehrsphase erfolgt durch Auswertung von Geschwindigkeit  $v$  und Dichte  $\rho$ . Es existieren drei Zustände des Verkehrssystems: *grün*  $\hat{=}$  freier Fluss – *gelb*  $\hat{=}$  synchroner Fluss – *rot*  $\hat{=}$  Stau.

Semantisch stimmt die Einteilung in die drei Bereiche aus Abbildung 3.2 mit den in Abschnitt 2.3.2 definierten Verkehrsphasen überein:

- *grün*  $\hat{=}$  freier Fluss,
- *gelb*  $\hat{=}$  synchroner Fluss,
- *rot*  $\hat{=}$  sich bewegender breiter Stau.

### 3.6 Festlegung der Ausgaben $y$ des Datensatz

Ziel dieser Arbeit ist die Vorhersage von Zustandsübergängen. Bei den drei möglichen Zuständen aus Abschnitt 2.3.2 sind maximal neun unterschiedliche Zustandsübergänge möglich. Das Verweilen in einem Zustand wird in diesem Kontext jedoch nicht als Zustandsübergang angesehen, sodass lediglich sechs wahre Transitionen existieren:

- $grün \rightarrow gelb$ ,
- $grün \rightarrow rot$ ,
- $gelb \rightarrow grün$ ,
- $gelb \rightarrow rot$ ,
- $rot \rightarrow grün$ ,
- $rot \rightarrow gelb$ .

Die beiden besonderen Zustandsübergänge Verkehrszusammenbruch und Entstehung von Schockwellen aus Abschnitt 2.3.3 sind also durch die Ausgaben  $grün \rightarrow gelb$  und  $gelb \rightarrow rot$  beschrieben.

Durch den Determinismus bei der Bestimmung der Verkehrsphasen ist die Erzeugung von mit Ausgaben versehenen Daten für die Trainings- und Testphase problemlos möglich.

Zum Zeitpunkt  $t$  kann die Verkehrsphase für ein Beispiel  $\mathbf{X}_t$  mit dem Verfahren aus Abschnitt 3.5 bestimmt werden. Dieses  $y_t$  ist der erste Teil der Ausgabe von  $\mathbf{X}_t$ . Nachdem der Vorhersagehorizont  $\Delta t_h$  abgelaufen ist kann die Verkehrsphase für den Zeitpunkt  $(t + \Delta t_h)$  bestimmt werden.

Die temporäre Ausgabe  $y_t$  vom zwischengespeicherten Beispiel  $\mathbf{X}_t$  wird dann zu  $y'_t = "y_t \rightarrow y_{(t+\Delta t_h)}"$ . Nur wenn  $y_t \neq y_{(t+\Delta t_h)}$  gilt, muss das Tupel  $(\mathbf{X}_t; y'_t)$  zur Beispielmenge  $\mathbf{Z}$  hinzugefügt werden, da nur bei unterschiedlichen Verkehrsphasen auch ein Zustandsübergang vorliegt.



## Motifs

Neben der Entdeckung von globalen Charakteristiken in Zeitreihen (z.B. durch eine Spektralanalyse [62]) ist die Suche nach *Motifs* eine weitere Möglichkeit zur Extraktion von Wissen aus Zeitreihen. Durch die Zerlegung der Zeitreihe in *Teilfolgen* werden hier jedoch vielmehr lokale Eigenschaften gesucht.

### DEFINITION 4.1: Teilfolge

Eine **Teilfolge**  $C$  einer Zeitreihe  $T$  mit einer Länge  $n$  ist eine Sequenz mit einer Länge  $k < n$ , die fortlaufende Elemente  $c_i, \dots, c_{i+(k-1)}$  aus  $T = \{c_1, \dots, c_i, \dots, c_{i+(k-1)}, \dots, c_n\}$  enthält.

Um Motifs unter diesen Teilfolgen zu bestimmen werden *Übereinstimmungen* gesucht.

### DEFINITION 4.2: Übereinstimmung

Gegeben sei ein reellwertiger Schwellwert  $R$  und eine Zeitreihe  $T$  mit den Elementen  $c_1, \dots, c_n$ . Ein Paar von Teilfolgen  $(C, C')$  ist eine **Übereinstimmung**, wenn gilt:

- $\text{dist}(C, C') \leq R$ , für eine beliebige Distanzfunktion
- $C = c_i, \dots, c_j, C' = c_k, \dots, c_l$ , mit  $j < k$ .

Zwei disjunkte Teilfolgen einer Zeitreihe bilden also Übereinstimmung, wenn sie unter Verwendung einer Distanzfunktion in einer Nachbarschaft mit Radius  $R$  liegen. Nicht disjunkte Teilfolgen sind von der Definition einer Übereinstimmung ausgeschlossen, da i. d. R. ein Paar  $(C = c_i, \dots, c_j, C' = c_{i+1}, \dots, c_{j+1})$  immer eine *triviale* Übereinstimmung bildet [68].

### BEISPIEL 4.1: Übereinstimmung

In der bereits vorgestellten Zeitreihe des Einzelhandelsumsatz in Deutschland findet sich eine sehr anschauliche Übereinstimmung. Wie in Abbildung 4.1 zu sehen ist, tritt gegen Ende eines jeden Jahres stets ein großer Sprung auf. Zunächst sinkt das Konsumverhalten im November, um im Dezember ein Jahreshoch zu erreichen. Dieser positive Trend resultiert im Januar in einem dramatischen Absinken der Kaufkraft.

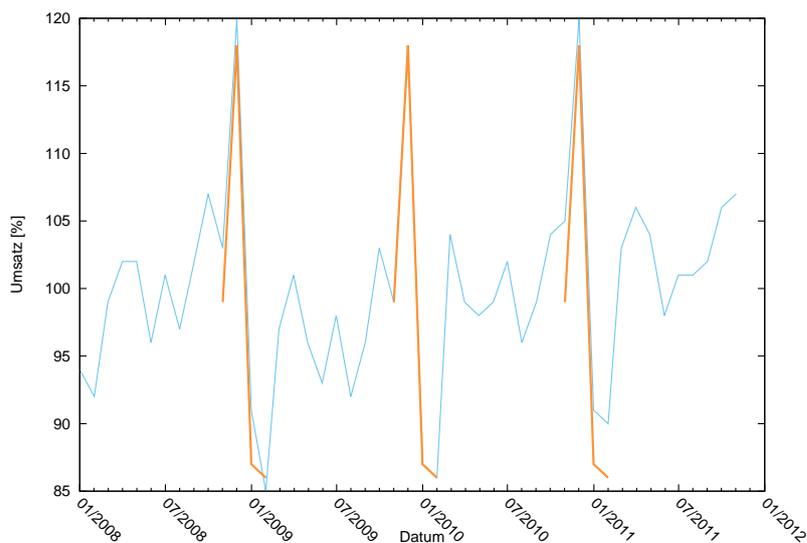


Abbildung 4.1: Motif im Verlauf des Einzelhandelsumsatz.

*Der negative Trend wird im Februar – wenn auch gebremst – fortgeführt. Auch wenn die Ausschläge von Jahr zu Jahr schwanken ist doch eine grundsätzliche ähnliche Form zu beobachten.*

Nicht jede Übereinstimmung ist ein Motif. Dies ist vielmehr das notwendige Kriterium für ein Motif:

**DEFINITION 4.3: Notwendiges Kriterium für ein Motif**

*Eine Teilfolge  $C$  einer Zeitreihe  $T$  kann nur dann ein Motif sein, wenn  $C$  in mindestens einer Übereinstimmung vorkommt.*

Welche Teilfolgen aus der Kandidatenmenge aller Übereinstimmungen letztendlich Motifs sind, hängt von der *Signifikanz* der einzelnen Teilfolgen ab. Die Signifikanz ist nach [68] die Häufigkeit mit der eine Teilfolge ein Partner in einer Übereinstimmung ist.

Wie viele Motifs in einer Zeitreihe (bzw. in einem betrachteten Abschnitt einer Zeitreihe) gefunden werden, hängt nicht nur von dem gewählten Schwellwert  $R$  ab. Wenn nur das signifikanteste Motif einer Zeitreihe bestimmt wird, ist dies das *1-Motif* [68].

**DEFINITION 4.4: 1-Motif**

Für eine Zeitreihe  $T$  ist das **1-Motif** diejenige Teilfolge mit den meisten Übereinstimmungspartnern in einer Nachbarschaft mit Radius  $R$ .

Wenn jedoch die  $k$  signifikantesten Motifs gesucht werden, müssen  $k$ -Motifs bestimmt werden [68].

**DEFINITION 4.5: k-Motif**

Das **k-Motif** einer Zeitreihe  $T$  ist die Teilfolge mit der  $k$ -größten Signifikanz unter allen potentiellen Motifs. Diese Teilfolge muss jedoch mindestens einen Abstand von  $2R$  zu allen signifikanteren  $k-1$  Motifs haben, da nur disjunkte Mengen von Übereinstimmungen betrachtet werden dürfen.

Dadurch besitzen Motifs Ähnlichkeiten zu Mustern. Während das Auffinden von Mustern jedoch das Ziel hat, im Vorfeld bekannte Teilfolgen zu detektieren, bezweckt die Suche nach Motifs das Auffinden bisher *unbekannter* „Muster“ [68].

## 4.1 Ursprung

Der Begriff „Motif“ wurde für Zeitreihen in [68] von Lonardi et. al eingeführt. Er entstammt ursprünglich der Bioinformatik (siehe z. B. [23, 54]). DNA-Sequenzen werden üblicherweise als Folge von Symbolen dargestellt. In der Medizin werden Motifs u. a. verwendet, um bestimmte Regulatoren in DNA-Sequenzen zu finden [66].

Auch Zeitreihen wurden anfänglich als Folge von Symbolen eines endlichen Alphabets interpretiert, um Motifs zu finden. Dieses *SAX-Verfahren* wurde von Lonardi et. al in [64] eingeführt. Die (reellen) Werte der Zeitreihe werden zunächst durch ein *Binning* in disjunkte Intervalle eingeteilt. Für jedes dieser Intervalle wird ein Symbol als Repräsentation gewählt. Ausgehend von dieser Art der Vorverarbeitung sind viele Verfahren zur Motif-Extraktion entstanden (u. a. [15, 49, 57, 64, 67, 73, 72, 88]).

Diese Vorverarbeitung ermöglicht es Verfahren für Folgen diskreter Werte auch für die Motifsuche in reellwertigen Zeitreihen zu verwenden. So wird es bspw. möglich, das zu betrachtende System als *Markov-Prozess* zu modellieren. Allerdings bringt die Diskretisierung auch Nachteile mit sich. Mörchen und Ultsch zeigen in [73], dass Systeme im Allgemeinen unterschiedliche Phasen durchlaufen. Eine Phase ist dadurch abgegrenzt, dass die Wahrscheinlichkeit im nächsten Zeitschritt in der selben Phase zu verbleiben größer ist, als die Wahrscheinlichkeit für einen Phasenwechsel. Da die Intervalle der unterschiedlichen Phasen i. d. R. nicht bekannt sind, besteht beim Binning die Gefahr, durch unvorteilhaft gewählte Intervallgrenzen eigentlich zusammenhängende Bereiche des Werteraums künstlich zu trennen.

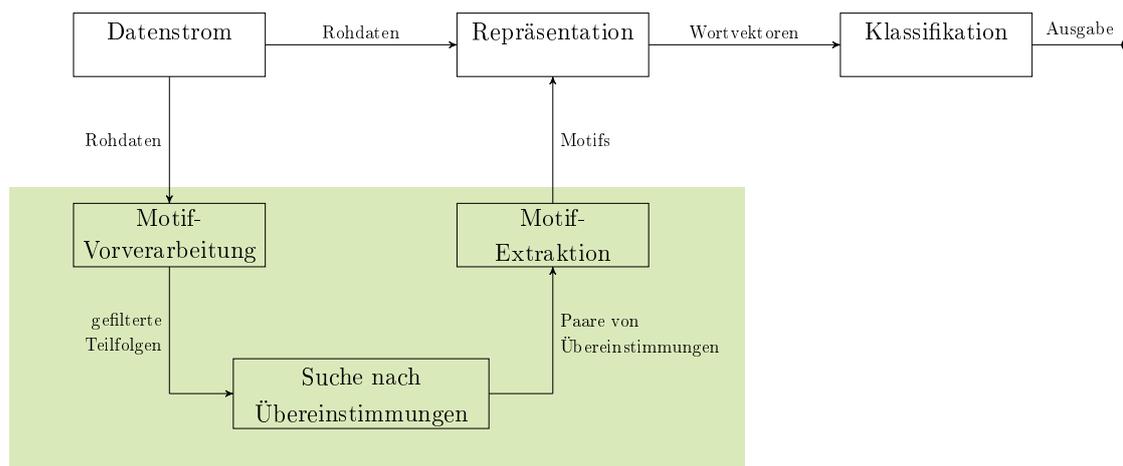


Abbildung 4.2: Die folgenden Abschnitte in diesem Kapitel widmen sich der Extraktion von Motifs. Zunächst werden die Teilfolgen durch Vorverarbeitungsschritte vorbereitet, um nach Übereinstimmungen durchsucht zu werden. Anschließend werden aus den Übereinstimmungen Motifs gewonnen.

Wie bereits in Abschnitt 2.3 erläutert, sind in Verkehrsdaten unterschiedliche Phasen zu beobachten. Darüber hinaus ist die Vorhersage von Phasenübergängen das Ziel der hier bearbeiteten Lernaufgabe. Deshalb wird in dieser Arbeit ein jüngerer Ansatz ohne Binning als Grundlage für den Motif-Algorithmus verwendet. Es handelt sich um den Ansatz zur Extraktion *exakter Motifs* aus [75], der in [74] zu einem Online-Algorithmus weiter entwickelt wurde.

Da sich das hier Verwendete Verfahren an einigen Stellen von der in den genannten Arbeiten vorgestellten Version unterscheidet, enthält der nächste Abschnitt eine Erläuterung des implementierten Verfahrens.

## 4.2 Algorithmus zur Extraktion von Motifs

Die nachfolgenden Abschnitte dieses Kapitels behandeln die drei unteren, grün eingefärbten Kästen in Abbildung 4.2. Es wird erläutert, wie im Rahmen dieser Arbeit Motifs zur späteren Repräsentation von Zeitreihen gewonnen werden.

### 4.2.1 Vorüberlegungen

Die Intention dieser Arbeit ist es Phasenübergänge zu erkennen und die entsprechenden Teilfolgen der Zeitreihe für die Vorhersage zu nutzen. Wie aus Abschnitt 2.3.3 bekannt ist, sind Phasenübergänge in Relation zu den Phasen selbst relativ kleine Zeiträume. Somit sind Teilfolgen, die Phasenübergänge beschreiben, als seltene Elemente in Bezug auf die Grundgesamtheit aller Teilfolgen des Datenstroms anzusehen.

Da der Algorithmus zur Extraktion von Motifs die Auftrittshäufigkeit in Übereinstimmungen als Kriterium für die Signifikanz einer Teilfolge verwendet, ist es zu erwarten, dass Motifs für die Gesamtmenge aller Teilfolgen die eigentlich interessanten Bereiche des Datensatzes nur schlecht beschreiben. Der Algorithmus wurde deshalb um eine Vorverarbeitungsstufe erweitert, die interessante Teilfolgen markiert.

Der Basis-Algorithmus sieht darüber hinaus eine feste Anzahl von Motifs pro Fenster vor. Im betrachteten Anwendungsfall ist dies jedoch nicht von Vorteil, da im Vorfeld nicht klar ist, wie viele Motifs pro Fenster zu erwarten sind. Die Anzahl wird außerdem von Fenster zu Fenster schwanken. Und nicht zuletzt muss eine Teilfolge, die in einem Fenster häufig in Übereinstimmungen vorkommt, auch kein Motif in der gesamten Trainingsmenge sein. Um dieses Problem zu behandeln werden in der hier verwendeten Abwandlung des Algorithmus Übereinstimmungen fensterübergreifend gesammelt. Aus dieser Kandidatenmenge werden anschließend Motifs extrahiert.

## 4.2.2 Vorverarbeitung

### Fensterung

Es werden zwei unterschiedliche Fensterungen benötigt, um online auf dem Datenstrom nach Motifs zu suchen. Neben dem Hauptfenster, das die Teilfolgen für den Motif Algorithmus enthält [74], wird ein weiteres Fenster zur Erzeugung der Teilfolgen selbst benötigt. Diese Fensterung fast jedoch nicht nur  $k$  Elemente des Datenstroms zu einer Teilfolge zusammen sondern verringert darüber hinaus das Rauschen der Messdaten.

Der verwendete Datensatz enthält einen Wertevektor  $\mathbf{x}_i$  pro Zeitschritt  $t_i$ . Die zeitliche Auflösung von einer Minute pro Zeitschritt ist für die Untersuchung von Verkehrssystemen auf Autobahnen weniger gut geeignet, weil den Messwerten ein stark ausgeprägtes Rauschen unterliegt [21, 92]. Ein gängiges Mittel zur Reduzierung von Rauschen ist, die Werte der Reihe durch den gleitenden Mittelwert zu ersetzen. Für einen gewählten Horizont  $h$  wird das Element an Stelle  $i$  durch den Mittelwert der letzten  $h$  Elemente ersetzt:

$$\bar{c}_i = \frac{1}{h} \cdot \sum_{j=i-(h-1)}^i c_j. \quad (4.1)$$

Mit steigendem  $h$  sinkt zwar das Rauschen, jedoch haben aktuelle Ereignisse des Datenstroms immer weniger Einfluss auf die  $\bar{c}_i$ . Deshalb wurde für die nachfolgenden Verfahren und Experimente das in der Literatur gefundene Minimum für eine sinnvolle Reduzierung des Rauschens von Induktionsschleifendaten verwendet. Es handelt sich hierbei um einen Horizont von  $h = 4$  Minuten [78].

Das Resultat dieses Schrittes der Vorverarbeitung ist ein Datenstrom bestehend aus  $k$ -elementigen Teilfolgen der ursprünglichen Zeitreihe:

$$C_i = \langle \bar{c}_i, \bar{c}_{i+1}, \dots, \bar{c}_{i+(k-1)} \rangle$$

Jedes  $\bar{c}_i$  ist dabei der Mittelwert über die letzten vier Elemente der Zeitreihe:

$$\bar{c}_i = \frac{c_{i-3} + c_{i-2} + c_{i-1} + c_i}{4} \quad (4.2)$$

### Markierung und Filterung der interessanten Teilfolgen

Damit die extrahierten Motifs auch tatsächlich die gesuchten Phänomene von Phasenübergängen abdecken und somit zur Erkennung solcher geeignet sind, muss der Datenstrom gefiltert werden. In Abschnitt 2.3.3 wurde erwähnt, dass die Varianz der Werte zwischen zwei Phasen gegenüber der Werte während einer Phase deutlich erhöht ist. Deshalb werden zunächst die Varianzen der Werte aller Teilfolgen der Trainingsmenge analysiert.

Es wird ein Quantil-Algorithmus verwendet, um sinnvolle Schwellwerte für die Varianz von interessanten Teilfolgen zu bestimmen. Für die in der Trainingsphase betrachteten Sensoren kann so durch Festlegung eines  $\phi$ -Werts für jede Teilfolge entschieden werden, ob sie zur Menge der interessanten Teilfolgen gehört oder nicht für die Motif-Extraktion verwendet werden soll. Hierbei gilt, dass je höher  $\phi$  gewählt wird, desto weniger Teilfolgen werden betrachtet, da die Filterung  $\phi \cdot 100\%$  aller Teilfolgen aus dem Datenstrom entfernt.

In Abbildung 4.3 ist das Zusammenspiel der einzelnen Vorverarbeitungsstufen noch einmal grafisch dargestellt.

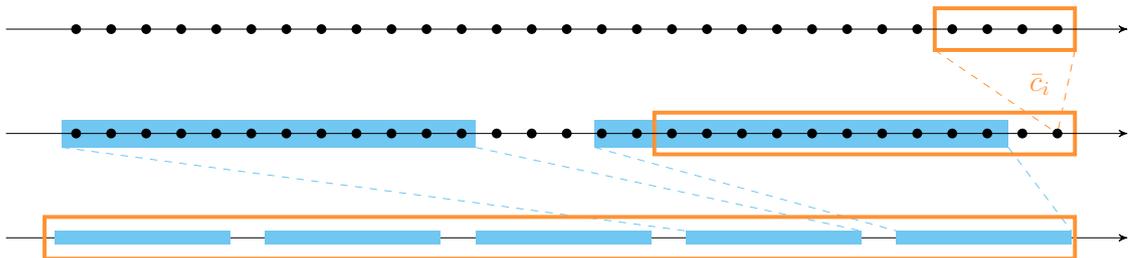


Abbildung 4.3: Abbildung der hier verwendeten Fensterungen. Zuoberst ist die Reduzierung des Rauschens dargestellt. Aus jeweils vier  $c_i$  wird der Mittelwert  $\bar{c}_i$  bestimmt. Für diese Mittelwerte wird mit dem mittleren Fenster die Varianz der einzelnen Teilfolgen  $C_j$  der Länge  $k$  bestimmt. Überschreitet die Varianz den Schwellwert für interessante Teilfolgen wird die Teilfolge markiert (blaue  $C_j$  im Schaubild) und in das eigentliche Fenster zur Suche nach Übereinstimmungen (unten) eingefügt.

### 4.2.3 Finden von Übereinstimmungen

Ziel dieses Moduls ist es, die beste Übereinstimmung  $(C, C_i)$  für jede Teilfolge im Fenster zu bestimmen. Dabei wird eine Teilfolge solange aktualisiert, bis sie das Fenster verlässt. Ein hieraus resultierendes Paar  $(C, C_i)$  impliziert also, dass während

der gesamten Verweilzeit von  $C$  im Fenster keine Teilfolge  $C_j$ , mit  $j \neq i$ , eine größere Ähnlichkeit zu  $C$  hatte als  $C_i$ .

Das hieraus resultierende Paar  $(C, C_i)$  sagt also aus, dass während der gesamten Verweildauer von  $C$  im Fenster, keine Teilfolge eine größere Ähnlichkeit aufwies als  $C_i$ .

Nach [75] ist die Ähnlichkeit zweier Teilfolgen durch ihre euklidische Distanz aus (2.1) definiert – je geringer der Abstand, desto größer die Ähnlichkeit (siehe Abbildung 4.4). Um die beste Übereinstimmung für eine Teilfolge  $C$  zu bestimmen, berechnet ein naiver Algorithmus also den paarweisen Abstand zu sämtlichen anderen Teilfolgen des aktiven Fensters. Da dieser Ansatz jedoch für realistische Fensterbreiten aufgrund der hohen Rechenzeiten nicht online-fähig ist, wird das im Folgenden beschriebene Verfahren verwendet. Die Teilfolgen werden hierzu auf einem *Ordnungsstrahl* platziert.

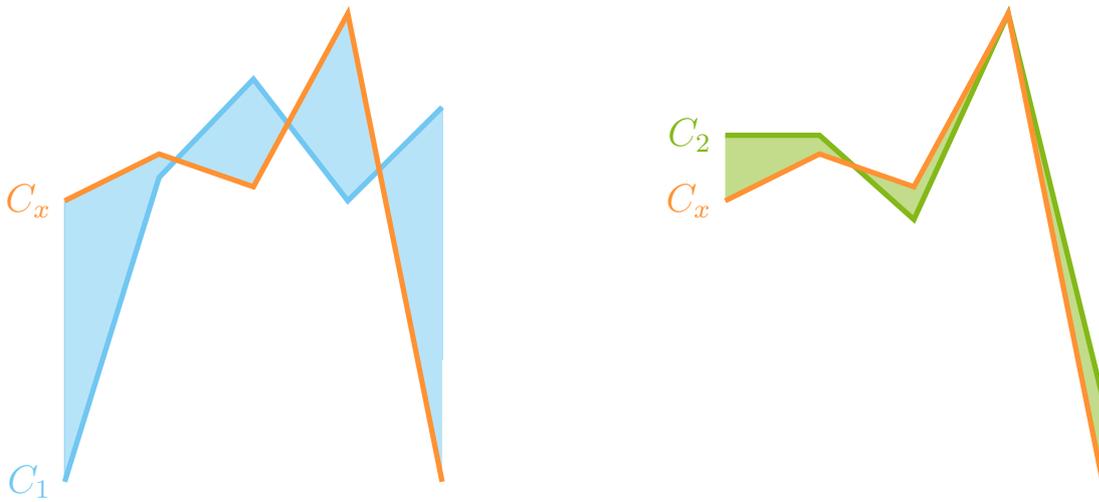


Abbildung 4.4: Zur Bestimmung der Ähnlichkeit von Teilfolgen wird der paarweise Abstand berechnet. In diesem zweidimensionalen Beispiel entspricht dieser Abstand der schattierten Flächen. Es wird hierbei der euklidische Abstand (siehe (2.1)) minimiert. In der Abbildung gilt  $dist_{euklid}(C_x, C_2) < dist_{euklid}(C_x, C_1)$ , weshalb das Paar  $(C_x, C_2)$  eine Übereinstimmung bilden.

### Ordnungsstrahl

Um die Anzahl der kostspieligen Berechnungen der euklidischen Distanz zweier Teilfolgen zu minimieren, wird der in [75] vorgestellte Ordnungsstrahl verwendet. Mithilfe eines Referenzvektors kann ein Ordnungsstrahl zur Generierung einer Kandidatenmenge für die Nachbarschaft zu einer Teilfolge  $C$  bestimmt werden.

Für einen beliebigen Referenzvektor  $\mathbf{r}$  wird eine sortierte Liste der Teilfolgen verwaltet. Kriterium für die Sortierung ist der Abstand der Teilfolge zu  $\mathbf{r}$ , sodass die

Teilfolge mit dem geringsten Abstand zu  $\mathbf{r}$  sich am Anfang der Liste befindet und die Indizes der weiteren Teilfolgen mit steigendem Abstand zu  $\mathbf{r}$  ebenfalls ansteigen. Diese Liste kann nun zur Bestimmung einer Kandidatenmenge für die Nachbarschaft beliebiger Teilmengen verwendet werden. Es muss lediglich das *notwendige Kriterium für Nachbarschaftskandidaten* geprüft werden:

**DEFINITION 4.6: Notwendiges Kriterium für Nachbarschaftskandidaten**  
 Sei  $\text{dist}_{\text{euklid}}(\mathbf{r}, C) = d$ . Der Radius der Nachbarschaft um  $C$  sei  $R$ . Dann gilt für alle Nachbarn  $C_i$  von  $C$ :

$$d - R \leq \text{dist}_{\text{euklid}}(\mathbf{r}, C_i) \leq d + R \quad (4.3)$$

Es genügt nun also, ausgehend vom Index der Teilfolge  $C$  alle Teilfolgen mit kleinerem Index und einem Abstand zu  $\mathbf{r}$  der höchstens um  $R$  kleiner ist, sowie alle Teilfolgen mit größerem Index und einem Abstand zu  $\mathbf{r}$  der höchstens um  $R$  größer ist zu betrachten, um alle Nachbarn von  $C$  zu finden.

In Abbildung 4.5 wird dieses Verfahren für den zweidimensionalen Raum noch einmal erläutert. In diesem Beispiel sollen alle Nachbarn von Punkt  $\mathbf{x}$  ermittelt werden. Zuerst wird der Abstand  $d$  von  $\mathbf{x}$  zum Referenzvektor  $\mathbf{r}$  bestimmt. Abhängig von  $d$  wird  $\mathbf{x}$  auf dem Ordnungsstrahl (rechts) platziert. Das notwendige Kriterium für Nachbarschaftskandidaten (siehe (4.3)) besagt nun, dass alle Nachbarn auf dem Ordnungsstrahl eine Distanz  $d'$  zum Referenzvektor haben müssen, die im Bereich  $[d - R; d + R]$  liegt. In Abbildung 4.5 sind dies die Punkte 2, 3 und 4, welche die Kandidatenmenge  $\text{cand}(\mathbf{x})$  für die Nachbarschaft von Punkt  $\mathbf{x}$  bilden.

Im linken Teil von Abbildung 4.5 zeigt sich, dass dieses Verfahren in der Tat alle Kandidaten für die Nachbarschaft erfasst. Der Wertebereich  $[d - R; d + R]$  für die gültigen Distanzen ist grün eingefärbt. Außerdem ist der Radius  $R$  der Nachbarschaft von  $\mathbf{x}$  durch die gestrichelte Linie um  $\mathbf{x}$  angetragen. Es zeigt sich anschaulich, dass alle Nachbarn zu  $\mathbf{x}$  im grün eingefärbten Bereich um den Referenzvektor liegen müssen. Bei der Distanz zweier Vektoren auf dem Ordnungsstrahl handelt es sich um die untere Schranke für die wahre Distanz [75]. Deshalb wird die paarweise euklidische Distanz aller Vektoren aus der Kandidatenmenge  $\text{cand}(\mathbf{x})$  zu  $\mathbf{x}$  bestimmt. Die beste Übereinstimmung ist das Paar mit minimalem euklidischen Abstand.

Da das notwendige Kriterium aus (4.3) für beliebige Vektoren gilt, kann die Kandidatenmenge drastisch verkleinert werden. Bei einem gleichzeitigen Einsatz von  $n$  Referenzvektoren  $(\mathbf{r}_1, \dots, \mathbf{r}_n)$  ist die Kandidatenmenge mit minimaler Anzahl von Vektoren für die Nachbarschaft einer Teilfolge  $C$  gegeben durch die Schnittmenge der einzelnen Kandidatenmengen  $\text{cand}_i(C)$ :

$$\text{cand}_{\text{all}}(C) = \bigcap_{i=1}^n \text{cand}_i(C) \quad (4.4)$$

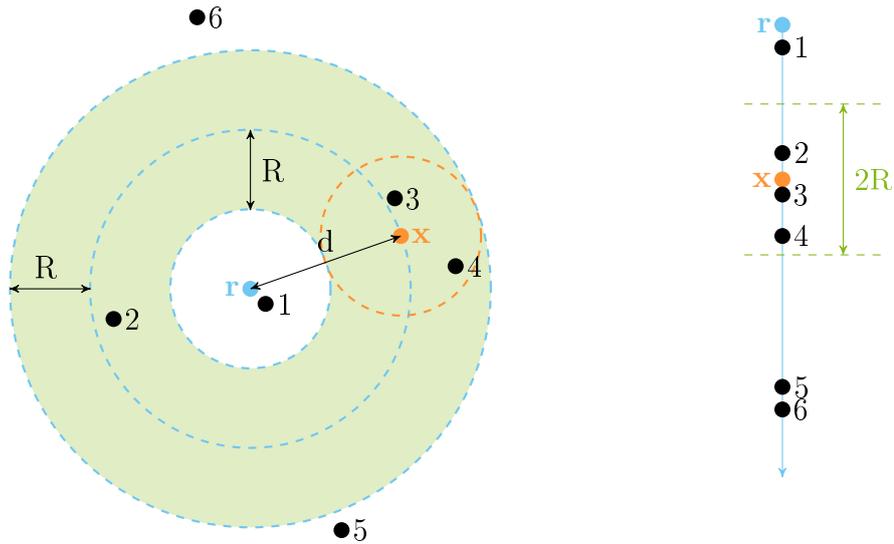


Abbildung 4.5: Grafische Darstellung des Verfahrens des Ordnungsstrahls.

### Aktualisierung des Fensters

Zu jedem Zeitpunkt an dem eine neue Teilfolge  $C$  in das Fenster aufgenommen wird, muss der Abstand des Vektors von  $C$  zum Referenzvektor bestimmt werden, um die Teilfolge auf dem Ordnungsstrahl zu platzieren. Anschließend wird die Kandidatenmenge für die Nachbarschaft von  $C$  bestimmt und durch die paarweise Berechnung der Distanz zwischen allen Kandidaten und  $C$  um Teilfolgen verringert, die nicht in der Nachbarschaft von  $C$  liegen. Dies ist notwendig, um die beste Übereinstimmung für  $C$  zu finden.

Durch die neue Teilfolge  $C$  kann sich jedoch auch die beste Übereinstimmung anderer Teilfolgen im Fenster ändern. Um nicht jede Teilfolge des Fenster überprüfen zu müssen, wird das *notwendige Kriterium für Übereinstimmungen* ausgenutzt:

#### DEFINITION 4.7: Notwendiges Kriterium für Übereinstimmungen

Eine Teilfolge  $C_i$  kann nur dann eine Übereinstimmung zur Teilfolge  $C$  sein, wenn  $C$  und  $C_i$  in einer Nachbarschaft liegen.

Es genügt somit die bereits für jeden Nachbarn  $C_i$  bestimmte Distanz  $dist_{euklid}(C, C_i)$ , mit der Distanz von  $C_i$  zur Teilfolge der bislang besten Übereinstimmung für  $C_i$  zu vergleichen. Nur wenn diese kleiner ist, muss auch die beste Übereinstimmung aktualisiert werden.

Bevor eine Teilfolge  $C_j$  das Fenster verlässt, wird die beste Übereinstimmung  $(C_j, C_m)$  betrachtet und die Teilfolge  $C_m$  zur Kandidatenliste für Motifs hinzugefügt.

Diese stetige Aktualisierung aller Elemente im Fenster verdoppelt die effektive Breite des Fensters, da beim Einfügen einer Teilfolge die beste Übereinstimmung unter den letzten  $w - 1$  Teilfolgen gesucht wird. Ebenso wird diese Übereinstimmung aktualisiert, wenn unter den nächsten  $w - 1$  Teilfolgen eine besser Übereinstimmung gefunden wird. Dies führt dazu, dass jede Teilfolge mit  $2 \cdot (w - 1)$  Elementen verglichen wird, auch wenn die Teilfolge selbst nur  $w$  Zeitpunkte im Fenster verweilt.

### 4.2.4 Extraktion von Motifs

Aus der Menge von ähnlichen Teilfolgen in allen Übereinstimmungen müssen nun Motifs extrahiert werden. Da die Auftrittshäufigkeit hierfür das maßgebende Kriterium ist [68], wird in dieser Arbeit ein *top-k* Algorithmus verwendet. Für eine Menge bzw. einen Strom von Daten ermittelt ein top-k Algorithmus die  $k$  häufigsten Elemente.

Zwar ist die Bestimmung von top-k-Elementen leicht auf unbeschränkten Datenströmen möglich (siehe z. B. [70]), jedoch wurde die Extraktion von Motifs im Rahmen dieser Arbeit auf relativ kleinen Datenmengen durchgeführt. Es kommen darüber hinaus keine neuen Motifs dazu, da Motifs hier einmalig im Vorfeld bestimmt werden um eine Repräsentation des Datenstroms zu ermöglichen. Aus diesem Grund wurde ein exakter top-k Algorithmus verwendet.

### 4.2.5 Parameter für die Motif-Extraktion

Die Ausführungen in diesem Kapitel verdeutlichen, dass bei der Extraktion von Motifs eine große Zahl von Parametern relevant ist. Die im nächsten Kapitel vorgestellte Repräsentation verwendet diese Motifs, um einen Datensatz für die Experimente aus Kapitel 6 zur Verwendung in *RapidMiner*<sup>1</sup> zu erzeugen. Die gewählte Parametrisierung ist also für einen Datensatz und das entsprechende Experiment fix. Es ist im Rahmen dieser Arbeit somit nicht möglich eine automatisierte Optimierung der Parameter der Verfahren in diesem und dem nächsten Kapitel durchzuführen. In diesem Abschnitt werden deshalb die für eine Anwendung in der Praxis relevanten Parameter zur Motif-Extraktion zusammengefasst. Tabelle 4.1 liefert einen Überblick sowie die für die weitere Arbeit gewählten Werte.

Um das Rauschen der Werte zu verringern werden die ursprünglichen Werte der Zeitreihe durch gleitende Mittelwerte mit einem festen Aggregationsintervall  $h_{\bar{c}}$  ersetzt. Die für diese Arbeit verfügbaren Messwerte sind für jeweils eine Minute aggregiert. Diese Auflösung eignet sich besser für die Betrachtung urbaner Verkehrssysteme als für die Untersuchung von Autobahnverkehr [21]. Bei der Wahl von  $h_{\bar{c}}$  muss jedoch auch beachtet werden, dass mit steigendem  $h_{\bar{c}}$  der Einfluss einzelner Werte sinkt und sich somit Änderungen der Werte im Datenstrom erst spät auf den Mittelwert auswirken. Die Kardinalität der Werte für  $h_{\bar{c}}$  ist somit begrenzt – insbesondere da die vorliegenden Daten nur ganzzahlige Werte zulassen.

---

<sup>1</sup><http://rapid-i.com>

Auch für die Wahl der Länge von Teilfolgen sind nur ganzzahlige Werte möglich. Zur Optimierung der Intervalllänge für die Aggregation von Induktionsschleifendaten existieren einige Studien. Allerdings beziehen sich die Autoren im Wesentlichen auf die optimale Intervalllänge für die Archivierung [81], die Vorhersage einzelner Verkehrsgrößen wie dem Verkehrsfluss oder der Geschwindigkeit [78] oder der Vorhersage der zu erwartenden Reisezeit [79]. Es sind keine ähnlichen Arbeiten für die kurzfristige Vorhersage von Transitionen bzw. Verkehrsphasen bekannt.

In Arbeiten, die sich mit der Vorhersage einzelner Verkehrsgrößen beschäftigen, werden für die Intervalllänge zur Zusammenfassung der Daten meist Werte zwischen fünf und zwanzig Minuten verwendet [16, 65, 89, 103, 104]. Auch in dieser Arbeit werden deshalb Teilfolgen mit einer Länge aus diesem Intervall gebildet.

Ein weiterer wichtiger Parameter wird verwendet, um festzulegen wie hoch die Varianz einer Teilfolge sein muss, damit sie für die Motifsuche betrachtet wird. Anstatt einen absoluten Schwellwert zu bestimmen verwendet das hier vorgestellte Verfahren  $\phi_{var}$ -Quantile. Der gewählte Parameter  $\phi_{var}$  legt also fest, welcher Anteil des Datenstroms nach Motifs durchsucht wird. Es gilt: je höher  $\phi_{var}$ , desto weniger Teilfolgen werden betrachtet. Der Grund hierfür ist, dass nur Teilfolgen als interessant markiert werden, wenn ihre Varianz zu den  $\phi_{var}\%$  der höchsten Werte zählt. Mögliche Werte für  $\phi_{var}$  sind reellwertig und entstammen dem Intervall  $[0; 1]$ . Der Bereich kann jedoch weiter eingegrenzt werden, wenn der Grundgedanke dieser Diplomarbeit bedacht wird: es sollen Phasenübergänge entdeckt werden. Aus Abschnitt 2.3.3 ist bekannt, dass Phasenübergänge kurze und seltene Ereignisse sind, bei denen die Werte der Zeitreihen stark schwanken. Dadurch können Werte für  $\phi_{var}$  unter 0,5 ausgeschlossen werden. Für die weitere Arbeit wird  $\phi_{var}$  deshalb auf 0,75 gesetzt.

Tabelle 4.1: Übersicht der Parameter dieses Kapitels.

Schritt	Parameter	Beschreibung	Wahl
Motif-Vorverarbeitung	$h_{\varepsilon}$	Horizont für den Mittelwert zur Reduzierung des Rauschens.	4
	$l_m$	Länge der Motifs (pro Attribut zu setzen).	$l_m = 12$
	$\phi_{var}$	Schwellwert für die Varianz von interessanten Teilfolgen.	0.75
Suche nach Übereinstimmungen	$R$	Maximale Distanz zwischen Nachbarn.	In Abhängigkeit von $\phi_R = 0.1$ .
	$w$	Fensterbreite für die Anzahl an Teilfolgen die zur Suche nach Übereinstimmungen betrachtet werden sollen.	$w = 30$
Motif-Extraktion	$k$	Anzahl an Motifs pro Attribut.	$k = 75$

Für jede Teilfolge im Fenster wird die beste Übereinstimmung (d.h. die Teilfolge mit der geringsten euklidischen Distanz) bestimmt. Der Schwellwert  $R$  für die maxi-

male Distanz zwischen Teilfolgen einer Übereinstimmung trägt also nur bedingt zur Güte der gefundenen Motifs bei. Durch die Wahl von  $R$  wird jedoch die Laufzeit des Algorithmus beeinflusst: mit steigendem  $R$  wird die Kandidatenmenge für die Nachbarn einer Teilfolge größer. Die Optimierung von  $R$  spielt somit insbesondere für die inkrementelle Extraktion von Motifs auf Datenströmen in Echtzeit eine Rolle. Um einen sinnvollen Radius für die Nachbarschaft von Teilfolgen zu finden, werden auch an dieser Stelle Quantile bestimmt. Für die Wahl von  $\phi_R$  gilt, dass mit sinkendem  $\phi_R$  auch der Radius der Nachbarschaft und somit die Zahl der Kandidaten sinkt.

Die Wahl der Fensterbreite  $w$  ist sowohl für die Laufzeit als auch für die Güte der gefundenen Motifs wichtig. Durch die Fensterbreite wird eine untere Schranke des am Stück betrachteten Zeitraums spezifiziert – Teilfolgen im Fenster decken mindestens den Zeitraum  $[t_{\text{aktuell}} - (w - 1); t_{\text{aktuell}}]$  ab. Dieser Grenzfall ist jedoch für realistische  $w$  und  $\phi_{\text{var}}$  unwahrscheinlich, da in diesem Zeitraum keine Teilfolge verworfen werden dürfte.

Darüber hinaus legt  $w$  fest, mit wie vielen Teilfolgen jede einzelne Teilfolge verglichen wird. Die Anzahl beträgt  $2 \cdot (w - 1)$  (siehe Abschnitt 4.2.3).

## Repräsentation des Datenstroms

Wie Schaubild 5.1 zeigt, ist das Resultat der Arbeit bis zu diesem Punkt eine Menge aus den Trainingsdaten extrahierter Motifs. Diese Motifs entstammen besonders dynamischen Abschnitten des Datenstroms. Sie sollen im Folgenden verwendet werden, um eine Repräsentation des gesamten Datenstroms zu finden. Eine Anforderung an solch eine Repräsentation ist, dass sie sich als Eingabe für eine Vielzahl von Lernverfahren eignet. Das Lernen eines Modells durch unterschiedliche Lernverfahren soll ohne eine weitere Transformation auf den selben Ausgangsdaten möglich sein.

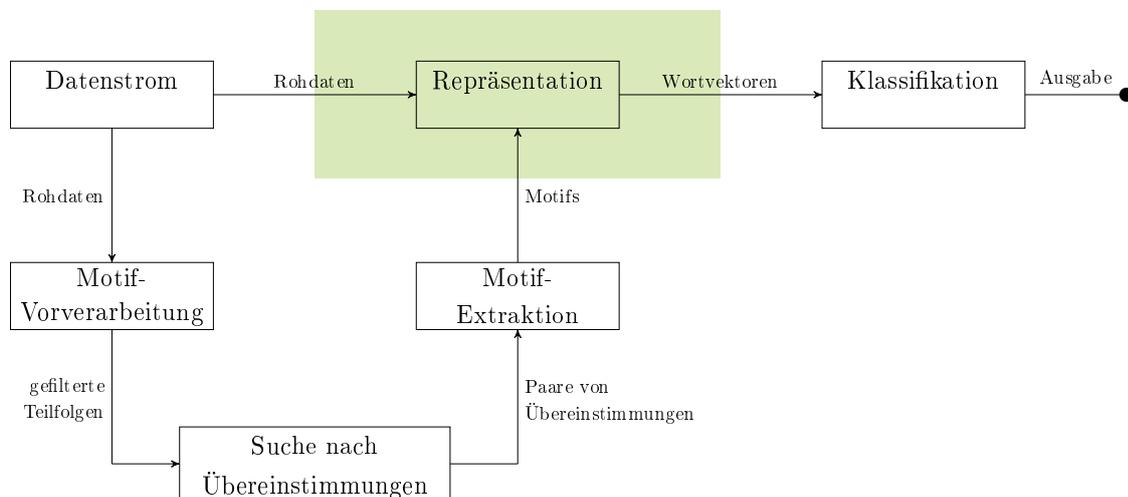


Abbildung 5.1: Die im letzten Kapitel gewonnenen Motifs werden in diesem Kapitel verwendet, um für die Experimente aus Kapitel 6 Datensätze zu generieren.

Darüber hinaus ist es wünschenswert, eine erweiterbare Repräsentation zu finden. Besonders wichtig erweist sich diese Anforderung, wenn zu einem späteren Zeitpunkt zusätzliche Informationen wie Wetter- oder Floating-Car-Daten einfließen sollen, die zur Zeit noch nicht verfügbar sind. Diese zusätzlichen Informationen können unterschiedlichster Natur sein und es ist nicht zu garantieren, dass auch hier Motifs verwendet werden können.

Besonders im Hinblick auf den zuletzt genannten Punkt ist zu beachten, dass es i. d. R. generell möglich ist nominale Werte bijektiv auf numerische Werte aus  $\mathbb{N}$  abzubilden. Der umgekehrte Fall einer allgemeinen bijektiven Abbildung numerischer Werte (wie z. B. kontinuierlicher Werte aus  $\mathbb{R}$ ) in endliche nominale Mengen ist im Allgemeinen nicht gewährleistet, da numerische Werte häufig nicht abzählbar endlich

sind. Wenn ein Intervall mehr als nur einen diskreten Wert umfasst, ist der nominale Wert für dieses Intervall immer weniger detailliert als der numerische. Deshalb verwendet die hier gewählte Repräsentation Werte aus  $\mathbb{R}$  für sämtliche Attribute.

## 5.1 Wortvektor-Analogie

Die Grundlage der entwickelten Repräsentation entstammt dem *Text Mining*. In dieser Disziplin des maschinellen Lernens werden *Dokumente*, bestehend aus aneinander gereihten *Wörtern* untersucht. Mit dieser Datenmenge als Ausgangsbasis werden diverse Lernaufgaben betrachtet [39]:

- Information Retrieval,
- Klassifizierung von Dokumenten,
- Clustering und Organisation von Dokumenten.

Üblicherweise werden Dokumente während der Datenvorverarbeitung in *Wortvektoren* umgewandelt. Es handelt sich hierbei um Vektoren mit reellwertigen Elementen, die eine Kompression des eigentlichen Dokuments darstellen. Neben der Verringerung des Datenvolumens werden unterschiedliche Dokumente mithilfe von Wortvektoren miteinander vergleichbar [100]. Darüber hinaus müssen natürlich auch keine Algorithmen speziell für Textdokumente entwickelt werden – es können prinzipiell alle Lernalgorithmen verwendet werden, die in anderen Bereichen des maschinellen Lernens zum Einsatz kommen und zur Verarbeitung von reellwertigen Vektoren geeignet sind.

Zunächst wird eine Menge von *Schlüsselwörtern* bestimmt, die in einem *Wörterbuch* abgelegt wird. Für jedes einzelne Dokument wird nun geprüft welche Schlüsselwörter mit welcher Häufigkeit vorhanden sind [106]. Abhängig von dieser Häufigkeit wird durch eine geeignete Funktion ein Gewicht für jedes Schlüsselwort errechnet und als Element im Wortvektor abgespeichert. Auch wenn die Einträge je nach Gewichtsfunktion binär, ganzzahlig oder reellwertig sein können, ist ein Eintrag des Vektors immer dann null, wenn das jeweilige Schlüsselwort gar nicht in einem Dokument vorkommt [100].

Im vorliegenden Fall werden Motifs auf eine ähnliche Art und Weise verwendet, um die ursprünglichen Zeitreihen durch eine vektorielle Darstellung zu repräsentieren. Die einzelnen Motifs übernehmen in der verwendeten Repräsentation die Rolle der Schlüsselwörter und bilden das Wörterbuch. Die Zeitreihen aller Verkehrsgrößen werden durch ein Fenster mit Länge  $l_m$  betrachtet. Die Teilfolge, die von einem Fenster abgedeckt wird, hat also die selbe Länge wie die entsprechenden Motifs einer Verkehrsgröße. Alle Teilfolgen der letzten  $\Delta t_w$  Zeitpunkte (siehe Abbildung 1.1) bilden ein Dokument. Der Datenstrom kann jetzt also als eine Folge von Dokumenten interpretiert werden.

Beachtet werden muss an dieser Stelle jedoch, dass die Entscheidung, ob ein Motif in einem Fenster vorhanden ist oder nicht, weniger eindeutig ist, als es im Text Mining der Fall ist. Wenn Schreibfehler, Pluralformen und Konjugationen nicht beachtet werden, kann im Text Mining eindeutig entschieden werden, ob ein Schlüsselwort in einem Dokument vorhanden ist oder nicht [100] – es wird für jedes Wort geprüft, ob die Buchstabenfolge identisch mit der eines Schlüsselworts ist. Da Motifs jedoch per Definition Teilfolgen einer Zeitreihe sind, die entweder exakt oder in ähnlicher Form auftreten, ist ein Vergleich auf absolute Übereinstimmung nicht ausreichend. Es sind zwei Möglichkeiten denkbar um zu entscheiden, ob ein Motif in einem Fenster vorliegt. Ein Motif liegt vor, wenn es:

- einen Schwellwert  $sim_{min}$  für die Ähnlichkeit zu einer Teilfolge nicht überschreitet,
- zu den ähnlichsten  $m$  Motifs einer Teilfolge gehört.

Bei der hier verwendeten Implementierung werden beide Möglichkeiten kombiniert. Es werden maximal  $m$  Motifs mit einer Ähnlichkeit von mindestens  $sim_{min}$  ausgewählt. In Abbildung 5.2 ist die Erstellung des Wortvektors  $\omega(C)$  unter Berücksichtigung der  $m = 1$  ähnlichsten Motifs und einer Gewichtsfunktion dargestellt, welche die Häufigkeit des Motifs im Fenster verwendet.

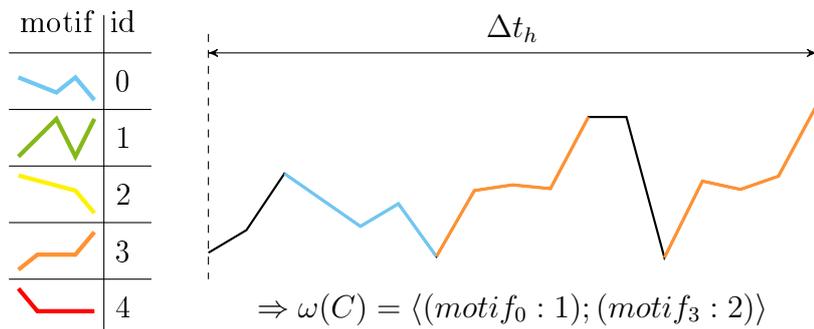


Abbildung 5.2: Beispiel für die Erzeugung eines Wortvektors  $\omega$  für eine Teilfolge  $C$  bei Verwendung von fünf Motifs.

Um zu verstehen, weshalb eine Repräsentation Wortvektoren vorteilhaft ist, betrachten wir erneut das Text Mining. Auch wenn ein Wörterbuch eine große Anzahl an Schlüsselwörtern enthält, sind dies meist nur aussagekräftige Wörter. Das bedeutet, Schlüsselwörter sind häufig genug, um in unterschiedlichen Dokumenten vorzukommen, jedoch selten genug, um Klassen von Dokumenten voneinander unterscheiden zu können. Dadurch sind in den Wortvektoren der einzelnen Dokumente die meisten Einträge mit null belegt, da nur wenige Schlüsselwörter tatsächlich im betrachteten Dokument auftreten [100]. Ein Wortvektor ist in der Regel also ein *spärlich besetzter Vektor*. Im Wesentlichen hat dies zwei Vorteile:

- zu Übertragungszwecken kann eine besonders hohe Komprimierungsrate erreicht werden, wenn Kommunikationskosten hoch sind (z.B. der Energieverbrauch des Funkmoduls eines Sensors [40]),
- es kann Rechenzeit bei der Erstellung des Modells eingespart werden, wenn spezielle Methoden zur Verarbeitung von spärlich besetzten Vektoren und Matrizen eingesetzt werden [93].

Die Erzeugung spärlich besetzter Vektoren ist deshalb auch hier das Ziel, da in Abhängigkeit vom gewählten  $k$  für die Anzahl an Motifs pro Attribut der Vektorraum sehr groß werden kann. Werden z.B. wie hier 75 Motifs pro Attribut bei fünf Attributen verwendet, hat der resultierende Vektorraum bereits  $5 \cdot 75 = 375$  Dimensionen.

Im nächsten Abschnitt wird beschrieben, wie die Gewichte der einzelnen Motifs für ein Fenster bestimmt werden.

### 5.2 TF\*IDF

Als Gewichtsfunktion wird das *TF\*IDF-Maß* verwendet. Wie bereits die Wortvektor-Repräsentation selbst, entstammt auch TF\*IDF dem Text Mining. Diese Gewichtsfunktion hat den Anspruch, Wörter stärker zu gewichten, die in einzelnen Dokumenten häufig, in der Menge aller Dokumente jedoch selten sind. Die grundsätzliche Idee besagt, dass Wörter, die in wenigen Dokumenten vorkommen besser zur Klassifikation geeignet sind. Jedoch bieten Wörter die insgesamt kaum auftreten wenig Potential um unbekannte Dokumente zu klassifizieren. Mit TF\*IDF wird also versucht, sowohl die Sensitivität als auch die Genauigkeit des Klassifikationsmodells zu optimieren. Hierfür verwendet TF\*IDF zwei Komponenten [100]:

- die *Vorkommenshäufigkeit* (englisch: *term-frequency*) gibt die Häufigkeit eines Worts in einem Dokument wieder,
- die *inverse Dokumenthäufigkeit* (englisch: *inverse-document-frequency*) entspricht der Bedeutung eines Worts in Relation zur gesamten Menge von Dokumenten.

#### DEFINITION 5.1: **Vorkommenshäufigkeit** ( $tf_i(j)$ )

Für ein gegebenes Dokument  $d_i$  und ein Wort  $j$  ist die **Vorkommenshäufigkeit**  $tf_i(j)$  die Anzahl der Vorkommen von  $j$  in  $d_i$ .

Um für die Klassifizierung von Dokumenten charakteristische Wörter von sehr allgemeinen Wörtern zu unterscheiden, wird mit der inversen Dokumenthäufigkeit  $idf(j)$  ein Skalierungsfaktor für jedes Wort  $j$  eingeführt. Zur Bestimmung von  $idf(j)$  wird zunächst die *Dokumenthäufigkeit* berechnet [106]:

**DEFINITION 5.2: Dokumenthäufigkeit ( $df(j)$ )**

Für eine gegebene Menge von Dokumenten  $D$  und ein Wort  $j$  ist die **Dokumenthäufigkeit**  $df(j)$  die Anzahl aller Dokumente in  $D$ , die  $j$  mindestens einmal enthalten.

Da die Dokumenthäufigkeit selbst Wörter stärker gewichtet, die in vielen Dokumenten vorkommen, wird sie mit (5.1) in die inverse Dokumenthäufigkeit umgewandelt [106]:

$$idf(j) = \log \frac{|D|}{df(j)}. \quad (5.1)$$

Das TF\*IDF-Maß eines Wortes  $j$  für das Dokument  $i$  ist dann schlicht das Produkt aus Vorkommenshäufigkeit und inverser Dokumenthäufigkeit [100]:

$$\begin{aligned} tf_i idf(j) &= tf_i(j) \cdot idf(j) \\ &= tf_i(j) \log \frac{|D|}{df(j)}. \end{aligned} \quad (5.2)$$

Bei näherer Betrachtung von (5.2) fällt auf, dass auch bei Verwendung des TF\*IDF-Maß jedes Gewicht einen Wert größer oder gleich null besitzt. Die Vorkommenshäufigkeit kann per Definition keine negativen Werte annehmen und auch die inverse Dokumenthäufigkeit muss positiv sein, da  $\frac{|D|}{df(j)}$  immer größer oder gleich eins ist.

Wie im vorhergehenden Abschnitt beschrieben, bilden alle Teilfolgen eines Zeitintervalls  $[t - \Delta t_w; t]$  ein Dokument. Die Vorkommenshäufigkeit eines Motifs ist somit die Anzahl der Teilfolgen eines Zeitintervalls, die dem Motif ähnlich sind. Analog ist die Dokumenthäufigkeit eines Motifs die Anzahl aller Zeitintervalle  $[t_i - \Delta t_w; t_i]$  in denen das Motif mindestens einmal vorgekommen ist. Mit der inversen Dokumentenhäufigkeit aus (5.1) kann dann auch für Motifs (5.2) zur Bestimmung des TF\*IDF-Maß verwendet werden.

### 5.2.1 Parameter für die Wortvektor-Repräsentation

Im Gegensatz zur Extraktion von Motifs in Kapitel 4 ist die Anzahl der Parameter zur Erstellung von Wortvektoren deutlich kleiner. Ein Teil der Parameter ist auch implizit durch die Parametrisierung in vorherigen Stufen gegeben.

Der erste zu betrachtende Parameter ist die Wortlänge  $l_w$ . Anders als beim Text Mining ist dieser Parameter nicht direkt durch die Daten gegeben. Im Grunde sind Wörter beliebiger Länge möglich, da diese durch die Anzahl der Elemente pro Teilfolge bestimmt ist. Damit die im Vorfeld extrahierten Motifs auch verwendet werden können, muss die Wortlänge  $l_w$  der Länge von Motifs  $l_m$  entsprechen.

Ähnlich ist die Anzahl von Wörtern pro Dokument  $\#w$  für den betrachteten Fall der Zeitreihen festzulegen. Da die Wortlänge  $l_m$  konstant ist und für die Vorhersage die letzten  $\Delta t_w$  Zeitpunkte betrachtet werden ist auch  $\#w$  implizit gegeben. Es wurde

in Kapitel 4 erläutert, dass die betrachteten  $\Delta t_w$  Elemente eines Fensters in nicht disjunkte Teilfolgen aufgeteilt werden – die Teilfolge  $C_i$  beinhaltet dabei die Elemente  $\{c_i, \dots, c_{i+(\Delta t_w-1)}\}$ , die nächste Teilfolge  $C_{i+1}$  analog die Elemente  $\{c_{i+1}, \dots, c_{i+\Delta t_w}\}$ . Somit beträgt die Anzahl der Wörter pro Dokument  $[(\Delta t_w + 1) - l_w]$ .

Um zu entscheiden, welche Elemente des Wortvektors von null verschieden sind, werden noch die Parameter  $sim_{min}$  für die minimale Ähnlichkeit und  $m$  für die maximale Anzahl von ähnlichen Motifs pro Teilfolge benötigt. Beide Parameter können kombiniert werden oder aber einzeln als Entscheidungsmerkmal verwendet werden. Erst genannter Parameter *aktiviert* ein Motif  $m_j$  als gültige Repräsentation für eine Teilfolge  $C_i$ , wenn gilt  $sim(m_j, C_i) \geq sim_{min}$ . Der Schwellwert wird erneut mithilfe von Quantilen bestimmt. Es werden sämtliche Abweichungen zwischen Teilfolgen und extrahierten Motifs bestimmt. Anschließend wird der Wert des 0.1-Quantils für  $sim_{min}$  verwendet. Generell gilt, dass der erzeugte Wortvektor abhängig vom gewählten Parameter  $sim_{min}$  sowie der extrahierten Motifs viele von null verschiedene Einträge enthalten kann.

Anders ist dies, wenn nur die  $m$  ähnlichsten Motifs aktiviert werden. Die Anzahl der Elemente des Wortvektors die nicht null sind beträgt dann höchstens

$$[(\Delta t_w + 1) - l_w] \cdot m. \quad (5.3)$$

In diesem Fall können spärlich besetzte Vektoren sogar garantiert werden. In den folgenden Experimenten werden  $m$  und  $sim_{min}$  kombiniert verwendet.

Eine Übersicht der in diesem Kapitel relevanten Parameter bietet Tabelle 5.1. Sie enthält sowohl eine Erläuterung der einzelnen Parameter als auch ihre konkrete Wahl für die Experimente in Kapitel 6.

Tabelle 5.1: Übersicht der Parameter dieses Kapitels.

Parameter	Beschreibung	Wahl
$l_w$	Legt die Wortlänge fest. Ist eine konstante Größe und entspricht der Wahl der Länge von Motifs $l_m$ .	$l_w = l_m = 12$
$\#w$	Beschreibt die Anzahl von Wörtern pro Dokument. Implizit durch $\Delta t_w$ gegeben.	durch $\Delta t_w$ bestimmt
$sim_{min}$	Schwellwert für die minimale Ähnlichkeit zwischen Motifs und Teilfolgen zur Entscheidung ob ein Vorkommen vorliegt.	Attribut-abhängig
$m$	Maximale Anzahl an vorhandenen Motifs pro Teilfolge.	1

*Parameter!Repräsentation!)*

## Experimente

In den Kapiteln 3 bis 5 dieser Arbeit wurde eigener Java-Quelltext verwendet, um die vorgestellte Repräsentation zu erzeugen und die Datensätze für die Experimente in diesem Kapitel vorzubereiten. Der Quelltext wurde im Kontext des *streams-Frameworks*<sup>1</sup> des Lehrstuhls für Künstliche Intelligenz der TU Dortmund ausgeführt. Um für die bevorstehenden Experimente auf eine Vielzahl von Klassifizierungs- und Optimierungsverfahren zurückgreifen zu können, wird für den Rest der Arbeit RapidMiner eingesetzt.

Das Open Source Programm RapidMiner wird in der Version 5.2 verwendet und besitzt zum Zeitpunkt an dem diese Arbeit erstellt wird keine *Operatoren* zur Verarbeitung ungebundener Datenströme. Deshalb werden Textdateien im *Sparse-Format* des *LibSVM*-Pakets<sup>2</sup> mit dem *streams-Framework* erzeugt, die einem Ausschnitt des Datenstroms entsprechen und als Eingabemenge für RapidMiner-Prozesse dienen.

### 6.1 Datensätze

Für die nachfolgenden Experimente werden i. d. R. die ersten vier Monate (Juli bis Oktober 2010) verwendet. Ausnahmen sind hier die Experimente in Abschnitt 6.7, welche die restlichen Daten (November und Dezember 2010) zur Evaluation verwenden.

Es stellt sich die Frage, welche Zählschleifen repräsentative Werte liefern. Nicht jeder Messquerschnitt liefert ähnliche Daten, da z.B. die gesetzlichen Vorgaben für die Höchstgeschwindigkeit variieren und auch die geografische Lage – insbesondere die Nähe zu Städten mit einer hohen Einwohnerzahl – das Verkehrsaufkommen beeinflusst. Deshalb werden die Zählschleifen zunächst durch ein Clustering mit dem *k-Means* Operator in RapidMiner gruppiert. Dabei wird berücksichtigt, dass die Werte aus wenig befahrenen Zeiträumen viele Ausreißer beinhalten. Der Anteil von synchronem Fluss und Stau ist während der Nacht z. B. weiterhin relativ hoch, da wenige langsam fahrende Fahrzeuge mit dem Schema aus Abbildung 3.2 zur Phase synchroner Fluss oder Stau führen.

In Abbildung 6.1 ist die über alle Messquerschnitte aggregierte Tagesganglinie für die Fahrzeugzahl dargestellt. Die Anzahl aller Fahrzeuge  $j\_ges$  und die Anzahl LKW  $j\_lkw$  sind hier in einem Histogramm aufsummiert. Gegen 03:30 Uhr (Minute 210) und 20:20 Uhr (Minute 1220) treten das erste bzw. letzte Maximum des Tages auf.

<sup>1</sup><http://www.jwall.org/streams/>

<sup>2</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

## 6 Experimente

Im Bereich zwischen diesen beiden Maxima ist die Anzahl an Fahrzeugen deutlich höher als vor bzw. nach diesem Intervall. Für LKW ist der Verlauf deutlich flacher und die Kurve beginnt bereits gegen 15 Uhr (900 Minuten) abzufallen.

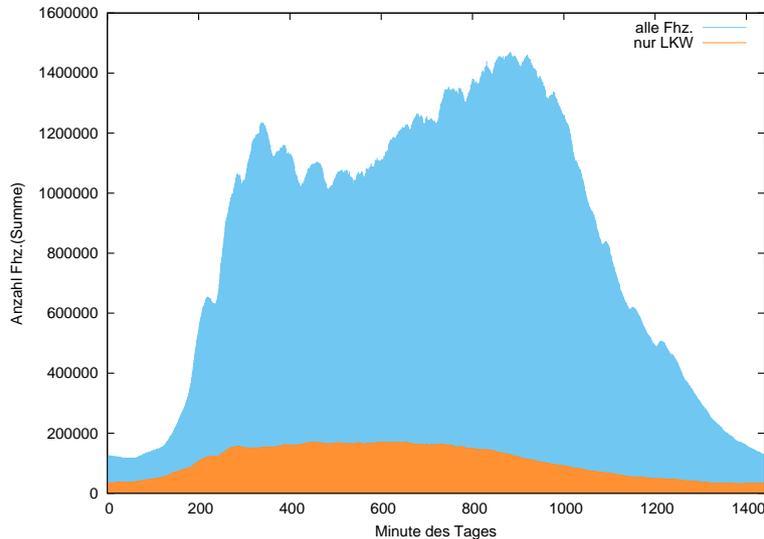


Abbildung 6.1: Das Histogramm stellt die aggregierte Tagesganglinie für alle 765 Zählschleifen des Sensornetzwerks für die Monate Juli bis Oktober 2010 dar. Die Höhe der einzelnen Balken bezeichnet die akkumulierte Anzahl aller Fahrzeuge  $j\_ges$  (blau) und LKW  $j\_pkw$  (orange) für die entsprechende Minute des Tages.

Werden das 0,05– und 0,95–Quantil für die Anzahl aller Fahrzeuge  $j\_ges$  betrachtet, bestätigt sich dieser Zeitraum: 90% der Fahrzeuge werden von Zählschleifen zwischen 03:56 Uhr und 20:18 Uhr detektiert. Für die Vektorrepräsentation von Sensoren als Eingabe für das Clustering wird also dieser Zeitraum betrachtet. Insgesamt werden vier Größen verwendet.

Der Anteil defekter Sensoren im betrachteten Zeitraum beträgt ca. 20%. Es ist zu erwarten, dass sich die Ausfallzeiten nicht gleichmäßig auf alle 765 Zählschleifen verteilen. Vielmehr werden Sensoren existieren, die z. B. aufgrund baulicher Maßnahmen dauerhaft keine Werte liefern. Um diesen Aspekt zu berücksichtigen, wird als erste Größe zur Charakterisierung von Zählschleifen der relative Anteil verwendet, den ein Sensor defekt war und keine Daten lieferte.

Durch unterschiedliche gesetzliche Vorgaben streuen die Werte für die Höchstgeschwindigkeiten an den Messquerschnitten. Als zweite Größe wird deshalb das 0,95–Quantil der PKW-Geschwindigkeiten  $v\_pkw$  als Schätzer für die Höchstgeschwindigkeit verwendet.

Auch die Auslastung von Autobahnabschnitten ist sehr unterschiedlich. Weil Phasenübergänge mit hohen Verkehrsdichten einhergehen, ist es sicherlich sinnvoll, stark ausgelastete von wenig befahrenen Zählschleifen zu trennen. Hierfür wird der relative Anteil bestimmt, den eine Zählschleife eine gestaute Verkehrsphase (synchroner Fluss, Stau) misst.

Die letzte für die Aufteilung verwendete Größe soll über den kompletten Zeitraum befahrene Zählschleifen von nur zu Stoßzeiten befahrenen Zählschleifen trennen. Zu diesem Zweck wurde der relative Anteil berechnet, den ein Sensor mindestens ein Fahrzeug pro Minute erfasst.

Da die Wertebereiche dieser vier Größen unterschiedlich sind und ein Clustering-Algorithmus eine Metrik zur Bestimmung der Ähnlichkeit verwendet, wäre ein Clustering mit diesen Vektoren durch den übermäßig großen Wert für die Geschwindigkeit bestimmt. Vor dem Clustering werden die Werte aller vier Größen also standardisiert. Da im Vorfeld nicht bekannt ist, wie viele Gruppen von Sensoren existieren, wird das Clustering für alle Werte  $k \in [2; 50]$  durchgeführt. Für jedes Clustering wird die *within cluster distance (wcd)* bestimmt. Es handelt sich hierbei um ein Gütemaß für die „Reinheit“ der gefundenen Cluster. Die Distanz aller Elemente eines Clusters zum Zentrum des Clusters wird bestimmt und gemittelt. Wird ein aus zwei Gruppen bestehendes Cluster getrennt, sinkt die within cluster distance stark. Wird dagegen ein Cluster aufgeteilt, dass nur eine Gruppen von Elementen enthält, tritt zwar ein weiterer Rückgang der within cluster distance ein, jedoch ist der Sprung deutlich kleiner als im ersten Fall. Zur Bestimmung der optimalen Zahl von Clustern wird dieses Verhalten der within cluster distance ausgenutzt. Wie in Abbildung 6.2 durchgeführt, wird die within cluster distance gegenüber  $k$  in einem Diagramm aufgetragen. Nun muss ein „Knick“ der Kurve gefunden werden, so dass

$$wcd_{i-1} - wcd_i \gg wcd_i - wcd_{i+1} \quad (6.1)$$

gilt [41]. In Abbildung 6.2 ist dies für  $k = 6$  der Fall. Es werden mit der oben beschriebenen Charakterisierung von Zählschleifen sechs unterschiedliche Gruppen identifiziert.

Die Clusterzentren, d. h. die Mittelwerte für die sechs Gruppen von Zählschleifen, sind in Tabelle 6.1 aufgelistet. Zusätzlich zu den für das Clustering verwendeten Größen sind die Anzahl an Sensoren pro Cluster, der Anteil an LKW sowie die durchschnittliche Geschwindigkeit für die Gruppe von Zählschleifen aufgeführt.

Die Cluster 1 und 2 liefern kaum Daten, da Zählschleifen in diesen beiden Gruppen sehr selten befahren bzw. häufig defekt sind. Auch Cluster 3 wird im Folgenden nicht weiter berücksichtigt, weil die niedrigen Geschwindigkeiten auf Geschwindigkeitsbegrenzungen von 60 oder 80  $\frac{km}{h}$  hinweisen und das hier verwendete Schema zur Bestimmung der Verkehrsphase aus Abschnitt 3.5 für diesen Fall nicht geeignet ist. Der Verkehr auf solchen Autobahnabschnitten wäre niemals im freien Fluss, da eine Geschwindigkeit unter 81  $\frac{km}{h}$  als synchroner Fluss eingestuft wird.

Die übrigen drei Cluster unterscheiden sich am deutlichsten bezüglich ihrer Auslastung, dem Anteil an LKW und der Höchstgeschwindigkeit. Zählschleifen aus Cluster

## 6 Experimente

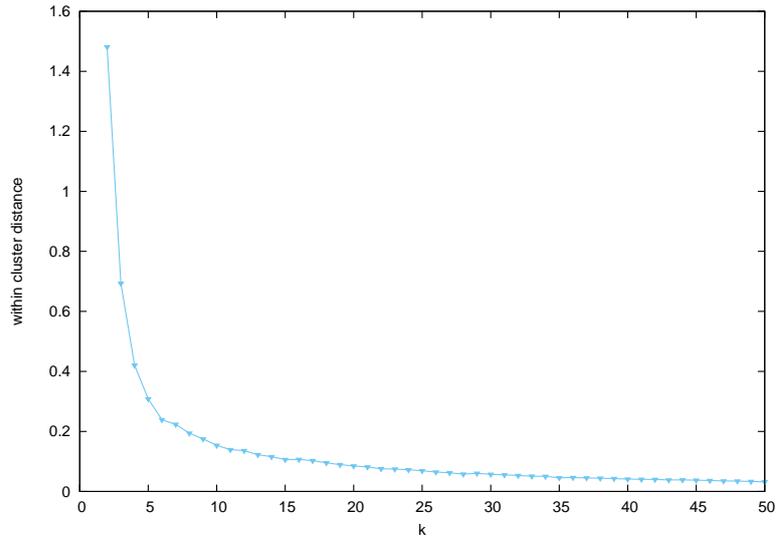


Abbildung 6.2: Die Zählschleifen können mit bei der hier gewählten Vektorrepräsentation in sechs Cluster aufgeteilt werden. Bei einer weiteren Aufteilung nimmt die within cluster distance nur noch geringfügig ab.

Tabelle 6.1: Die Werte der Clusterzentren für eine Aufteilung des Sensornetzwerks in sechs Cluster.

id	#(zs)	Anteil defekt [%]	Anteil gestaut [%]	Anteil belegt [%]	Anteil LKW [%]	$\phi_{0,95}(v)$ $\frac{km}{h}$	$\bar{v}$ $\frac{km}{h}$
1	68	1,36	0,01	0,01	41,94	1	0
2	12	95,05	65,96	41,80	12,37	20	5
3	47	14,75	38,23	53,67	16,92	84	68
4	63	68,71	3,34	33,69	12,93	135	102
5	263	8,27	1,02	17,47	2,47	145	114
6	312	7,46	3,79	52,89	19,75	116	98

6 sind vergleichsweise stark ausgelastet und detektieren einen hohen Anteil an LKW. Im Mittel sind 3,79% aller Messungen den Verkehrsphasen synchrones Fluss und Stau zuzuordnen. Mit 312 Sensoren gehören rund 41% des Sensornetzwerks diesem Cluster an.

Auch Cluster 5 deckt mit 263 Zählschleifen einen großen Teil des Netzwerks ab (34%). Es werden jedoch die weniger stark befahrenen Zählschleifen mit einem deutlich niedrigeren Anteil an LKW (2,47%) dem Cluster 5 zugeordnet. Die durchschnittliche Reisegeschwindigkeit für Streckenabschnitte dieses Clusters sind mit  $114 \frac{km}{h}$  vergleichsweise hoch.

Die Werte aus Cluster 4 befinden sich zwischen den Werten aus den Clustern 5 und 6. Der LKW-Anteil beträgt 12,95%. Die durchschnittliche Reisegeschwindigkeit sowie der Anteil der gestauten Phasen ähnelt den Werten für das Cluster 6. Die

Höchstgeschwindigkeit ist jedoch um einiges höher als in Cluster 6 und nähert sich dem Wert aus Cluster 5 an. Nur der Anteil der Ausfallzeiten von Sensoren liegt mit 68,71% deutlich über den Werten für die Cluster 5 und 6. Das Cluster 4 deckt mit 63 Zählschleifen rund 8% des Sensornetzwerks ab.

Die große Zahl von zugeordneten Zählschleifen sowie die hohe Auslastung in Cluster 6 sind der Grund, weshalb für die meisten folgenden Experimente Daten aus diesem Cluster verwendet werden. Für die Experimente in den Abschnitten 6.5 und 6.9 werden jedoch auch Zählschleifen aus den Clustern 4 und 5 berücksichtigt.

Für jedes der drei Cluster werden zwei Zählschleifen zufällig ausgewählt. In Abschnitt 6.5 werden die Daten dieser Zählschleifen verwendet, um Modelle zu erzeugen und die Vorhersagegüte in den einzelnen Clustern zu vergleichen.

- Das Cluster 4 wird durch die Zählschleifen 1758464770 (A42 zwischen Castrop-Rauxel und AK Castrop-Rauxel-Ost; im Folgenden *CaRa*) und 1762267937 (A40 zwischen Mülheim-Winkhausen und Mülheim-Dümpten; im Folgenden *MüWi*) repräsentiert.
- Aus Cluster 5 werden die Zählschleifen 1753482530 (A44 zwischen Witten-Stockum und Witten-Zentrum; im Folgenden *WiSt*) und 1863457570 (A3 zwischen Duisburg-Wedau und AK Breitscheid; im Folgenden *DuWe*) zur Erzeugung von Datensätzen verwendet.
- Für Cluster 6 stammen die Daten von den Zählschleifen 1409876001 (A40 zwischen Dortmund-West und Dortmund-Kley; im Folgenden *DoWe*) und 1759316001 (A42 zwischen Essen-Altenessen und Essen-Nord; im Folgenden *EsAl*).

Tabelle 6.2 enthält charakteristische Werte dieser sechs Zählschleifen.

Tabelle 6.2: Eigenschaften der in den folgenden Experimenten zur Datenerzeugung verwendeten Zählschleifen.

zs	Kürzel	Cluster	Anteil belegt [%]	Anteil gestaut [%]	Anteil LKW [%]	$\phi_{0,95}(v)$ $\frac{km}{h}$	$\bar{v}$ $\frac{km}{h}$
1409876001	DoWe	6	57,86	4,90	18,97	102	88
1753482530	WiSt	5	21,69	0,42	5,89	161	131
1758464770	CaRa	4	29,21	2,08	11,32	139	110
1759316001	EsAl	6	53,64	1,75	18,30	115	97
1762267937	MüWi	4	43,62	1,06	15,56	107	95
1863457570	DuWe	5	28,33	2,01	4,87	134	112

Alle in den anschließenden Experimenten verwendeten Datensätze sind in Tabelle 6.3 aufgeführt. Neben der Anzahl an Beispielen pro Datensatz sind auch die Anteile der sechs Klassen aufgeführt. Der Schlüssel eines Datensatzes setzt sich aus dem Kürzel für die Zählschleife, dem Cluster, dem Zeitraum aus dem die Daten stammen und dem Vorhersagehorizont zusammen. So stammen die Beispiele aus Datensatz *CaRa-4-0710-10* von der Zählschleife CaRa (1758464770) zwischen **Castrop-Rauxel** und AK Castrop-Rauxel-Ost aus Cluster 4. Sie umfassen Wortvektoren der Monate Juli bis Oktober (**0710**) und die Ausgaben der Beispiele beziehen sich auf einen Vorhersagehorizont von **10** Minuten.

Tabelle 6.3: Anzahl der Beispiele sowie die Klassenverteilung für die verwendeten Datensätze.

Datensatz	#Beispiele	Anteil $\mathbf{Y}$ [%]					
		<i>grün</i> $\rightarrow$ <i>gelb</i>	<i>grün</i> $\rightarrow$ <i>rot</i>	<i>gelb</i> $\rightarrow$ <i>rot</i>	<i>gelb</i> $\rightarrow$ <i>grün</i>	<i>rot</i> $\rightarrow$ <i>grün</i>	<i>rot</i> $\rightarrow$ <i>gelb</i>
CaRa-4-0710-10	6662	21,72	4,95	23,43	21,24	5,34	23,30
DoWe-6-0710-05	4332	45,82	0,46	3,33	46,49	0,37	3,51
DoWe-6-0710-10	4868	45,54	1,36	3,14	45,50	1,38	3,06
DoWe-6-0710-15	5232	45,44	2,08	2,50	45,35	1,68	2,92
DoWe-6-0710-20	5526	45,29	2,55	2,19	45,15	2,24	2,55
DoWe-6-0710-25	5749	45,47	2,92	1,90	44,72	2,59	2,38
DoWe-6-0710-30	6015	44,97	3,34	1,71	44,62	2,98	2,36
DoWe-6-0712-10	9374	46,27	0,94	3,23	45,50	0,82	3,23
DoWe-6-1112-10	4507	47,04	0,49	3,33	42,48	0,22	3,42
DuWe-5-0710-10	1357	41,93	1,55	8,33	38,54	1,40	8,18
EsAl-6-0710-10	2977	30,20	10,55	8,60	31,71	9,04	9,88
MüWi-4-0710-10	907	41,57	2,32	5,73	41,90	2,54	5,84
WiSt-5-0710-10	2439	28,95	4,76	27,55	16,24	4,80	17,67
ALLE-0710-10	19205	32,36	4,52	13,18	32,02	4,43	13,49

## 6.2 Aufbau des Experiments

Sofern nicht im Text auf einen anderen Aufbau hingewiesen wird, wurde der Basisprozess aus Abbildung 6.3 verwendet. Die spärlich besetzten Vektoren  $Z$  dienen als Eingabe für die Lernalgorithmen. Zur Optimierung der Parameter  $\alpha$  wurde entweder der `Optimize Parameters (Grid)` oder der `Optimize Parameter (Evolutionary, Parallel)` Operator in RapidMiner eingesetzt. Um die zu erwartende Klassifikationsgüte für unabhängige Trainingsdaten abzuschätzen, wurde mit dem `X-Validation (Parallel)` Operator eine fünffache Kreuzvalidierung durchgeführt. Für die Bestimmung der Klassifikationsgüte wurde die `accuracy` mit dem `Performance (Classification)` Operator bestimmt. Für jede Parametrisierung wurde die Klassifikationsgüte der Modelle  $\hat{f}_\alpha$  protokolliert.

## 6.3 Vergleich von Lernalgorithmen

In diesem Abschnitt werden zunächst eine Reihe von Lernalgorithmen zum Trainieren eines Modells verwendet. Um zu evaluieren welche Lernalgorithmen grundsätzlich für die Vorhersage von Zustandsübergängen in Verkehrssystemen geeignet sind, wird das Basisexperiment aus Abbildung 6.3 für alle Lernverfahren durchgeführt. Die Klassifikationsgüte jedes Modells wird protokolliert, um anschließend mit den Ergebnissen der anderen Modelle verglichen zu werden. Besonderes Augenmerk liegt hier auf den Phasenübergängen *grün*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *rot*. Diese Phasenübergänge sind für die Verkehrsbeeinflussung interessant, weil nur durch eine rechtzeitige Erkennung Maßnahmen ergriffen werden können, um die Wahrscheinlichkeit eines Verkehrszusammenbruchs sowie die Entstehung von Stauwellen durch eine Reduzierung der Verkehrsdichte zu verringern.

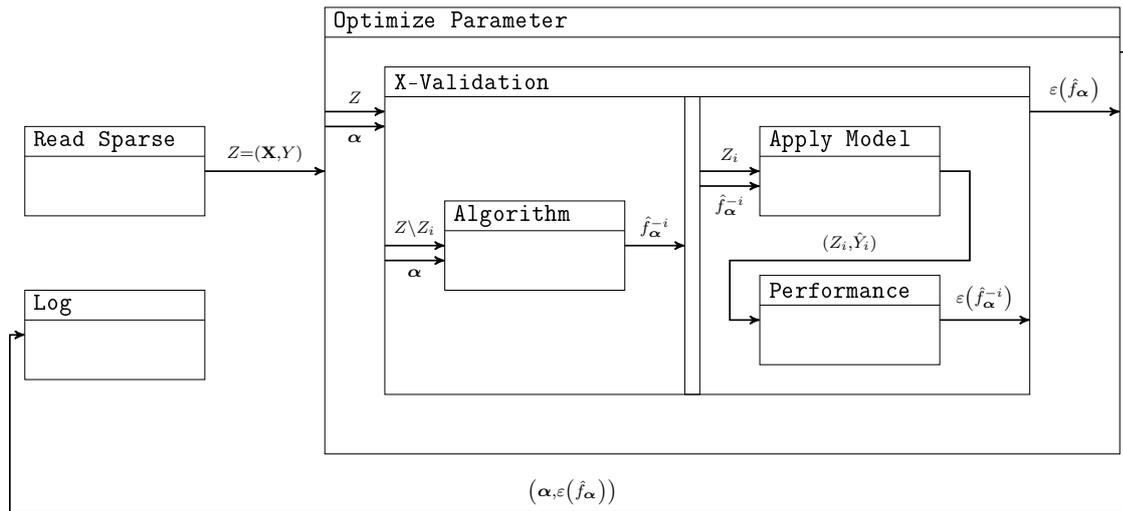


Abbildung 6.3: Das Schaubild stellt den grundsätzlichen Aufbau der Modellerzeugung dar.

### 6.3.1 Verwendete Lernalgorithmen

In diesem Kapitel werden fünf Algorithmen verwendet, um ein Klassifikationsmodell zu erstellen. Eine formale Einführung dieser Lernalgorithmen übersteigt den Rahmen dieser Arbeit. Deshalb sollen in diesem Abschnitt nur die Grundprinzipien der Klassifikation mit diesen Verfahren vorgestellt werden. Für den interessierten Leser werden darüber hinaus noch Quellen mit weiterführender Literatur genannt.

#### Naive Bayes

Die Idee des *Naive Bayes Klassifikators* beruht auf dem Bayestheorem. Es wird ein probabilistisches Modell aus den Trainingsdaten gewonnen. Die bedingten Wahrscheinlichkeiten der einzelnen Attribute, gegeben die Ausgaben der Trainingsbeispiele, werden als unabhängig angenommen [84]. Um ein Klassifikationsmodell zu erzeugen, werden die Häufigkeiten der Ausgaben bestimmt. Damit werden für alle Klassen  $y_i$  die Apriori-Wahrscheinlichkeiten  $P(y_i)$  für die Ausgabe  $y_i$  abgeschätzt. Darüber hinaus wird noch die bedingte Wahrscheinlichkeit  $P(x_j|y_i)$  eines Attributes  $x_j$ , gegeben Ausgabe  $y_i$  benötigt. Die Wahrscheinlichkeiten werden in der Trainingsphase durch die Bestimmung der Häufigkeiten in der Trainingsmenge fest gelegt. Der Naive Bayes Klassifikator ist dann durch (6.2) gegeben:

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \arg \max_i (P(y_i) \cdot P(\mathbf{x}|y_i)) \\ &= \arg \max_i \left( P(y_i) \cdot \prod_{j=0}^{p-1} P(x_j|y_i) \right). \end{aligned} \quad (6.2)$$

Weitere Information zu Modellierung mit dem Bayestheorem finden sich z. B. in [3] und [82].

**Parameter** Es wurde der in RapidMiner integrierte **Naive Bayes Operator** eingesetzt. Dieses Verfahren ist frei von Parametern, wodurch die Parameteroptimierung wegfällt.

### **$k$ nächste Nachbarn**

Bei  *$k$  nächste Nachbarn* handelt es sich um einen so genannten *trägen Lerner*. Die Generalisierung findet nicht während einer Trainingsphase durch Erzeugung eines Modells, sondern während der Klassifikation selbst statt. Für ein Beispiel  $\mathbf{x}$  werden die  $k$  nächsten Beispiele der Trainingsmenge ermittelt. Hierfür bedient sich der Algorithmus einer Metrik wie z.B. der euklidischen Distanz (siehe Abschnitt 2.1.3). Die Ausgabe  $y$  von  $\mathbf{x}$  ist dann die Hauptklasse der Nachbarschaft von  $\mathbf{x}$  [39]:

$$\hat{f}(\mathbf{x}) = \arg \max_i \sum_{\forall \mathbf{x}_j \in N(\mathbf{x})} (1 \cdot (y_i = y_j)). \quad (6.3)$$

In Abbildung 6.4 ist ein Beispiel für die Klassifikation mit  $k$  nächste Nachbarn für  $k = 3$  im  $\mathbb{R}^2$  zu sehen. Die entsprechenden Nachbarn der zu klassifizierenden Beispiele  $\mathbf{x}_1$  und  $\mathbf{x}_2$  sind als ausgefüllte Kreise kenntlich gemacht. Durch einen Mehrheitsentscheid über die Klassen der drei Nachbarn von  $\mathbf{x}_2$  wird dieses Beispiel der blauen Klasse zugeordnet. Für das Beispiel  $\mathbf{x}_1$  sind alle Nachbarn Beispiele der orangenen Klasse, weshalb auch  $\mathbf{x}_1$  dieser Klasse zugeordnet wird.

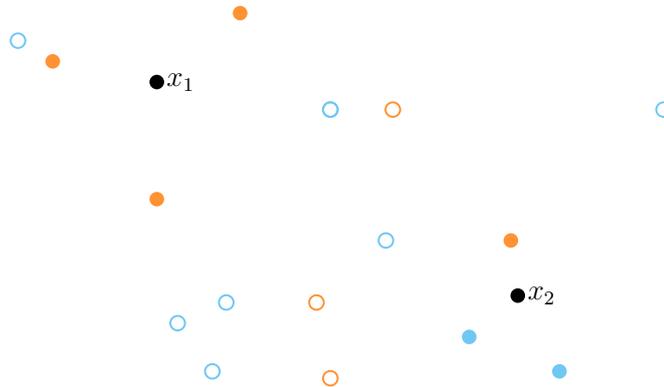


Abbildung 6.4: Beispiel für die Klassifikation durch  $k = 3$  nächste Nachbarn.

Das Verfahren der  $k$  nächsten Nachbarn ist ein vielseitig einsetzbarer Algorithmus und Informationen zu diesem Verfahren finden sich in vielen Standardwerken wie z.B. [39] und [41].

**Parameter** Zur Erstellung eines  $k$ -NN Modells wurde der in RapidMiner integrierte Operator **k-MN** verwendet. Die Wahl von  $k$  ist der bestimmender Faktor für die Klassifikationsgüte eines  $k$  nächste Nachbarn Modells. Darüber hinaus kann noch die Metrik für die Bestimmung der Distanzen ausgewählt werden. Hier sind jedoch keine großen Unterschiede aufgefallen, weshalb nachfolgende Ergebnisse nur für die

euklidische Distanz dokumentiert sind. Für die Optimierung von  $k$  bietet sich die Rasteroptimierung an, da lediglich positive, ganzzahlige Werte in Frage kommen. Abbildung 6.5 stellt die Klassifikationsgüte dem Parameter  $k$  gegenüber. Für das

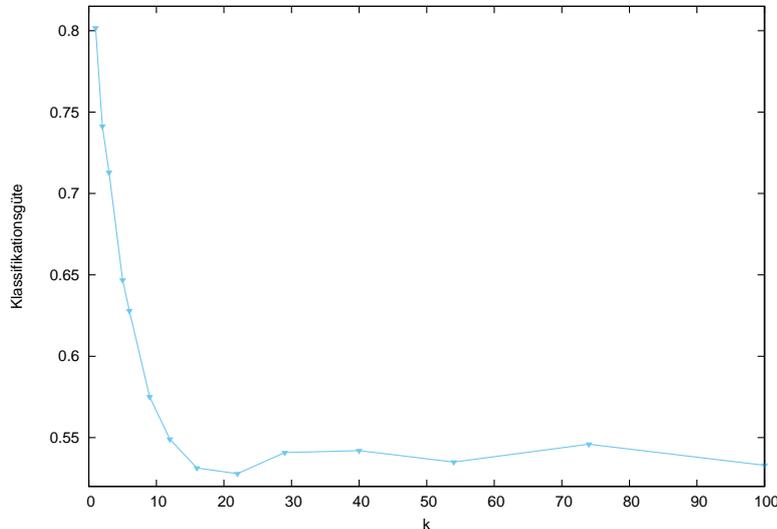


Abbildung 6.5: Die Klassifikationsgüte von  $k$  nächste Nachbarn in Abhängigkeit von  $k$ . Allgemein sinkt die Güte mit steigendem  $k$ . Ab einem  $k$  von zwölf schwankt die Güte um 54%.

vorliegende Problem ist  $k = 1$  optimal. Im Allgemeinen sinkt die Klassifikationsgüte mit steigendem  $k$  und die Klassifikationsgüte ist bereits für  $k = 3$  um rund 10% niedriger als für den optimalen Wert. Ab einem  $k$  von zwölf pendelt sich die Klassifikationsgüte bei ca. 54% ein.

## Random Forests

Für die Klassifikation mit *Random Forests* werden *Bäume* verwendet. Ein Baum besteht aus hierarchisch geordneten *Knoten*, an denen jeweils ein Attribut betrachtet wird, um zu entscheiden welcher Knoten als nächstes betrachtet wird. Für die Entscheidung, welches Attribut wann betrachtet wird, verwendet der Baumlerner in der Trainingsphase ein Gütemaß. Das Attribut, welches die Güte bzw. den Informationsgewinn des Baums maximiert wird für den nächsten Knoten verwendet. Um ein Beispiel zu klassifizieren, wird ein Ast des Baums bis zu einem *Blatt* verfolgt. Die zu wählenden Abzweigungen beim Abstieg zu einem Blatt werden durch die Ausprägungen der Attribute des Beispiels bestimmt. Jedes Blatt enthält eine Ausgabe  $y$ , die als  $\hat{y}$  für das Beispiel  $\mathbf{x}$  dient.

Durch die *Split-Bedingungen* an den Knoten wird der Beispierraum in Bereiche mit orthogonalen Begrenzungen aufgeteilt, die parallel zu den Koordinatenachsen liegen.

## 6 Experimente

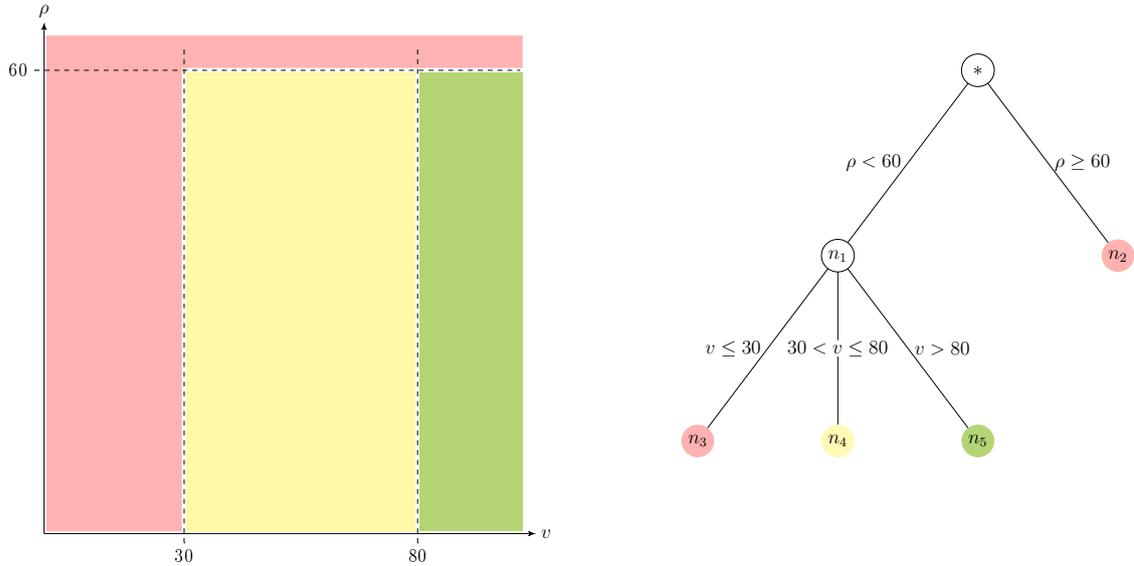


Abbildung 6.6: Beispiel für die Klassifikation mit einem Baum.

Im rechten Teil von Abbildung 6.6 ist ein Baum zu sehen, der sich aus dem Schema zur Bestimmung der Verkehrsphase aus Abschnitt 3.5 ergibt. In Abhängigkeit der Verkehrsdichte  $\rho$  und der Geschwindigkeit  $v$  wird beginnend bei der *Wurzel*  $*$  ein Weg zu einem Blatt für jedes Beispiel  $\mathbf{x}$  gefunden. Neben der Wurzel  $*$  ist lediglich  $n_1$  ein *innerer* Knoten. Die Knoten  $n_2$  bis  $n_5$  sind Blätter und beschreiben die Ausgaben  $\hat{y}$ . Die Baumstruktur ist äquivalent zur Aufteilung des Beispielraums im linken Teil der Abbildung.

Bäume besitzen generell einen niedrigen Bias und eine hohe Varianz [41]. Da sich der Fehler eines Modells auf den Bias und die Varianz reduzieren lässt, wird mit Random Forests versucht, die Varianz durch die Zusammenfassung vieler unkorrelierter Bäume zu einem Wald zu reduzieren [36]. Für  $I$  Bäume mit einer Varianz von  $\sigma_B^2$  und Korrelation  $\varrho$ , ist die Varianz des Waldes  $\sigma_W^2$  bestimmt durch [41]:

$$\sigma_W^2 = \varrho \cdot \sigma_B^2 + \frac{1 - \varrho}{I} \cdot \sigma_B^2. \quad (6.4)$$

Der rechte Term strebt dabei für eine steigende Zahl von Bäumen  $I$  gegen null. Um auch den linken Term zu minimieren, muss also die paarweise Korrelation  $\varrho$  der Bäume reduziert werden. Zu diesem Zweck wird an zwei Stellen randomisiert:

1. Für das Training jedes Baums wird zufällig eine Teilmenge der Trainingsdaten ausgewählt.
2. Für jeden Split werden aus der Menge aller möglichen Attribute  $K$  Kandidaten zufällig ausgewählt.

Das Prinzip der Random Forests hat Breiman in [8] eingeführt.

**Parameter** Als Implementierung wurde der `W-RandomForest` Operator aus dem *WEKA-Plugin*<sup>3</sup> in RapidMiner verwendet. Nach [41] wurde die Tiefe eines Baums unbeschränkt gelassen. Der Operator hat nun noch die Parameter Anzahl der Bäume  $I$  sowie Anzahl der für einen Split zu betrachtende Attribute  $K$ . Beides sind ganzzahlige, positive Werte. Anders als bei  $k$  nächste Nachbarn wurde für Random Forest zunächst in einigen wenigen Durchläufen mit grober Schrittweite ermittelt, welche Bereiche besonders viel versprechende Parametrisierungen beinhalten. Anschließend wurden diese Regionen detaillierter betrachtet und ein Optimum für  $I = 108$  und  $K = 75$  gefunden. In Abbildung 6.7 ist die Klassifikationsgüte in Abhängigkeit von

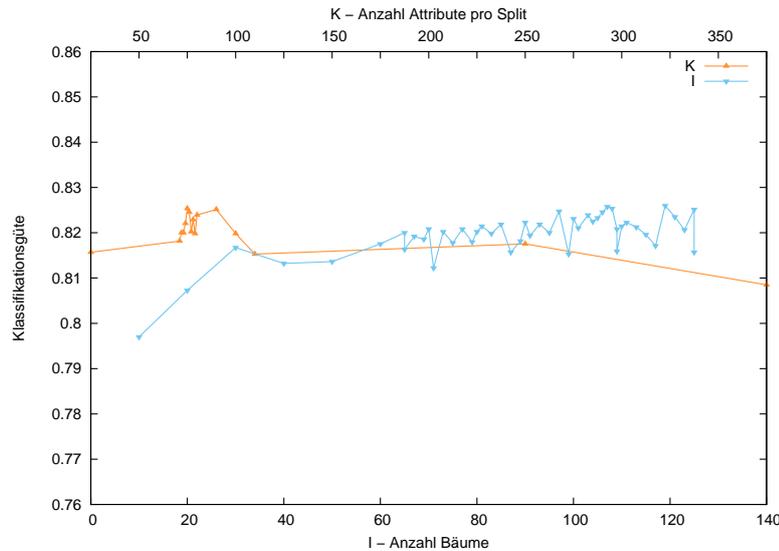


Abbildung 6.7: Die Klassifikationsgüte von Random Forests in Abhängigkeit von der Anzahl an Bäumen und Attributen. Es ist jeweils die höchste Klassifikationsgüte dargestellt.

$I$  und  $K$  abgebildet. Es wurde dabei jeweils die höchste Klassifikationsgüte für die jeweilige Ausprägung der Parameter gewählt.

## Support Vector Machines

Eine *Support Vector Machine* erzeugt eine lineare Entscheidungsfunktion zur Trennung von zwei Klassen in einer Beispielmenge. Dazu wird eine *separierende Hyperebene* bestimmt, die allgemein durch

$$\beta_0 + \boldsymbol{\beta} \cdot \mathbf{x} = 0 \quad (6.5)$$

<sup>3</sup><http://sourceforge.net/projects/rapidminer/files/2.%20Extensions/Weka/>

## 6 Experimente

bestimmt ist [41]. In Abbildung 6.8 ist eine separierende Hyperebene im  $\mathbb{R}^2$  für linear perfekt trennbare Beispiele dargestellt. Um eine möglichst hohe Klassifikationsgüte zu gewährleisten, werden  $\beta_0$  und  $\boldsymbol{\beta}$  so bestimmt, dass ein maximaler Rand um die Hyperebene entsteht, der frei von Beispielen ist. Dazu wird der Abstand der Beispiele beider Klassen mit minimaler Distanz zur Hyperebene maximiert. In Abbildung 6.8 ist der Rand durch die gestrichelten Linien gekennzeichnet. Beispiele mit minimalem Abstand zur Hyperebene liegen auf dem Rand und werden als *Stützvektoren* bezeichnet. Für beliebige Beispiele  $\mathbf{x}_i$  kann mit

$$f(\mathbf{x}_i) = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}_i \quad (6.6)$$

die vorzeichenbehaftete Distanz von  $\mathbf{x}_i$  zur Hyperebene bestimmt werden. Es ist also ausreichend das Vorzeichen von (6.6) zu bestimmen, um entscheiden zu können, zu welcher Klasse ein Beispiel  $\mathbf{x}_i$  zuzuordnen ist:

$$\hat{y} = \text{sign}(f(\mathbf{x}_i)) = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}_i. \quad (6.7)$$

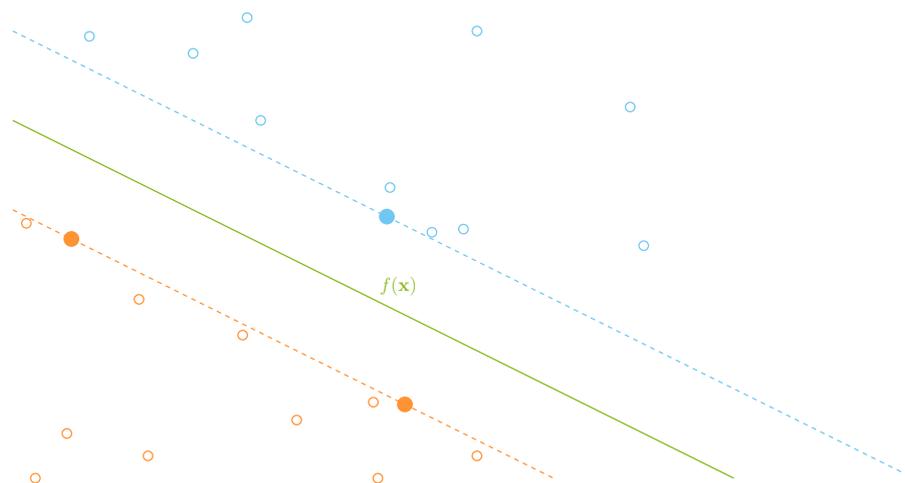


Abbildung 6.8: Zwei Klassen werden im  $\mathbb{R}^2$  linear durch eine Hyperebene  $f(\mathbf{x})$  getrennt. Die Stützvektoren sind die Beispiele beider Klassen, die minimalen Abstand zur Hyperebene haben (gefüllte Kreise auf den gestrichelten Linien).

Eine Support Vector Machine besitzt zwei Komponenten, um auch nicht linear trennbare Daten klassifizieren zu können. Zum einen werden Beispiele auf der falschen Seite des Rands einer Klasse zugelassen. Hierfür werden Schlupfvariablen  $\xi_i \geq 0$  eingeführt, die nur für Beispiele  $\mathbf{x}_i$  innerhalb des Rands oder auf der falschen Seite der Hyperebene größer als null sind. Die Summe der  $\xi_i$  wird in der Trainingsphase durch einen Kostenparameter  $C$  nach oben beschränkt. Die Menge der Stützvektoren wird um alle Beispiele mit  $\xi_i > 0$  erweitert [41].

Oft sind Mengen von  $p$ -dimensionalen Beispielen im  $\mathbb{R}^p$  nicht linear trennbar. Support Vector Machines verwenden Transformationen  $\Phi(\mathbf{x})$  der Beispiele in einen Raum  $\mathbb{R}^q$  mit höherer Dimension, um auch für diesen Fall eine hohe Klassifikationsgüte zu erreichen. Die eigentliche Transformation der Beispiele muss nicht explizit durchgeführt werden. Die Gleichung der Hyperebene kann mithilfe einer Lagrange-Transformation in eine Linearkombination ihrer Stützvektoren überführt werden:

$$\beta_0 + \sum_{i=1}^N \alpha_i \cdot y_i \cdot \langle \mathbf{x}_i, \mathbf{x} \rangle = 0. \quad (6.8)$$

Die  $\alpha_i$  sind hierbei Lagrange-Multiplikatoren und nur für Stützvektoren gilt  $\alpha_i > 0$ . Wird eine Transformation  $\Phi(\mathbf{x})$  durchgeführt ergibt sich die Gleichung der Hyperebene durch

$$\beta_0 + \sum_{i=1}^N \alpha_i \cdot y_i \cdot \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle = 0 \quad (6.9)$$

und es wird deutlich, dass lediglich das innere Produkt von  $\Phi(\mathbf{x}_i)$  und  $\Phi(\mathbf{x})$  im  $\mathbb{R}^q$  benötigt wird. Der Term  $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle$  wird dann durch eine *Kernfunktion*  $K(\mathbf{x}_i, \mathbf{x})$  ersetzt, die das innere Produkt für  $\mathbf{x}_i$  und  $\mathbf{x}$  im  $\mathbb{R}^p$  direkt bestimmt [39].

Die Klassifikation eines Beispiels  $\mathbf{x}_j$  erfolgt nach (6.7) und (6.9) dann durch die Auswertung des Klassifikators

$$\hat{f}(\mathbf{x}_j) = \text{sign}(f(\mathbf{x}_j)) = \text{sign}\left(\beta_0 + \sum_{i=1}^N \alpha_i \cdot y_i \cdot K(\mathbf{x}_i, \mathbf{x}_j)\right). \quad (6.10)$$

Um mehrklassige Probleme mit Support Vector Machines zu lösen, kann z. B. für jede Klasse  $k$  eine Entscheidungsfunktion  $f_k(\mathbf{x})$  erzeugt werden, die Beispiele mit Ausgabe  $y_k$  von Beispielen mit Ausgaben  $y_j \neq y_k$  trennt. Bei dieser *one-against-all* Entscheidungsfunktion wird die Klasse gewählt, die maximalen positiven Abstand für das zu klassifizierende Beispiel bedeutet [48]. Weitere Informationen zu Support Vector Machines und der Erzeugung von Hyperebenen sind u. a. in [86] und [96] zu finden.

**Parameter** Mit dem Operator `LibSVM` wurde eine *C-SVM* mit *Radialbasisfunktion (RBF)* als Kernfunktion trainiert. Die Radialbasisfunktion ist durch

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2} \quad (6.11)$$

definiert. Andere Kernfunktionen wurden nicht betrachtet, da Keerthi und Lin in [56] zeigen, dass sich eine Support Vector Machine mit RBF-Kernfunktion im Grenzfall  $\gamma \rightarrow 0$  dem Verhalten einer linearen Support Vector Machine annähert. Außerdem schlagen die Autoren von [13] und [47] die Verwendung von Radialbasisfunktionen als Kernfunktion vor, da so im Allgemeinen gute Ergebnisse zu erwarten sind.

## 6 Experimente

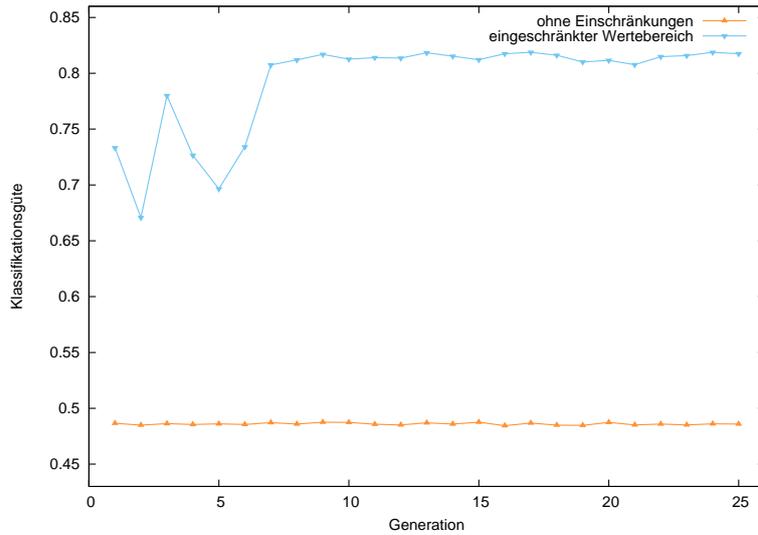


Abbildung 6.9: Die Parameteroptimierung der Support Vector Machine mit unbeschränktem Wertebereich für  $\gamma$  erreicht keine hohe Klassifikationsgüte, da sehr kleine Werte für  $\gamma$  nicht ausgewählt wurden. Beim Durchlauf mit  $\gamma \in [0; 0, 1]$  ist die Güte von Beginn an höher und konvergiert bereits nach etwa sieben Generationen gegen 82%.

Die Parameteroptimierung muss somit neben dem Straffaktor  $C$  noch die *Varianz*  $\gamma$  bestimmen [53]. Da beide Parameter reellwertig und praktisch unbeschränkt sind, wurde von einer Rasteroptimierung abgesehen und stattdessen eine evolutionäre Parameteroptimierung durchgeführt. Um eine gute Region für Werte von  $\gamma$  vorzugeben, wurde eine „Faustregel“ aus [13] verwendet. Die Varianz  $\gamma$  wird hier durch die Breite des Kerns  $\varsigma$  ausgedrückt:

$$\gamma \hat{=} \frac{1}{2 \cdot \varsigma^2}. \quad (6.12)$$

Für  $\varsigma$  wird als guter Parameter

$$\varsigma^* = p \cdot \arg \max_i |\max(x_i) - \min(x_i)|, \quad (6.13)$$

also das Produkt aus Dimensionalität sowie der Kardinalität des Wertebereiches der Beispiele vorgeschlagen. Jeder Wortvektor entstammt dem  $\mathbb{R}^{375}$  und die Werte aller Dimensionen liegen im Bereich  $[0; 25]$ . Nach [13] ist somit ein Wert

$$\gamma^* = \frac{1}{2 \cdot (375 \cdot 25)^2} \approx 5,7 \cdot 10^{-9} \quad (6.14)$$

zu erwarten. Mit diesem Hintergrund sind hohe Klassifikationsgüten für  $\gamma > 0, 1$  unwahrscheinlich und der Wertebereich wurde auf  $[0; 0, 1]$  festgelegt. Ein Testlauf mit

größeren Werten für  $\gamma$  hat diese Wahl bestätigt. Für den Straffaktor  $C$  wurde die Anzahl an Beispielen als obere Schranke gewählt. Als Ergebnis der Parameteroptimierung wurde die Parametrisierung  $C = 574, 531$  und  $\gamma = 0$  ausgegeben.

In Abbildung 6.9 ist die Entwicklung der Klassifikationsgüte für die Parameteroptimierung dargestellt. Zum Vergleich ist auch der Testlauf mit unbeschränkten Intervallen eingetragen.

## Neuronale Netze

*Neuronale Netze* sind durch das Verarbeiten von Informationen und das Erlernen von Wissen durch das menschliche Gehirn inspiriert. Analog zu den Zellen und Synapsen eines Gehirns besteht ein neuronales Netz aus *Neuronen*, die in *Schichten* angeordnet und über *Verknüpfungen* mit Neuronen höherer Schichten verbunden sind [44].

In der untersten Schicht befinden sich  $p$  Neuronen für die  $p$  Attribute der Beispiele. In der obersten Schicht sind  $k$  Neuronen für die  $k$  Ausprägungen der Ausgaben  $y$  implementiert. Dazwischen können sich eine oder mehrere *versteckte Schichten* mit einer variablen Zahl von Neuronen befinden [41]. Häufig werden zweilagige neuronale Netze wie in Abbildung 6.10 verwendet.

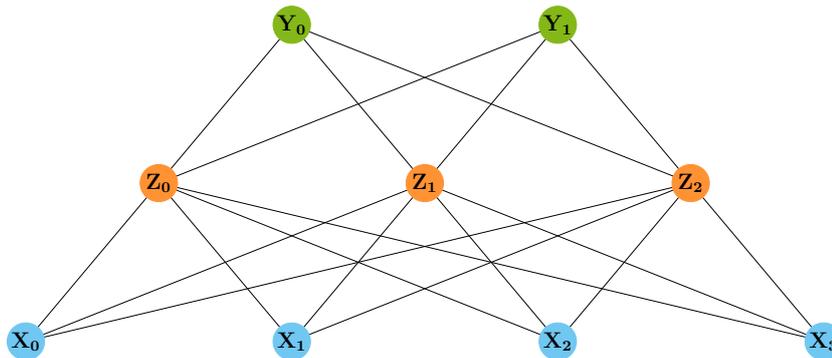


Abbildung 6.10: Beispiel für ein neuronales Netz zur Klassifikation von zweiklassigen Beispielen mit vier Attributen als Eingabe unter Verwendung von einer versteckten Schicht mit drei Neuronen.

Wenn ein Beispiel  $\mathbf{x}$  mit den Attributen  $\langle x_0, \dots, x_p \rangle$  an der untersten Schicht „angelegt“ wird, werden die  $d$  Neuronen  $Z_i$  der versteckten Schicht mit den Werten einer *Aktivierungsfunktion*

$$Z_i = \sigma \cdot (\alpha_{0,i} + \boldsymbol{\alpha}_i^T \cdot \mathbf{X}) \quad (6.15)$$

belegt. Im Folgenden wird hierfür die *sigmoid-Funktion* (6.16) verwendet [41]

$$\sigma(v) = \frac{1}{1 + e^{-v}}. \quad (6.16)$$

Die Eingabe für die oberste Schicht ist dann  $\mathbf{T} = \langle T_0, \dots, T_d \rangle$  mit

$$T_i = \beta_{0,i} + \boldsymbol{\beta}_i^T \cdot \mathbf{Z}, \quad (6.17)$$

## 6 Experimente

welche mit der *softmax-Funktion*:

$$g_k(\mathbf{T}) = \frac{e^{T_k}}{\sum_{l=1}^k e^{T_l}} \quad (6.18)$$

schließlich zur Bestimmung der Ausgabe  $Y_k$  verwendet wird [41]. Weitere Informationen zu neuronalen Netzen sind z. B. in [44] und [83] zu finden.

**Parameter** Zur Erstellung eines neuronalen Netzes zur Klassifikation wurde der RapidMiner-Operator `Neural Net` verwendet. Es wurde eine versteckte Schicht mit den Standardeinstellungen für die Anzahl an Neuronen verwendet. Während der Parameteroptimierung müssen nun noch drei Parameter bestimmt werden [44]. Zwei der Parameter sind reelwertige Größen (*Trägheit* und *Lernrate*) aus  $[0; 1]$ , die einen Kompromiss zwischen Lerngeschwindigkeit und -güte bilden. Bei dem dritten Parameter handelt es sich um einen ganzzahligen, positiven Parameter, der die Anzahl der *Trainingszyklen* beschreibt.

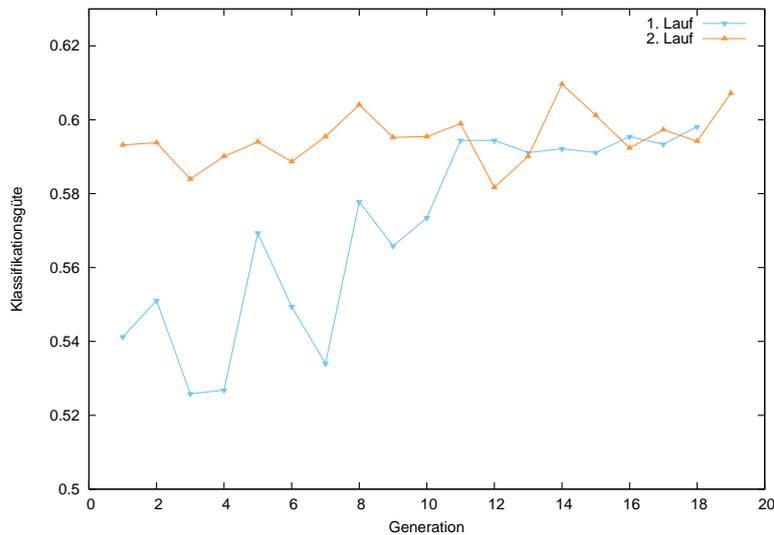


Abbildung 6.11: Vergleich der Entwicklung der Klassifikationsgüte für die evolutionäre Parameteroptimierung eines neuronalen Netzes. Im ersten Lauf wurden für die Lernrate und das das Trägheitsmoment sämtliche Werte zwischen 0 und 1 zugelassen. Die Anzahl der Trainingszyklen wurde durch 100 nach oben beschränkt. Im zweiten Lauf wurden die Grenzen um die im ersten Lauf gefundenen Optima herum gelegt.

Aufgrund des kontinuierlichen Wertebereichs für die Lernrate sowie die Trägheit wurde an dieser Stelle die evolutionäre Parameteroptimierung eingesetzt. Zunächst wurde für die Trägheit und Lernrate der volle Bereich aus  $[0; 1]$  zugelassen und eine

maximale Anzahl von 100 Trainingszyklen festgelegt. In einem zweiten Durchlauf wurde der Wertebereich mit den Ergebnissen der ersten Parameteroptimierung korrigiert, um den optimalen Bereich detaillierter abzutasten. Die Klassifikationsgüte konvergierte bei einer Trägheit von ca. 0,687, einer Lernrate von ca. 0.232 und 51 Trainingszyklen.

Der Verlauf der Klassifikationsgüte beider Trainingsläufe ist in Abbildung 6.11 zu sehen. Zum Ende des ersten Laufs schwankt die Parameteroptimierung um ein lokales Minimum. Nach Korrektur der Grenzen konnte die Klassifikationsgüte, wenn auch nur im niedrigen einstelligen Prozentbereich, verbessert werden.

### 6.3.2 Resultate

In einer empirischen Studie von überwachten Lernverfahren für die Klassifikation [14] haben Caruana und Niculescu-Mizil u. a. die fünf hier verwendeten Algorithmen untersucht. Sie kommen zu dem Schluss, dass Random Forests und Support Vector Machines in vielen Situationen gute Ergebnisse erzielen. Neuronale Netze sind konkurrenzfähige Alternativen, die abhängig vom verwendeten Datensatz nur geringfügig schlechter abschneiden. Aus den Experimenten in [14] konnte außerdem geschlossen werden, dass sich die Klassifikationsgüte von  $k$  nächste Nachbarn im Mittelfeld befindet und Naive Bayes Klassifikatoren häufig besonders schlechte Ergebnisse liefern. Bei Betrachtung der Gegenüberstellung der Klassifikationsgüten für die in diesem Abschnitt durchgeführten Experimenten in Abbildung 6.12 fällt auf, dass diese Resultate mit Ausnahme von  $k$  nächste Nachbarn auch hier gültig sind.

Die Klassifikationsgüte von Random Forests und Support Vector Machines ist auf dem verwendeten Datensatz mit 82,19% (RF) und 81,88% (SVM) nahezu identisch. Beide Verfahren profitieren davon, relativ geschützt vor einer Überanpassung an die Trainingsdaten zu sein [41]. Dies ist bei neuronalen Netzen anders. Eine Überanpassung kann bei einer zu hohen Zahl von Neuronen oder Trainingszyklen eintreten [95]. Außerdem ist die Parameteroptimierung schwierig, weil neuronale Netze in lokalen Optima „hängen bleiben“ können und das Ergebnis von den Startwerten abhängig ist [44]. Die Klassifikationsgüte von 60,96% im durchgeführten Experiment muss also nicht bedeuten, dass neuronale Netze für die Vorhersage von Verkehrszuständen auf Autobahnen nicht geeignet sind. Allerdings sprechen die guten Ergebnisse von Random Forests sowie Support Vector Machines gegen einen Einsatz von neuronalen Netzen, da das Training dieser beiden Verfahren deutlich weniger Zeit in Anspruch nimmt und nicht von den Startwerten abhängig ist. Besonders Random Forests sind aufgrund der intuitiven Parameter für Anzahl der Bäume  $I$  und Anzahl der Kandidaten  $K$  für einen Split auch durch weniger versierte Anwender leicht zu trainieren.

Eine ähnlich einfache Trainingsphase bietet auch das  $k$  nächste Nachbarn Verfahren, das mit 80,17% nur marginal schlechter als Random Forests und Support Vector Machines abgeschnitten hat. Darüber hinaus ist das Verfahren sehr einfach inkrementell zu gestalten und auf den ersten Blick für die Verarbeitung von Datenströmen wie

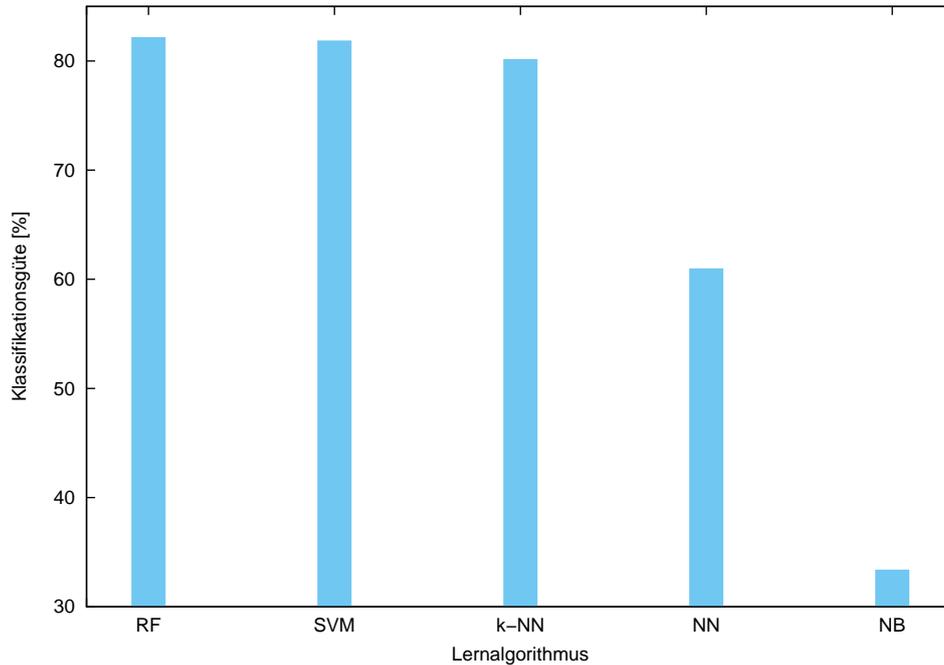


Abbildung 6.12: Plot der Klassifikationsgüten der einzelnen Lernalgorithmen – Random Forests (RF), Support Vector Machines (SVM),  $k$  nächste Nachbarn (k-NN), neuronale Netze (NN), Naive Bayes (NB).

den hier betrachteten Verkehrsdaten gut geeignet. Weil dieses Verfahren jedoch ein träger Lerner ist und erst zur Testphase ein Modell erzeugt, kann eine inkrementelle Verwendung zu Problemen bei der Echtzeit-Verarbeitung von Datenströmen führen, wenn zu viele Beispiele verglichen werden müssen.

Die Klassifikationsgüte von 33,39% für den Naive Bayes Klassifikator schließt dieses Verfahren für die Vorhersage von Zustandsübergängen in Verkehrssystemen aus. Der Grund für das schlechte Abschneiden von Naive Bayes ist die stark unbalancierte Klassenverteilung [32]. Die Apriori-Wahrscheinlichkeiten von seltenen Transitionen sind sehr viel kleiner als Apriori-Wahrscheinlichkeiten für die Hauptklassen *grün*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *grün*.

Dieses Resultat spiegelt sich auch in den Konfusionsmatrizen in Tabelle 6.4 wieder. Wie bereits zu Beginn dieses Kapitels angedeutet, sind besonders die Sensitivität und Genauigkeit bei der Erkennung der Klassen *grün*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *rot* von Interesse. Wenn solche Transitionen erkannt werden, die Sensitivität also hoch ist, können Maßnahmen eingeleitet werden, um die Verkehrsdichte auf dem Streckenabschnitt zu reduzieren. Andererseits sollte die Anzahl an „Fehlalarmen“ niedrig und damit die Genauigkeit hoch sein. Ansonsten sinkt das Vertrauen in ein Vorhersagesystem und

Warnungen werden von den Zuständigen mehr und mehr ignoriert [5].

Für die drei Modelle mit hoher Klassifikationsgüte liegen Sensitivität und Genauigkeit für den Verkehrszusammenbruch bei etwa 80%. Der „Gesamtsieger“ ist hier das Modell der Support Vector Machines, welches sich mit über 82% in beiden Kategorien leicht von Random Forests und  $k$  nächsten Nachbarn absetzt. Hohe Werte sind hier jedoch aufgrund des hohen Anteils an Beispielen mit Ausgabe *grün*  $\rightarrow$  *gelb* in der Trainingsmenge zu erwarten, wenn die Gesamtgüte eines Verfahrens hoch ist.

Trotz des mit 3% relativ geringen Anteils an Zustandsübergängen *gelb*  $\rightarrow$  *rot* Transitionen sind auch für diese Klasse Sensitivität und Genauigkeit für Support Vector Machines, Random Forests und  $k$  nächste Nachbarn hoch. Die Sensitivität beträgt für alle drei Verfahren 79,74%. Nur bei der Genauigkeit setzten sich die Support Vector Machines mit 85,31% leicht von Random Forests (83,65%) und etwas deutlicher von  $k$  nächste Nachbarn (79,74%) ab. Auch in dieser Kategorie bieten Support Vector Machines somit insgesamt etwas bessere Ergebnisse.

Die beiden Verfahren mit einer schlechteren Gesamtgüte schneiden auch bei der Betrachtung der Transitionen *grün*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *rot* erwartungsgemäß schlechter ab: Neuronale Netze erreichen für die Klasse *grün*  $\rightarrow$  *gelb* noch rund 62% Genauigkeit und Sensitivität, sind mit der verwendeten Parametrisierung jedoch nicht für die Erkennung von Übergängen vom synchronen Fluss zum Stau geeignet (20,26% Sensitivität und 40,26% Genauigkeit). Der Naive Bayes Klassifikator erreicht für die Klasse *grün*  $\rightarrow$  *gelb* aufgrund der hohen Apriori-Wahrscheinlichkeit für diesen Zustandsübergang noch 61,91% Genauigkeit. Jedoch beträgt die Sensitivität lediglich 26,62%. Auch in Bezug auf die Klasse *gelb*  $\rightarrow$  *rot* unterliegt Naive Bayes allen anderen Verfahren (14,38% Sensitivität und 5,14% Genauigkeit).

Auch die direkten Zustandsübergänge in entspanntere Verkehrsphasen (*rot*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *grün*) werden von Random Forests, Support Vector Machines und  $k$  nächste Nachbarn mit Genauigkeit und Sensitivität von über 80% korrekt klassifiziert. Überhaupt liegen die Maxima für Genauigkeit und Sensitivität dieser drei Verfahren bei der Klasse *rot*  $\rightarrow$  *gelb*. Für die kurzfristige Prävention von Verkehrszusammenbrüchen und Schockwellen sind diese Transitionen weniger wichtig. Eine hohe Sensitivität und Genauigkeit ist hier jedoch sicherlich nützlich, wenn Verkehrsprognosen für die Fahrzeugführer selbst erstellt werden. Aber auch in der Verkehrsüberwachung sind Vorhersagen zur Entspannung der Verkehrslage von Interesse. Ein sich auflösender Stau kann wegen dem dadurch steigenden Verkehrsfluss zu Folgestaus stromaufwärts führen, falls die Verkehrsdichte hier bereits erhöht ist [9].

Für die nachfolgenden Experimente werden aufgrund der geringen Unterschiede in der Klassifikationsgüte zu Support Vector Machines sowie der zeitlich deutlich kürzeren Trainingsphase Random Forests verwendet. Die Resultate der Parameteroptimierung sind in allen Fällen ähnlich und werden deshalb von jetzt an nicht weiter aufgeführt.

## 6 Experimente

Tabelle 6.4: Konfusionsmatrizen der fünf Lernalgorithmen für das hier vorgestellte Experiment.

Naive Bayes		Y						Genauigkeit
		<i>grün → gelb</i>	<i>grün → rot</i>	<i>gelb → rot</i>	<i>gelb → grün</i>	<i>rot → grün</i>	<i>rot → gelb</i>	
$\hat{f}(\mathbf{X})$	<i>grün → gelb</i>	590	3	3	356	1	0	61,91%
	<i>grün → rot</i>	668	49	45	539	10	21	3,68%
	<i>gelb → rot</i>	171	4	22	224	1	6	5,14%
	<i>gelb → grün</i>	712	1	3	854	2	2	54,26%
	<i>rot → grün</i>	17	5	26	89	42	52	18,18%
	<i>rot → gelb</i>	59	4	54	153	11	68	19,48%
Sensitivität		26,61%	74,21%	14,38%	38,56%	62,69%	45,64%	

(a) Naive Bayes

kNN		Y						Genauigkeit
		<i>grün → gelb</i>	<i>grün → rot</i>	<i>gelb → rot</i>	<i>gelb → grün</i>	<i>rot → grün</i>	<i>rot → gelb</i>	
$\hat{f}(\mathbf{X})$	<i>grün → gelb</i>	1773	14	5	381	0	3	81,48%
	<i>grün → rot</i>	31	45	9	6	1	0	48,91%
	<i>gelb → rot</i>	13	6	122	2	0	10	79,74%
	<i>gelb → grün</i>	392	0	4	1795	21	5	80,97%
	<i>rot → grün</i>	7	0	1	26	42	6	51,22%
	<i>rot → gelb</i>	1	1	12	5	3	125	85,03%
Sensitivität		79,97%	68,18%	79,74%	81,04%	62,69%	83,89%	

(b)  $k$  nächste Nachbarn

Random Forest		Y						Genauigkeit
		<i>grün → gelb</i>	<i>grün → rot</i>	<i>gelb → rot</i>	<i>gelb → grün</i>	<i>rot → grün</i>	<i>rot → gelb</i>	
$\hat{f}(\mathbf{X})$	<i>grün → gelb</i>	1889	20	5	403	0	4	81,39%
	<i>grün → rot</i>	15	35	8	0	0	0	60,34%
	<i>gelb → rot</i>	6	7	122	2	0	9	83,56%
	<i>gelb → grün</i>	305	3	10	1790	26	7	83,61%
	<i>rot → grün</i>	1	0	0	17	38	3	64,41%
	<i>rot → gelb</i>	1	1	8	3	3	126	88,73%
Sensitivität		82,21%	80,81%	79,74%	80,81%	56,72%	84,56%	

(c) Random Forests

SVM		Y						Genauigkeit
		<i>grün → gelb</i>	<i>grün → rot</i>	<i>gelb → rot</i>	<i>gelb → grün</i>	<i>rot → grün</i>	<i>rot → gelb</i>	
$\hat{f}(\mathbf{X})$	<i>grün → gelb</i>	1820	22	5	351	1	4	82,61%
	<i>grün → rot</i>	27	38	9	0	0	0	51,35%
	<i>gelb → rot</i>	7	4	122	4	0	6	85,31%
	<i>gelb → grün</i>	359	1	8	1835	21	7	82,25%
	<i>rot → grün</i>	1	0	0	21	42	4	61,76%
	<i>rot → gelb</i>	3	1	9	4	3	128	86,49%
Sensitivität		82,09%	57,58%	79,74%	82,84%	62,69%	85,91%	

(d) Support Vector Machines

NN		Y						Genauigkeit
		<i>grün → gelb</i>	<i>grün → rot</i>	<i>gelb → rot</i>	<i>gelb → grün</i>	<i>rot → grün</i>	<i>rot → gelb</i>	
$\hat{f}(\mathbf{X})$	<i>grün → gelb</i>	1377	39	5	739	3	5	62,79%
	<i>grün → rot</i>	0	0	0	0	0	0	0%
	<i>gelb → rot</i>	7	13	31	12	13	13	40,26%
	<i>gelb → grün</i>	831	22	0	1457	33	34	59,91%
	<i>rot → grün</i>	2	1	55	1	6	1	54,55%
	<i>rot → gelb</i>	0	3	37	6	12	96	62,34%
Sensitivität		62,11%	0%	20,26%	65,78%	8,96%	64,43%	

(e) Neuronale Netze

## 6.4 Einfluss des Vorhersagehorizonts

Um einen Verkehrszusammenbruch bzw. die Stautentstehung zu verhindern oder wenigstens hinauszuzögern, ist es wünschenswert, möglichst früh auf einen wahrscheinlichen Zustandsübergang hingewiesen zu werden. In diesem Abschnitt wird deshalb untersucht, wie sich die Vorhersagegüte ändert, wenn der Vorhersagehorizont verlängert bzw. verkürzt wird. Dazu wurde das Experiment aus Abschnitt 6.3 mit Random Forests verwendet und mit unterschiedlichen Datensätzen für die Zählschleife *DoWe* aus Tabelle 6.3 ausgeführt. Als Vorhersagehorizont wurden Werte aus [5; 30] mit einer Schrittweite von fünf Minuten gewählt. In Abbildung 6.13 ist die Vorhersagegüte

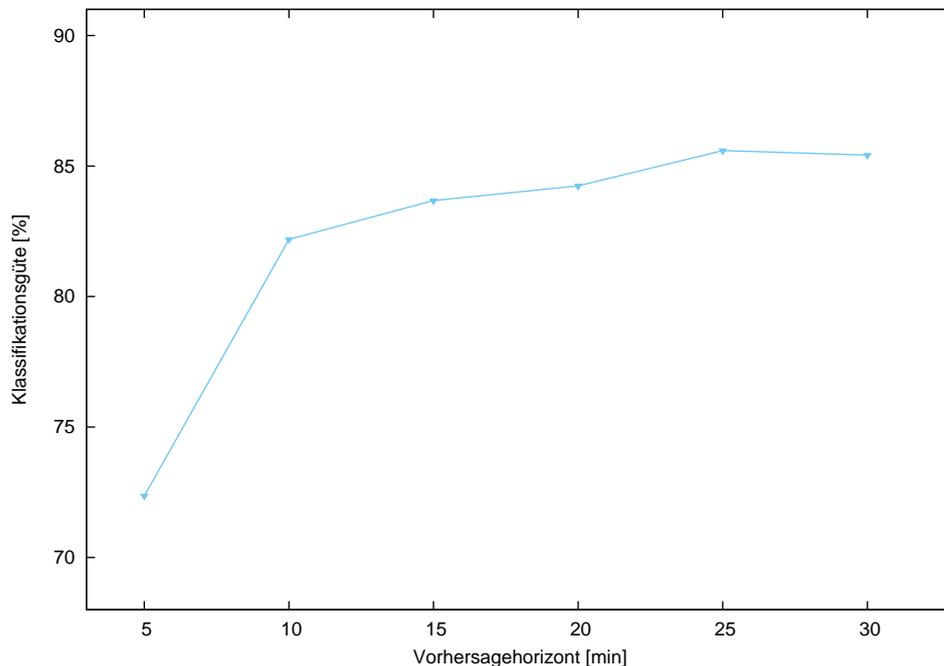


Abbildung 6.13: Plot der Vorhersagegüte für das Random Forest Verfahren bei unterschiedlichem Vorhersagehorizont.

in Abhängigkeit des Vorhersagehorizonts dargestellt. In Relation zum Vorhersagehorizont von zehn Minuten aus den Experimenten in Abschnitt 6.3 steigt die klassenübergreifende Vorhersagegüte von 82,19% zunächst annähernd konstant bis auf 85,59% für einen Vorhersagehorizont von 25 Minuten, bevor sie für 30 Minuten leicht auf 85,42% absinkt. Wird der Vorhersagehorizont auf fünf Minuten verkürzt, werden lediglich 72,36% aller Beispiele korrekt klassifiziert.

Mit dem Wissen, dass sich ein Verkehrssystem nach einem Zustandsübergang mit hoher Wahrscheinlichkeit eine längere Zeit in ein und der selben Verkehrsphase be-

finden wird (siehe Abschnitt 2.3.3), lässt sich die steigende Vorhersagegüte erklären. Befindet sich das Verkehrssystem in einem Phasenübergang, wird es in endlicher Zeit von der aktuellen Verkehrsphase in eine andere Phase wechseln. Wird z. B. die Transition elf oder zwölf Minuten nach der Erkennung abgeschlossen ist eine Klassifikation  $grün \rightarrow gelb$  für das Modell mit zehn Minuten Vorhersagehorizont falsch. Der synchrone Fluss bleibt jedoch im Anschluss i. d. R. eine längere Zeit bestehen, wodurch Modelle mit einem längeren Vorhersagehorizont dieses Beispiel korrekt klassifizieren und insgesamt eine höhere Klassifikationsgüte erreichen.

Tabelle 6.5: Konfusionsmatrix für einen Random Forest, der mit dem Datensatz *DoWe-6-0710-25* trainiert wurde und einen Vorhersagehorizont von 25 Minuten besitzt.

Random Forest		Y						Genauigkeit
		$grün \rightarrow gelb$	$grün \rightarrow rot$	$gelb \rightarrow rot$	$gelb \rightarrow grün$	$rot \rightarrow grün$	$rot \rightarrow gelb$	
$\hat{f}(X)$	$grün \rightarrow gelb$	2346	38	5	366	0	0	84,24%
	$grün \rightarrow rot$	22	121	11	0	0	0	78,57%
	$gelb \rightarrow rot$	4	7	81	0	1	2	85,26%
	$gelb \rightarrow grün$	241	2	7	2143	35	3	88,15%
	$rot \rightarrow grün$	1	0	1	29	108	11	72,00%
	$rot \rightarrow gelb$	0	0	4	3	5	121	90,98%
	Sensitivität	89,75%	72,02%	74,31%	83,35%	72,48%	88,32%	

Wird die Konfusionsmatrix für 25 Minuten Vorhersagehorizont aus Tabelle 6.5 mit der für einen Random Forest bei zehnminütigem Vorhersagehorizont aus Tabelle 6.4(c) verglichen, fällt auf, dass Sensitivität und Genauigkeit für den Verkehrszusammenbruch  $grün \rightarrow gelb$  steigen. Dieser Phasenübergang wird mit 89,75% Sensitivität (+4,54%) und 84,24% Genauigkeit (+2,85%) erkannt.

Die Transition  $gelb \rightarrow rot$  wird jedoch nur noch mit einer Sensitivität von 74,31% (-5,43%) und einer Genauigkeit von 78,57% (-4,99%) vorhergesagt. Wird der Verlauf der Klassenverteilung aus Tabelle 6.3 in Abschnitt 6.1 betrachtet, scheint dieser Güteverlust für die Klasse  $gelb \rightarrow rot$  auf den ersten Blick mit dem stark abnehmenden Anteil dieser Klasse zusammenzuhängen. Unter den Phasenübergängen, die zu einer Verschärfung der Verkehrslage führen halbiert sich der Anteil von Beispielen mit Ausgabe  $gelb \rightarrow rot$  annähernd von 3,33% für fünf Minuten auf 1,71% bei einem Vorhersagehorizont von 25 Minuten. Der relativ hohe Anteil von Beispielen mit Ausgabe  $grün \rightarrow gelb$  sinkt für die Datensätze nur marginal von 45,82% auf 44,97%, während die, eine Phase „überspringende“, Transition  $grün \rightarrow rot$  mit 3,34% innerhalb von 25 Minuten deutlich häufiger als innerhalb von fünf Minuten (0,46%) beobachtet wird.

Für die zu einer Entspannung der Verkehrslage führenden Phasenübergänge sind jedoch ähnliche Zu- und Abnahmen zu beobachten – der Anteil von  $gelb \rightarrow grün$  und  $rot \rightarrow gelb$  sinken um 1,87% auf 44,62% bzw. um 1,15% auf 2,36% während der Anteil der Klasse  $rot \rightarrow grün$  um 2,61% auf 2,98% steigt. Sowohl Sensitivität und Güte steigen für die Klasse  $rot \rightarrow gelb$ , sodass der sinkende Anteil einer Klasse nicht direkt als Ursache für ein Absinken von Sensitivität und Genauigkeit dieser Klassen verantwortlich gemacht werden kann.

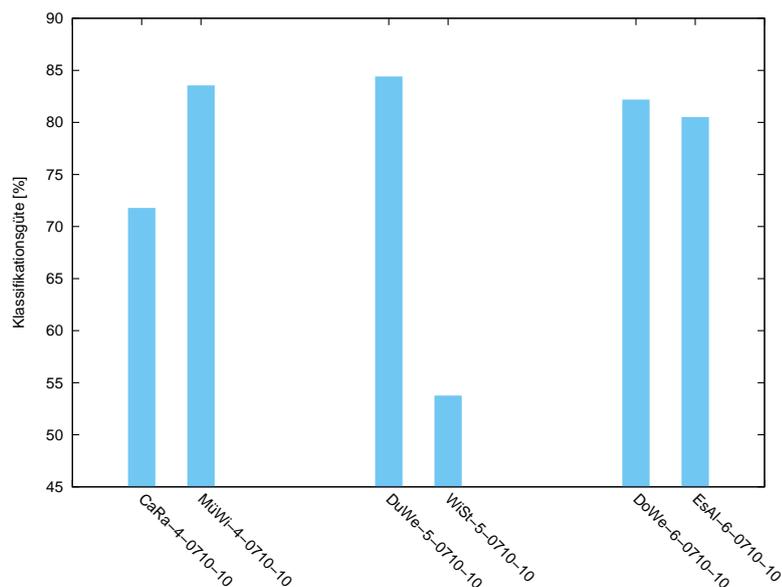


Abbildung 6.14: Plot der Klassifikationsgüte für Datensätze aus unterschiedlichen Clustern.

## 6.5 Klassifikationsgüte in den einzelnen Clustern

In diesem Abschnitt wird untersucht, ob das vorgestellte Verfahren für die zu Beginn dieses Kapitels identifizierten Gruppen von Zählschleifen vergleichbare Ergebnisse liefert. Dazu wurde das Basisexperiment für Random Forests mit den Datensätzen

- *CaRa-4-0710-10*
- *DoWe-6-0710-10*
- *DuWe-5-0710-10*
- *EsAl-6-0710-10*
- *MüWi-4-0710-10*
- *WiSt-5-0710-10*

durchgeführt. Abbildung 6.14 zeigt, dass die Modelle unter Verwendung von Datensätzen aus Cluster 6 vergleichbar hohe Klassifikationsgüten erreichen. Bei den Clustern 4 und 5 unterscheidet sich die Klassifikationsgüte jedoch stark. Mit jeweils einem Datensatz werden über 80% erreicht, während die Modelle für den anderen Datensatz deutlich schlechter abschneiden.

Wird zunächst nur Cluster 4 betrachtet fällt anhand Tabelle 6.2 auf, dass schlechtere Ergebnisse mit Datensätzen erzielt werden, wenn sie von einer Zählschleife mit

niedrigem Belegungsgrad stammen. Jedoch wird auch für den Datensatz *DuWe-5-0710-10* eine hohe Klassifikationsgüte erreicht. Der Belegungsgrad der Zähl Schleife *DuWe* liegt mit 28,33% aber zwischen dem Belegungsgrad der Zähl Schleifen *CaRa* und *WiSt*. Auch die restlichen Werte der Zähl Schleifen lassen keine eindeutigen Schlüsse zu.

Wird jedoch die Klassenverteilung für die sechs Datensätze in Tabelle 6.3 berücksichtigt fällt auf, dass niedrige Klassifikationsgüten mit vergleichsweise geringen Anteilen der Klassen *grün*  $\rightarrow$  *gelb* und *gelb*  $\rightarrow$  *grün* einhergehen. Insbesondere der Anteil von *gelb*  $\rightarrow$  *rot* ist für die Datensätze *CaRa-4-0710-10* und *WiSt-5-0710-10* im Vergleich zu allen anderen Datensätzen sehr hoch. Diese Beobachtung legt die Schlussfolgerung nahe, dass das hier vorgestellte Verfahren vom Ungleichgewicht der Klassen profitiert.

## 6.6 Verwendung von Modellen zur Vorhersage für andere Zähl Schleifen

Sensornetzwerke sind meist nicht statisch, sondern es können Sensoren entfernt, ausgetauscht oder hinzugefügt werden [29]. Auch für das Sensornetzwerk aus Abbildung 3.1, dem die hier verwendeten Daten entstammen, sind Änderungen möglich. Bei baulichen Maßnahmen können z. B. zusätzliche Sensoren ausgebracht werden. Ebenso können sich die Rahmenbedingungen wie z. B. Geschwindigkeitsbegrenzungen oder die Anzahl von Spuren mit der Zeit ändern und es entsteht ein Bruch in den Zeitreihen, da nun andere Werte gemessen werden. In solchen Situationen sind prototypische Modelle nützlich, um nicht erst über Monate Daten sammeln zu müssen, um ein lokales Modell zu erzeugen.

Deshalb wird in im Folgenden untersucht, ob die im vorherigen Abschnitt erzeugten Modelle für die Klassifikation der Beispiele aus anderen Datensätzen geeignet sind. Aufgrund der Ergebnisse aus Abbildung 6.14 beschränken sich die Experimente auf das jeweils beste Modell pro Cluster. In Tabelle 6.6 ist die Klassifikationsgüte der Modelle für die Beispiele der einzelnen Datensätze aufgeführt. Es zeigt sich deutlich, dass sich keines der bislang gefundenen Modelle eignet, als Modell für eine der anderen Zähl Schleifen zu fungieren. Es fällt jedoch auf, dass die Daten für *CaRa* und *WiSt* erneut die schlechtesten Ergebnisse liefern.

Auch ein gemeinsames Modell für die sechs Datensätze ist nicht als globales Vorhersagemodell geeignet. Mit 50% der Beispiele aus dem Datensatz *ALLE-0710-01* wurde ein Modell trainiert, das Wortvektoren aller betrachteten Zähl Schleifen in der Trainingsphase verwendet hat. Mit der stratifiziert gezogenen Testmenge der restlichen 50% aller Beispiele aus dem Datensatz wurde eine Klassifikationsgüte von 68,74% erreicht, die weiterhin weit unter den Werten von lokalen Modellen liegt.

Tabelle 6.6: Klassifikationsgüte von drei Modellen für Datensätze, die nicht zum Training verwendet wurden und von fremden Zählschleifen stammen.

Datensatz	Modell trainiert für		
	DoWe-6-0710-10	DuWe-5-0710-10	MüWi-4-0710-10
CaRa-4-0710-10	22,04%	20,51%	21,00%
DoWe-6-0710-10	—	45,28%	43,97%
DuWe-5-0710-10	48,38%	—	42,18%
EsAl-6-0710-10	33,13%	28,19%	34,11%
MüWi-4-0710-10	39,85%	39,85%	—
WiSt-5-0710-10	28,01%	16,16%	29,25%

## 6.7 Evaluation mit Daten anderer Zeiträume

Dieser Abschnitt untersucht, ob Modelle robust genug sind, um Daten aus späteren Zeiträumen zu klassifizieren oder ob ein inkrementelles Modell verwendet werden muss. Dazu werden die Modelle von Random Forests, Support Vector Machines und  $k$  nächste Nachbarn aus dem Abschnitt 6.3 verwendet, um die Beispiele aus dem Datensatz *DoWe-6-1112-10* zu klassifizieren. Die Modelle selbst wurden also ohne die Monate November und Dezember 2010 der Testmenge trainiert.

Die Klassifikationsgüte nimmt für alle drei betrachteten Modelle stark ab. Abbildung 6.15 zeigt, dass die Klassifikationsgüte im Vergleich zum Datensatz *DoWe-6-0710-10* in Abbildung 6.12 jeweils um rund 30% sinkt. Die Klassenverteilungen in den beiden Datensätzen unterscheidet sich nur geringfügig. Lediglich die Klassen *grün*  $\rightarrow$  *rot* und *rot*  $\rightarrow$  *grün* sind in der in diesem Abschnitt verwendeten Testmenge relativ klein.

In einem weiteren Experiment wurde der Datensatz *DoWe-6-0712-10* verwendet, der Wortvektoren sämtlicher Monate enthält. Es wurden 50% der Beispiele für eine Testphase zurückgehalten. Die Aufteilung erfolgte durch Stratifikation. Mit der Hälfte der Beispiele in der Trainingsmenge wurde ein Random Forest Modell nach dem Schema aus Abbildung 6.3 in Abschnitt 6.3 erzeugt. Die in der Trainingsphase nicht berücksichtigten Beispiele wurden anschließend verwendet um das Modell zu evaluieren. Es erreicht eine Klassifikationsgüte von 80,81% und nähert sich so dem Level für den Datensatz *DoWe-6-0710-10* an. Es lässt sich somit vermuten, dass ein Konzeptdrift für den betrachteten Zeitraum vorliegt und sich die Motifs mit der Zeit ändern.

Wie bereits im Abschnitt 2.3 erwähnt, haben Wetterbedingungen einen Einfluss auf das Verkehrssystem. So sinken bei Regen, Schnee, Nebel und ähnlich schlechten Wetterbedingungen die gemessenen Geschwindigkeiten um bis zu 15% und der Verkehrsfluss kann um bis zu 30% abnehmen [20, 69]. Auch das Unfallrisiko steigt bei signifikant schlechterem Wetter an [55]. Fahrer werden darüber hinaus auch durch fehlendes Tageslicht negativ beeinflusst [12]. Mit diesem Hintergrund kann also die These aufgestellt werden, dass sich über ein Jahr verteilt der Verlauf und die Anzahl von Motifs in Verkehrsdaten ändern werden. Da jedoch weder detaillierte Wetterda-

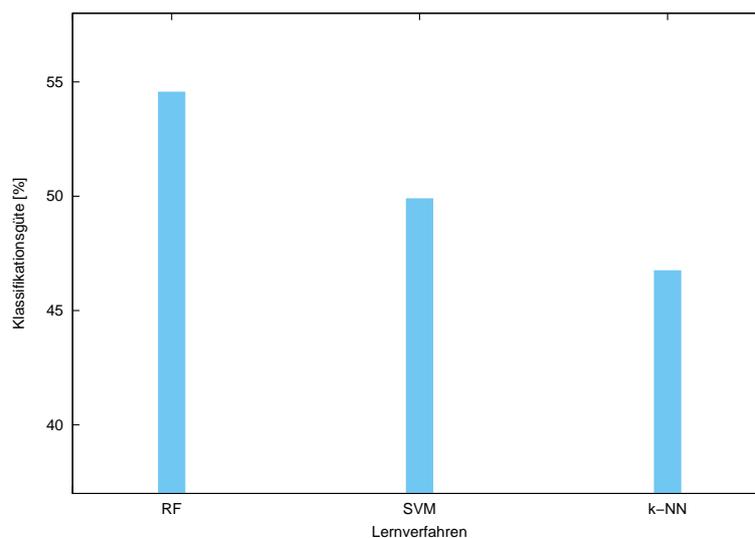


Abbildung 6.15: Die Klassifikationsgüte der Modelle Random Forests (RF), Support Vector Machines (SVM) und  $k$  nächste Nachbarn (k-NN) für die Monate November und Dezember nach Training mit Daten aus den Monaten Juli bis Oktober.

ten für den betrachteten Zeitraum noch Verkehrsdaten für andere Jahre vorliegen, kann diese Vermutung im Rahmen der vorliegenden Arbeit nicht empirisch gestützt werden.

### 6.8 Anreicherung der Wortvektoren mit zusätzlichen Informationen

In Kapitel 5 wurde betont, dass die vorgestellte Repräsentation geeignet ist, um durch zusätzliche Informationen erweitert zu werden. In diesem Abschnitt wird deshalb der Datensatz *WiSt-5-0710-10* angereichert, für den in Abschnitt 6.5 eine nur sehr niedrige Klassifikationsgüte erreicht wurde. Da jedoch keine Daten von externen Datenquellen wie Floating-Cars oder Wetterstationen für den Zeitraum vorliegen, werden hierfür Daten des selben Sensornetzwerks verwendet. Mit dem Wochentag und der Minute des Tages werden zwei zeitliche Attribute eingefügt. Darüber hinaus werden die Verkehrsdichten der letzten fünf Minuten der beiden benachbarten Spuren in den spärlich besetzten Vektor aufgenommen. Der Datensatz *WiSt-5-0710-10\** besitzt also pro Beispiel zwölf weitere Attribute, die in der Trainingsphase berücksichtigt werden.

Für die Trainingsphase wurde der selbe Prozess wie in Abschnitt 6.5 verwendet.

Während das Modell für den Datensatz *WiSt-5-0710-10* eine Klassifikationsgüte von lediglich 53,77% erreicht hat, konnte durch die Anreicherung der Wortvektoren mit dem Datensatz *WiSt-5-0710-10\** eine Steigerung um rund 10% erreicht werden. Das Random Forest Modell für den Datensatz mit zeitlichen Informationen und der Verkehrsdichte der benachbarten Spuren erreicht eine Klassifikationsgüte von 63,34% bei ansonsten konstant bleibenden Charakteristiken wie der Klassenverteilung und den Wortvektoren aus dem für das Experiment in Abschnitt 6.5 verwendeten Datensatz.

## 6.9 Vergleich mit anderen Ansätzen

Zum Abschluss dieses Kapitels wird nun betrachtet, was für eine Vorhersagegüte bei direkter Verwendung der Rohdaten in der Trainingsphase erreicht wird. Anstatt von spärlich besetzten Vektoren wird ein Beispiel  $(\mathbf{x}, y)$  nun durch das Datentupel

$$(\langle \text{minute\_tag}, \text{werktag}, j\_ges, j\_lkw, p\_bel, v\_lkw, v\_pkw \rangle, \text{transition})$$

repräsentiert. Die im Folgenden verwendeten Datensätze wurden für die selben Zählschleifen und Zeiträume wie in den bisherigen Experimenten erzeugt. Die Klassenverteilung ist dementsprechend identisch.

Es werden zwei unterschiedliche Ansätze verfolgt. Analog zur Verwendung von Motifs wird zunächst versucht, die Werte der Attribute selbst zur Vorhersage von Zustandsübergängen zu verwenden. Ein anderer Ansatz nutzt aus, dass viele Staus aufgrund des Wochentags und der Tageszeit vorhergesagt werden können. So sind durch Untersuchung der Tagesganglinie für einzelne Streckenabschnitte Zeiträume identifizierbar, die mit einer hohen Wahrscheinlichkeit gestaut sind [18]. Die Eingabemenge für letztern Ansatz sind also die Attribute *minute\_tag* und *werktag*, sowie der Zustandsübergang. Auch auf diesen Datensätzen erzielten Support Vector Machines,  $k$  nächste Nachbarn und Random Forests ähnliche Ergebnisse. Hier werden deshalb exemplarisch die Ergebnisse für Random Forests repräsentiert.

In Abbildung 6.16 ist zunächst die klassenübergreifende Klassifikationsgüte der beiden Ansätze für alle sechs betrachteten Zählschleifen dargestellt. Zum Vergleich sind die Werte des Motif-Verfahren aus Abbildung 6.14 in Abschnitt 6.5 erneut abgebildet.

Es fällt auf, dass eine Vorhersage von Zustandsübergängen alleine anhand Wochentag und Tageszeit für die betrachteten Zählschleifen nicht möglich ist. Dieses Resultat ist mit dem Hintergrund zur Entstehung von Verkehrsstaus in Abschnitt 2.3.3 nicht überraschend. Zufällige Ereignisse sind Auslöser für Staus die sich Minuten später bilden. Nur die Erhöhung der Verkehrsdichte an einem Autobahnabschnitt kann mit einem solchen Modell prognostiziert werden, da z. B. Zeiten für den Berufsverkehr bekannt sind.

Bei der Betrachtung der Ergebnisse der Modelle, die ohne Motifs auf den Attributen der Zeitreihe trainiert wurden, ist die Klassifikationsgüte für alle der sechs Zählschleifen höher als bei Modellen, die mit Motifs trainiert wurden. Auf den ersten

## 6 Experimente

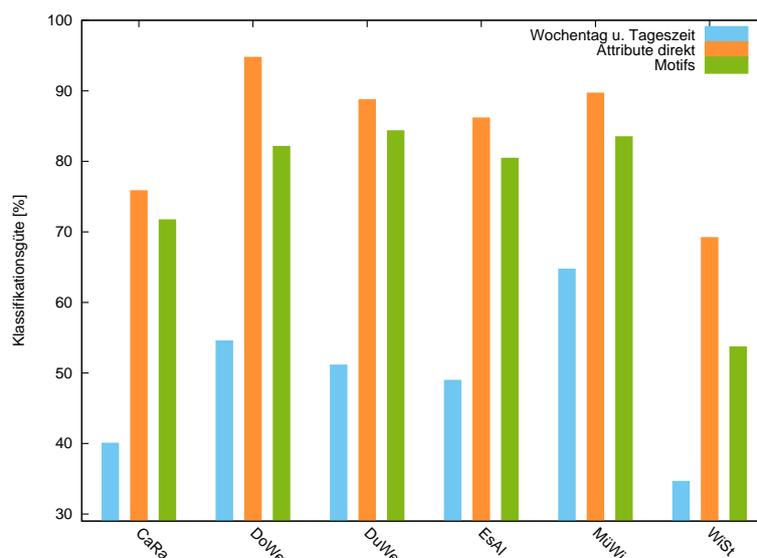


Abbildung 6.16: Die Klassifikationsgüte der drei unterschiedlichen Ansätze.

Blick scheint die in dieser Arbeit vorgestellten Repräsentation somit eine Abnahme der Modellgüte zu verursachen. Wird das Ergebnis in Form von Sensitivität und Genauigkeit für die unterschiedlichen Zustandsübergänge weiter aufgeschlüsselt muss diese Folgerung zumindest im Kontext des betrachteten Anwendungsfalls relativiert werden.

Tabelle 6.7: Die Tabelle beinhaltet Sensitivität und Genauigkeit der drei Ansätze für die einzelnen Zustandsübergänge.

		keine Motifs		Motifs
		Zeit	Attribute	
<i>grün</i> → <i>gelb</i>	Sensitivität	56,77	100	82,21
	Genauigkeit	56,48	98,44	81,39
<i>grün</i> → <i>rot</i>	Sensitivität	16,67	0	80,81
	Genauigkeit	20,00	0	60,34
<i>gelb</i> → <i>rot</i>	Sensitivität	33,33	23,33	79,74
	Genauigkeit	37,74	53,85	83,56
<i>gelb</i> → <i>grün</i>	Sensitivität	56,81	98,71	80,81
	Genauigkeit	55,88	94,58	83,61
<i>rot</i> → <i>grün</i>	Sensitivität	29,41	18,75	56,72
	Genauigkeit	38,46	37,50	64,41
<i>rot</i> → <i>gelb</i>	Sensitivität	33,33	87,72	84,56
	Genauigkeit	35,85	76,92	88,73

In Tabelle 6.7 werden Sensitivität und Genauigkeit der drei Modelle in Relation gesetzt. Zwar erreicht das Modell auf den Attributen der Zeitreihe ausgezeichnete Werte für die Klassen *grün* → *gelb* und *gelb* → *grün*, jedoch ist das Modelle mit einer Sensitivität von 23,33% bei 53,85% Genauigkeit für die Vorhersage von Transi-

tionen *gelb*  $\rightarrow$  *rot* weniger gut geeignet. Noch schlechter werden Zustandsübergänge vom freien Verkehr in den Stau vorhergesagt. Das Motif-Verfahren erreicht hier eine deutlich größere Vorhersagegüte. Diese Beobachtung trifft auch für die Ergebnisse der hier nicht näher betrachteten anderen fünf Zählschleifen zu.

Durch die Ergebnisse in Tabelle 6.7 bekräftigt sich auch die These, dass zeitliche Modelle für die Vorhersage von Zeiträumen mit hoher Verkehrsdichte besser geeignet sind als für Transitionen zwischen synchronem Fluss und Stau. Um die Güte hier besser abschätzen zu können, wurden die Datensätze noch einmal leicht abgewandelt. Anstatt der schwierig genau zu terminierenden Zustandsübergänge soll das zeitliche Modell nun die Verkehrsphase abhängig von Wochentag und Tageszeit bestimmen. Die Phasen synchroner Fluss und Stau werden hierzu vereinfacht zur Klasse *gestaut* zusammengefasst. Die Klasse *frei* steht dementsprechend für den freien Fluss.

Da jetzt die Phasen selbst und keine Transitionen zwischen Phasen betrachtet werden steigt die Zahl von Beispielen beträchtlich, da die meisten Beispiele eine Ausgabe von *grün*  $\rightarrow$  *grün* haben und bislang im Vorfeld aussortiert wurden. Die klassenübergreifende Klassifikationsgüte beträgt für alle sechs Fälle über 90% und ist wegen des hohen Anteils der Klasse *frei* nicht sonderlich aussagekräftig. Deshalb sind in Tabelle 6.8 die Konfusionsmatrizen für die sechs unterschiedlichen Datensätze gegenüber gestellt. Sensitivität und Genauigkeit für die interessantere Prognose *gestaut* liegen lediglich für die beiden Zählschleifen *CaRa* und *WiSt* um 80%. In den vier anderen Fällen sind die Stauzeiten an den entsprechenden Streckenabschnitten also weniger zuverlässig an Wochentage und Tageszeiten gebunden. Werden mit diesen Erkennt-

Tabelle 6.8: Konfusionsmatrizen für das zeitliche Modell zur Vorhersage der Verkehrsphase anhand von Tageszeit und Wochentag.

CaRa		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	15941	1033	93,91%
	<i>gestaut</i>	943	4074	81,20%
Sensitivität		94,41%	79,77%	

(a) CaRa

DoWe		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	53991	2773	95,11%
	<i>gestaut</i>	375	28	6,59%
Sensitivität		99,31%	1,00%	

(b) DoWe

DuWe		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	26187	1320	95,20%
	<i>gestaut</i>	694	430	38,26%
Sensitivität		97,42%	24,57%	

(c) DuWe

EsAl		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	59371	1855	96,97%
	<i>gestaut</i>	206	8	3,74%
Sensitivität		99,65%	0,43%	

(d) EsAl

MüWi		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	32429	536	98,37%
	<i>gestaut</i>	100	0	0,00%
Sensitivität		99,69%	0,00%	

(e) MüWi

WiSt		Y		Genauigkeit
		<i>frei</i>	<i>gestaut</i>	
$\hat{f}(X)$	<i>frei</i>	19136	695	96,50%
	<i>gestaut</i>	731	2644	78,34%
Sensitivität		96,32%	79,19%	

(f) WiSt

nissen erneut die Ergebnisse des Motif-Verfahrens bezüglich der sechs Datensätze in

## 6 Experimente

Abbildung 6.14 betrachtet fällt auf, dass die Klassifikationsgüte des Motif-Verfahrens immer dann hoch ist, wenn Sensitivität und Genauigkeit des zeitlichen Modells für die Klasse *gestaut* niedrig sind. Eine Repräsentation von Verkehrsdaten durch Motifs und Wortvektoren ist nach diesen Beobachtungen also in Situationen von Vorteil, in denen Staus nicht völlig regelmässig auftreten.

## Fazit und Ausblick

Diese Arbeit thematisiert die Vorhersage von Zustandsübergängen in Verkehrssystemen. Diesem Text übergeordnet ist dabei die Frage, ob sich Verfahren des maschinellen Lernens für diese Aufgabe eignen. Der Fokus lag dabei auf den Zustandsübergängen zwischen den drei Phasen freier Fluss, synchroner Fluss und sich bewegendem breiter Stau. Diese mehrklassige Lernaufgabe wurde am Beispiel des Autobahnnetzes im Ruhrgebiet untersucht.

Im Allgemeinen kann bestätigt werden, dass sich Verfahren des maschinellen Lernens für die Verarbeitung von Verkehrsdaten in Form von Zeitreihen eignen. Insbesondere die Verfahren Random Forests und Support Vector Machines eignen sich gut für die betrachtete Aufgabe. Mit  $k$  nächste Nachbarn hat sich jedoch auch ein sehr einfaches Modell bewährt. Vom Einsatz eines Naive Bayes Klassifikators oder neuronalen Netzes sollte bei der Vorhersage von Zustandsübergängen in Verkehrssystemen abgesehen werden. Ersteres Verfahren eignet sich nicht zur Vorhersage, da die verwendeten Apriori-Wahrscheinlichkeiten stark unbalanciert sind. Besonders für die verhältnismäßig seltenen Zustandsübergänge in die gestaute Phase wird keine zuverlässige Prognose erreicht. Gegen die Klassifikation mit neuronalen Netzen spricht die im Vergleich zu Support Vector Machines und Random Forests relativ schwierige und zeitintensive Parameteroptimierung dieses Verfahrens.

Es wurden zwei grundsätzliche Ansätze zur Vorhersage von Phasenübergängen verfolgt. Zum einen wurde die Regelmäßigkeit von hohen Verkehrsdichten zu den Stoßzeiten auf Autobahnen genutzt, um ein Modell zu erzeugen, welches Verkehrsstaus anhand von Wochentag und Tageszeit prognostiziert. Es hat sich gezeigt, dass solch ein zeitliches Modell nicht für die Prognose von Zustandsübergängen selbst geeignet ist und auch für die Vorhersage von, im Wesentlichen durch hohe Verkehrsdichten bestimmten, gestauten Zeiträumen nur unter besonderen Voraussetzungen gute Ergebnisse erzielt. Wenn solche Verkehrsmuster nicht zuverlässig zu festen Zeiten auftreten, ist ein zeitliches Modell nur schlecht für Prognosen geeignet, die der Verkehrsbeeinflussung dienen sollen.

Bei dem zweiten Ansatz wurden die Werte der erfassten Verkehrsgrößen berücksichtigt und deutlich bessere Ergebnisse erzielt. Bereits unter Verwendung der Rohdaten wurde eine hohe Klassifikationsgüte erreicht. Für einige Zustandsübergänge waren die Ergebnisse jedoch unbrauchbar. Diese Schwachstelle kann mit der hier vorgestellten Repräsentation behoben werden. So wurden vergleichsweise seltene Zustandsübergänge mit ähnlicher Güte wie sehr häufige Transitionen unter Verwendung der selben Lernverfahren klassifiziert.

Für die Repräsentation wurde das Prinzip von Motifs in Zeitreihen verwendet, um

charakteristische Sequenzen in den stark verrauschten Verkehrsdaten zu entdecken. Mit diesen Motifs war es anschließend möglich, Verfahren aus dem Text Mining zur Klassifikation von Dokumenten auf Verkehrsdaten anzuwenden. Dies hat den Vorteil, dass mithilfe spärlich besetzter Vektoren die Lernaufgabe in Räume mit prinzipiell beliebiger Dimension überführt werden kann, um so bessere Grenzen zwischen den einzelnen Klassen zu finden.

Durch einen Vergleich mit den Ergebnissen des zeitlichen Modells wurde festgestellt, dass sich das hier vorgestellte Verfahren insbesondere für Streckenabschnitte eignet, die weniger Regelmäßigkeiten im Tagesverlauf aufweisen. Es wurde auch gezeigt, dass die entwickelte Repräsentation sehr einfach um zusätzliche Informationen erweitert werden kann. Für einen Datensatz, dessen Beispiele um Informationen benachbarter Zählschleifen und zeitlichen Informationen angereichert wurden, ist es gelungen die Ergebnisse um 10% zu verbessern. Hier bietet sich besonders durch die Integration von Wetterdaten und Informationen zu wiederkehrenden Ereignissen mit hohem Verkehrsaufkommen wie z.B. Fußballspielen oder Ferienzeiten die Möglichkeit einer Verbesserung der Modellgüte. Dies liegt daran, dass Motifs unter der Prämisse erzeugt wurden, dass ähnliche Randbedingungen zu ähnlichen Teilfolgen in der Zeitreihe führen. Informationen zu solchen Randbedingungen sollten sich also positiv auf die Vorhersagefähigkeit des Modells auswirken.

Auch die Parametrisierung des Verfahrens bedarf weiterer Untersuchung. Wie in den entsprechenden Kapiteln und Abschnitten angedeutet, ist die Zahl an möglichen Parametrisierungen für die Extraktion von Motifs und Repräsentation des Datenstroms hoch. Eine händische Optimierung ist deshalb wenig aussichtsreich. Deshalb wurde in dieser Arbeit nur exemplarisch der Vorhersagehorizont untersucht. Insbesondere eine Untersuchung der Motiflänge in Abhängigkeit der Verkehrsgröße erscheint sinnvoll, da in der Literatur unterschiedliche Aggregationsintervalle für unterschiedliche Attribute verwendet werden.

Aber auch das Verfahren zur Extraktion von Motifs selbst kann weiter verbessert werden. Bislang ist die Ähnlichkeit von Teilfolgen rein über die euklidische Distanz der einzelnen Elemente definiert. Eine charakteristische Teilfolge muss jedoch nicht nur anhand der Werte selbst definiert werden. Auch die Form und der Verlauf einer Teilfolge kann als Eigenschaft verwendet werden. Denkbar sind z. B. die Zahl der Steigungswechsel oder die Differenz von Maximum und Minimum der Teilfolge. Dies ist auch ein Ansatzpunkt um globale oder prototypische Modelle zu erzeugen. Bislang sprechen die Ergebnisse der hier durchgeführten Experimente für eine Verwendung von lokalen Modellen für einzelne Zählschleifen. Unter diesem Aspekt muss jedoch auch noch untersucht werden, welche Merkmale von Zählschleifen geeignet sind, um Gruppen von Sensoren zu ermitteln, dessen Zeitreihen ähnliche Motifs enthalten.

Darüber hinaus muss geklärt werden, ob eine endliche Zahl von Motifs ausreicht, die über einen festen Zeitraum von z. B. einem Jahr erhoben wurde, um ein robustes Modell für die Zukunft zu erzeugen oder das Modell durch Verfahren wie *test then train* inkrementell aktualisiert werden muss. Die vorhandenen Daten umfassen lediglich sechs Monate und lassen somit keine konkreten Aussagen zu diesem Aspekt zu.

Bei der in dieser Arbeit vorgestellten Methode zur Vorhersage von Zustandsübergängen in Verkehrssystemen handelt es sich also um einen aussichtsreichen Ansatz mit großem Optimierungspotential. Auch wenn der Fokus dieser Arbeit auf dem Anwendungsfall der Verkehrsbeeinflussung lag und somit die kurzfristige Vorhersage von Transitionen zwischen freiem und synchronem Fluss sowie synchronem Fluss und Stau als Ziel hatte, deuteten sich an diversen Stellen weitere potentielle Anwendungsgebiete an. So erzielte die Erkennung von Transitionen in entspanntere Verkehrsphasen ähnliche Ergebnisse. Auch für einen längeren Vorhersagehorizont bis 30 Minuten wurde eine gute und zum Teil bessere Modellgüte erreicht. Dies eröffnet dem präsentierten Verfahren ein weites Feld von Anwendungen im Verkehrswesen wie z. B. der Bestimmung von Reisezeiten oder der intelligenten Navigation.



# Literaturverzeichnis

- [1] *Highway Capacity Manual: Metric Units*. Transportation Research Board, 2001. – ISBN 0309066816
- [2] AHMED, Mohamed S. ; COOK, Allen R.: Freeway traffic time series data using Box and Jenkins techniques. In: *Transportation Research Record 722* (1979), S. 1–9. – ISSN 0361–1981
- [3] ANDO, Tomohiro: *Bayesian Model Selection and Statistical Modeling (Statistics: A Series of Textbooks and Monographs)*. Chapman and Hall/CRC, 2010. – ISBN 1439836140
- [4] ARASU, A. ; MANKU, G.S.: Approximate counts and quantiles over sliding windows. In: *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* ACM, 2004, S. 286–296
- [5] AXELSSON, S.: The base-rate fallacy and the difficulty of intrusion detection. In: *ACM Transactions on Information and System Security (TISSEC)* 3 (2000), Nr. 3, S. 186–205
- [6] BABCOCK, B. ; BABU, S. ; DATAR, M. ; MOTWANI, R. ; WIDOM, J.: Models and issues in data stream systems. In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* ACM, 2002, S. 1–16
- [7] BARTH, Matthew ; BORIBOONSOMSIN, Kanok: Real-World CO2 Impacts of Traffic Congestion / University of California Transportation Center. 2008. – University of California Transportation Center, Working Papers
- [8] BREIMAN, L.: Random forests. In: *Machine learning* 45 (2001), Nr. 1, S. 5–32
- [9] BRETON, P. ; HEGYI, A. ; DE SCHUTTER, B. ; HELLENDORRN, H.: Shock wave elimination/reduction by optimal coordination of variable speed limits. In: *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on IEEE*, 2002, S. 225–230
- [10] BURKSCHAT, Marco ; CRAMER, Erhard ; KAMPS, Udo: *Beschreibende Statistik: Grundlegende Methoden der Datenanalyse (EMIL@A-stat) (German Edition)*. Springer Spektrum, 2012. – ISBN 978–3–642–30013–4
- [11] BVU, I.C.: Prognose der deutschlandweiten Verkehrsverflechtungen 2025. In: *München, Freiburg. Gutachten im Auftrag des BMVBS* (2007)
- [12] BYRNES, V.A.: Visual factors in automobile driving. In: *Transactions of the American Ophthalmological Society* 60 (1962), S. 60
- [13] CAPUTO, B. ; SIM, K. ; FURESJO, F. ; SMOLA, A.: Appearance-based Object Recognition using SVMs: Which Kernel Should I Use? In: *Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision*, Whistler Bd. 2002, 2002
- [14] CARUANA, R. ; NICULESCU-MIZIL, A.: An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd international conference on Machine learning* ACM, 2006, S. 161–168
- [15] CASTRO, N. ; AZEVEDO, P.: Multiresolution motif discovery in time series. In: *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, 2010, Columbus, Ohio, USA*, 2010, S. 665–676

- [16] CASTRO-NETO, Manoel ; JEONG, Young-Seon ; JEONG, Myong-Kee ; HAN, Lee D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. In: *Expert Syst. Appl.* 36 (2009), April, S. 6164–6173. – ISSN 0957–4174
- [17] CHEN, H. ; GRANT-MULLER, S. ; MUSSONE, L. ; MONTGOMERY, F.: A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. In: *Neural Computing & Applications* 10 (2001), Nr. 3, S. 277–286
- [18] CHROBOK, R. ; KAUMANN, O. ; WAHLE, J. ; SCHRECKENBERG, M.: Three categories of traffic data: Historical, current, and predictive. In: *Proceedings of the 9th IFAC Symposium Control in Transportation Systems*, 2000, S. 250–255
- [19] COCHRAN, William G.: *Stichprobenverfahren*. Gruyter, 1972. – ISBN 3110020408
- [20] COOLS, M. ; MOONS, E. ; WETS, G.: Assessing the impact of weather on traffic intensity. In: *Weather, Climate, and Society* 2 (2010), Nr. 1, S. 60–68
- [21] CORINNE ; LEDOUX: An urban traffic flow model integrating neural networks. In: *Transportation Research Part C: Emerging Technologies* 5 (1997), Nr. 5, S. 287 – 300. – ISSN 0968–090X
- [22] DARWIN, C.: *The origin of species*. Collier, 1937
- [23] DAS, M.K. ; DAI, H.K.: A survey of DNA motif finding algorithms. In: *BMC bioinformatics* 8 (2007), Nr. Suppl 7, S. S21
- [24] DAVIS, Gary A. ; NIHAN, Nancy L.: Nonparametric Regression and Short-Term Freeway Traffic Forecasting. In: *Journal of Transportation Engineering* 117 (1991), Nr. 2, S. 178
- [25] DING, AiLing ; ZHAO, XiangMo ; JIAO, LiCheng: Traffic flow time series prediction based on statistics learning theory. In: *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, 2002, S. 727 – 730
- [26] DUTZIK, Tony ; PREGULMAN, Robert: More Roads, More Traffic - Why Highway Construction Won't Solve Traffic Congestion in Washington / WashPIRG Foundation. 2003. – Report
- [27] EFRON, B. ; GONG, G.: A leisurely look at the bootstrap, the jackknife, and cross-validation. In: *American Statistician* (1983), S. 36–48
- [28] EL-GENEIDY, A.M. ; KRIZEK, K.J. ; IACONO, M.J.: Predicting bicycle travel speeds along different facilities using GPS data: a proof of concept model. In: *Proceedings of the 86th Annual Meeting of the Transportation Research Board, Compendium of Papers*, 2007
- [29] ESTRIN, D. ; GOVINDAN, R. ; HEIDEMANN, J. ; KUMAR, S.: Next century challenges: Scalable coordination in sensor networks. In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* ACM, 1999, S. 263–270
- [30] FAHRMEIR, L.: *Statistik: Der Weg zur Datenanalyse*. Springer, 2007
- [31] FEURTEY, F.: Simulating the collision avoidance behavior of pedestrians. In: *Master's Thesis* (2000)
- [32] FRANK, Eibe ; BOUCKAERT, Remco: Naive Bayes for Text Classification with Unbalanced Classes. In: FÜRNKRANZ, Johannes (Hrsg.) ; SCHEFFER, Tobias (Hrsg.) ; SPILIOPOULOU, Myra (Hrsg.): *Knowledge Discovery in Databases: PKDD 2006* Bd. 4213. Springer Berlin / Heidelberg, 2006. – ISBN 978–3–540–45374–1, S. 503–510
- [33] FRÖHLICH, H. ; ZELL, A.: Efficient parameter selection for support vector machines in classification and regression via model-based global optimization. In: *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on* Bd. 3 IEEE, 2005, S. 1431–1436

- [34] GABER, M.M. ; ZASLAVSKY, A. ; KRISHNASWAMY, S.: Mining data streams: a review. In: *ACM Sigmod Record* 34 (2005), Nr. 2, S. 18–26
- [35] GEISTEFELD, Justin ; LOHHOFF, Jan: Stausituation auf den Autobahnen in Nordrhein-Westfalen / Ministerium für Wirtschaft, Energie, Bauen, Wohnen und Verkehr des Landes Nordrhein-Westfalen. 2011. – Studie
- [36] GEURTS, P.: Contributions to decision tree induction: bias/variance tradeoff and time series classification. In: *Liege, Belgium: University of Liege* 11 (2002)
- [37] GOTTHOLMSEDER, Georg ; NOWOTNY, Klaus ; PRUCKNER, Gerald J. ; THEURL, Engelbert: Stress perception and commuting. In: *Health Economics* 18 (2009), Nr. 5, S. 559–576
- [38] GREENWALD, M. ; KHANNA, S.: Space-efficient online computation of quantile summaries. In: *ACM SIGMOD Record* Bd. 30 ACM, 2001, S. 58–66
- [39] HAN, Jiawei ; KAMBER, Micheline ; PEI, Jian: *Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 2011. – ISBN 0123814790
- [40] HASSANI, M. ; MÜLLER, E. ; SEIDL, T.: EDISKCO: energy efficient distributed in-sensor-network k-center clustering with outliers. In: *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data* ACM, 2009, S. 39–48
- [41] HASTIE, Trevor ; TIBSHIRANI, Robert ; FRIEDMAN, Jerome: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. 2009. Corr. 3rd printing 5th Printing. Springer, 2009 (Springer Series in Statistics). <http://www-stat.stanford.edu/~tibs/ElemStatLearn/main.html>. – ISBN 0387848576
- [42] HATHAWAY, D. H. ; WILSON, R. M. ; REICHMANN, E. J.: The shape of the sunspot cycle. In: *Solar Physics* 151 (1994), April, S. 177–190
- [43] HELBING, Dirk: *Verkehrsdynamik: Neue physikalische Modellierungskonzepte (German Edition)*. 1. Springer, 1997. – ISBN 3540619275
- [44] HERTZ, John A. ; PALMER, Richard G. ; KROGH, Anders: *Introduction to the Theory of Neural Computation (Santa Fe Institute Studies in the Sciences of Complexity)*. Addison Wesley Publishing Company, 1991. – ISBN 0201503956
- [45] HOOGENDOORN, Serge P. ; BOVY, Piet H. L.: State-of-the-art of Vehicular Traffic Flow Modelling. In: *Delft University of Technology, Delft, The*, 2001, S. 283–303
- [46] HORVITZ, Eric ; APACIBLE, Johnson ; SARIN, Raman ; LIAO, Lin: Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. In: *In Twenty-First Conference on Uncertainty in Artificial Intelligence*, 2005
- [47] HSU, C.W. ; CHANG, C.C. ; LIN, C.J. u. a.: *A practical guide to support vector classification*. 2003
- [48] HSU, C.W. ; LIN, C.J.: A comparison of methods for multiclass support vector machines. In: *Neural Networks, IEEE Transactions on* 13 (2002), Nr. 2, S. 415–425
- [49] HUGUENEY, B.: Adaptive segmentation-based symbolic representations of time series for better modeling and lower bounding distance measures. In: *Knowledge Discovery in Databases: PKDD 2006* (2006), S. 545–552
- [50] IOERGER, Thomas R. ; MEEKS, John H. ; NELSON, Paul: *Investigation of Density and Flow Relationships in Congested Traffic Using Videogrammetric Data*. 2001
- [51] ISHAK, S. ; KOTHA, P. ; ALECSANDRU, C.: Optimization of dynamic neural network performance for short-term traffic prediction. In: *Transportation Research Record: Journal of the Transportation Research Board* 1836 (2003), Nr. -1, S. 45–56

- [52] JAWORSKI, Wojciech: Model Selection and Assessment for Classification Using Validation. In: SLEZAK, Dominik (Hrsg.) ; WANG, Guoyin (Hrsg.) ; SZCZUKA, Marcin (Hrsg.) ; DÜNTSCH, Ivo (Hrsg.) ; YAO, Yiyu (Hrsg.): *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* Bd. 3641. Springer Berlin / Heidelberg, 2005, S. 481–490
- [53] JOACHIMS, T.: Text categorization with support vector machines: Learning with many relevant features. In: *Machine Learning: ECML-98* (1998), S. 137–142
- [54] KANTETY, R.V. ; LA ROTA, M. ; MATTHEWS, D.E. ; SORRELLS, M.E.: Data mining for simple sequence repeats in expressed sequence tags from barley, maize, rice, sorghum and wheat. In: *Plant molecular biology* 48 (2002), Nr. 5, S. 501–510
- [55] KEAY, K. ; SIMMONDS, I.: The association of rainfall and other weather variables with road traffic volume in Melbourne, Australia. In: *Accident Analysis & Prevention* 37 (2005), Nr. 1, S. 109–124
- [56] KEERTHI, S.S. ; LIN, C.J.: Asymptotic behaviors of support vector machines with Gaussian kernel. In: *Neural computation* 15 (2003), Nr. 7, S. 1667–1689
- [57] KEOGH, E. ; LIN, J. ; FU, A.: Hot sax: Efficiently finding the most unusual time series subsequence. In: *Data Mining, Fifth IEEE International Conference on Ieee*, 2005, S. 8–pp
- [58] KERNER, B.S.: Experimental Features of Self-Organization in Traffic Flow. In: *Phys. Rev. Lett.* 81 (1998), Oct, S. 3797–3800
- [59] KERNER, B.S.: Three-phase traffic theory and highway capacity. In: *Physica A: Statistical Mechanics and its Applications* 333 (2004), S. 379–440
- [60] KERNER, B.S.: *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory*. Springer, 2009. – ISBN 9783642026041
- [61] KIRBY, Howard R. ; WATSON, Susan M. ; DOUGHERTY, Mark S.: Should we use neural networks or statistical models for short-term motorway traffic forecasting? In: *International Journal of Forecasting* 13 (1997), Nr. 1, S. 43 – 50. – ISSN 0169–2070
- [62] KREISS, J.P. ; NEUHAUS, G.: *Einführung in Die Zeitreihenanalyse*. Springer, 2006 (Statistik Und Ihre Anwendungen). – ISBN 9783540256281
- [63] LERMAN, P. M.: Fitting Segmented Regression Models by Grid Search. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 29 (1980), Nr. 1, S. pp. 77–84. – ISSN 00359254
- [64] LIN, J. ; KEOGH, E. ; WEI, L. ; LONARDI, S.: Experiencing SAX: a novel symbolic representation of time series. In: *Data Mining and Knowledge Discovery* 15 (2007), Nr. 2, S. 107–144
- [65] LIN, Wei-Hua: A Gaussian maximum likelihood formulation for short-term forecasting of traffic flow. In: *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 2001, S. 150 –155
- [66] LIU, X. ; BRUTLAG, D.L. ; LIU, J.S. u. a.: BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In: *Pac Symp Biocomput* Bd. 6, 2001, S. 127–138
- [67] LKHAGVA, B. ; SUZUKI, Y. ; KAWAGOE, K.: New time series data representation ESAX for financial applications. In: *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on IEEE*, 2006, S. x115–x115
- [68] LONARDI, S. ; LIN, J. ; KEOGH, E. ; PATEL, P.: Finding motifs in time series. In: *Proc. of the 2nd Workshop on Temporal Data Mining*, 2002, S. 53–68

- [69] MAZE, T.H. ; AGARWAI, M. ; BURCHETT, G.: Whether weather matters to traffic demand, traffic safety, and traffic operations and flow. In: *Transportation Research Record: Journal of the Transportation Research Board* 1948 (2006), Nr. -1, S. 170–176
- [70] METWALLY, A. ; AGRAWAL, D. ; EL ABBADI, A.: Efficient computation of frequent and top-k elements in data streams. In: *Database Theory-ICDT 2005* (2005), S. 398–412
- [71] MIERSWA, I. ; MORIK, K.: Automatic feature extraction for classifying audio data. In: *Machine learning* 58 (2005), Nr. 2, S. 127–149
- [72] MINNEN, D. ; STARNER, T. ; ESSA, I. ; ISBELL, C.: Improving activity discovery with automatic neighborhood estimation. In: *Int. Joint Conf. on Artificial Intelligence*, 2007, S. 6–12
- [73] MÖRCHEN, Fabian ; ULTSCH, Alfred: Optimizing time series discretization for knowledge discovery. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA : ACM, 2005 (KDD '05). – ISBN 1-59593-135-X, S. 660–665
- [74] MUEEN, A. ; KEOGH, E.: Online discovery and maintenance of time series motifs. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, 2010, S. 1089–1098
- [75] MUEEN, A. ; KEOGH, E. ; ZHU, Q. ; CASH, S. ; WESTOVER, B.: Exact discovery of time series motifs. In: *Proc. of 2009 SIAM International Conference on Data Mining: SDM*, 2009, S. 1–12
- [76] MUÑOZ, J.C. ; DAGANZO, C.F.: Structure of the Transition Zone Behind Freeway Queues. In: *Transportation Science* 37 (2003), Nr. 3, S. 312–329
- [77] NAGEL, Kai ; SCHRECKENBERG, Michael: A cellular automaton model for freeway traffic. In: *Journal de Physique I* 2 (1992), Dezember, Nr. 12, S. 2221–2229. <http://dx.doi.org/10.1051/jp1:1992277>. – DOI 10.1051/jp1:1992277. – ISSN 1155-4304
- [78] OH, C. ; RITCHIE, S.G. ; OH, J.S.: Exploring the relationship between data aggregation and predictability to provide better predictive traffic information. In: *Transportation Research Record: Journal of the Transportation Research Board* 1935 (2005), Nr. -1, S. 28–36
- [79] PARK, D.P.D. ; YOU, S.Y.S. ; RHO, J.R.J. ; CHO, H.C.H. ; LEE, K.L.K.: Investigating optimal aggregation interval sizes of loop detector data for freeway travel-time estimation and prediction. In: *Canadian Journal of Civil Engineering* 36 (2009), Nr. 4, S. 580–591
- [80] PATROUMPAS, Kostas ; SELLIS, Timos: Window Specification over Data Streams. In: GRUST, Torsten (Hrsg.) ; HÖPFNER, Hagen (Hrsg.) ; ILLARRAMENDI, Arantza (Hrsg.) ; JABLONSKI, Stefan (Hrsg.) ; MESITI, Marco (Hrsg.) ; MÜLLER, Sascha (Hrsg.) ; PATRANJAN, Paula-Lavinia (Hrsg.) ; SATTLER, Kai-Uwe (Hrsg.) ; SPILIOPOULOU, Myra (Hrsg.) ; WIJSEN, Jef (Hrsg.): *Current Trends in Database Technology - EDBT 2006* Bd. 4254. Springer Berlin / Heidelberg, 2006. – ISBN 978-3-540-46788-5, S. 445–464
- [81] QIAO, F. ; WANG, X. ; YU, L.: Optimizing aggregation level for ITS data based on wavelet decomposition. In: *82nd Transportation Research Board Annual Meeting, National Research Council*, 2003
- [82] RAO, C.R. ; DEY, Dipak K.: *Essential Bayesian Models*. North Holland, 2011. – ISBN 0444537325
- [83] RIPLEY, Brian D.: *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996. – ISBN 0521460867
- [84] RISH, I.: An empirical study of the naive Bayes classifier. In: *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence* Bd. 3, 2001, S. 41–46

- [85] SCHLITZGEN, R. ; STREITBERG, B.H.J.: *Zeitreihenanalyse*. Oldenbourg Wissenschaftsverlag, 2001
- [86] SCHÖLKOPF, Bernhard ; SMOLA, Alexander J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, 2001. – ISBN 0262194759
- [87] SHEFER, Daniel ; RIETVELD, Piet: Congestion and Safety on Highways: Towards an Analytical Model. In: *Urban Studies (Routledge)* 34 (1997), Nr. 4, S. 679 – 692. – ISSN 00420980
- [88] SHIEH, J. ; KEOGH, E.: iSAX: indexing and mining terabyte sized time series. In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, 2008, S. 623–631
- [89] SMITH, B.L. ; OSWALD, R.K.: Effects of parameter selection on forecast accuracy and execution time in nonparametric regression. In: *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, 2000, S. 252 –257
- [90] SOM, R.K. u. a.: *A manual of sampling techniques*. Heinemann Educational Books Ltd., 1973. – ISBN 0435538659
- [91] SUN, Shiliang ; YU, Guoqiang ; ZHANG, Changshui: Short-term traffic flow forecasting using Sampling Markov Chain method with incomplete data. In: *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, S. 437 – 441
- [92] THOMAS, I.: Spatial data aggregation: exploratory analysis of road accidents. In: *Accident Analysis & Prevention* 28 (1996), Nr. 2, S. 251–264
- [93] TINNEY, W.F. ; BRANDWAIN, V. ; CHAN, S.M.: Sparse Vector Methods. In: *Power Apparatus and Systems, IEEE Transactions on PAS-104* (1985), feb., Nr. 2, S. 295 –301. – ISSN 0018–9510
- [94] TREIBER, Martin ; KESTING, Arne: *Verkehrsdynamik und -simulation: Daten, Modelle und Anwendungen der Verkehrsflussdynamik (Springer-Lehrbuch) (German Edition)*. 1st Edition. Springer, 2010. – ISBN 3642052274
- [95] TU, Jack V.: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. In: *Journal of Clinical Epidemiology* 49 (1996), Nr. 11, S. 1225 – 1231. – ISSN 0895–4356
- [96] VAPNIK, Vladimir: *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 2010. – ISBN 1441931600
- [97] VLAHOIANNI, Eleni I. ; GOLIAS, John C. ; KARLAFTIS, Matthew G.: Short-term traffic forecasting: Overview of objectives and methods. In: *Transport Reviews: A Transnational Transdisciplinary Journal* 24 (2004), Nr. 5, S. 533–557
- [98] VOORT, Mascha Van D. ; DOUGHERTY, Mark ; WATSON, Susan: Combining kohonen maps with arima time series models to forecast traffic flow. In: *Transportation Research Part C: Emerging Technologies* 4 (1996), Nr. 5, S. 307 – 318. – ISSN 0968–090X
- [99] WEICKER, Karsten: *Evolutionäre Algorithmen*. Teubner, 2002. – ISBN 3519003627
- [100] WEISS, S.M.: *Text mining: predictive methods for analyzing unstructured information*. Springer-Verlag New York Inc, 2005
- [101] WILLIAMS, Billy M. ; HOEL, Lester A.: Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. In: *Journal of Transportation Engineering* 129 (2003), Nr. 6, S. 664–672
- [102] WITTEN, Ian H. ; EIBE, Frank: *Data Mining*. Hanser Fachbuch, 2001. – ISBN 3446215336

- [103] YASDI, R.: Prediction of road traffic using a neural network approach. In: *Neural computing & applications* 8 (1999), Nr. 2, S. 135–142
- [104] YU, Guoqiang ; HU, Jianming ; ZHANG, Changshui ; ZHUANG, Like ; SONG, Jingyan: Short-term traffic flow forecasting based on Markov chain model. In: *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003, S. 208 – 212
- [105] ZHANG, H.M.: A mathematical theory of traffic hysteresis. In: *Transportation Research Part B: Methodological* 33 (1999), Nr. 1, S. 1–23
- [106] ZHONG, N. ; LIU, J. ; YAO, Y.: *Web intelligence*. Springer-Verlag New York Inc, 2003
- [107] ZOU, Xi ; LEVINSON, David: Detecting the Breakdown of Traffic / University of Minnesota: Nexus Research Group. 2006 (000034). – Working Papers



# Index

- Überanpassung, 17
- Übereinstimmung, 33
  
- Anzahl Fahrzeuge, 28
- Ausgaben, 32
  
- Baum, 59
- Beispielmenge, 12, 15, 32
- Belegung, 22, 29
  
- Clustering, 51–53
  
- Datensatz, 55
- Datenstrom, 13–14
- Dokument, 46
- Drei-Phasen-Verkehrstheorie, 23, 31
  
- Engstelle, 25
- euklidische Distanz, 39
- Euklidischer Abstand, 11
- evolutionäre Algorithmen, 20
  
- Fahrer-Fahrzeug-Einheit, 20
- Fahrzeugzahl, 28
- Fenster, 14, 41
- Fensterung, 37
- Fluch der hohen Dimension, 17
- Fluss-Dichte-Beziehung, 22
- freier Fluss, 23, 31
- Fundamentaldiagramm, 31
  
- Genauigkeit, 18
- Geschwindigkeit, 21, 29
  
- Highway Capacity Manual, 23
  
- Induktionsschleife, *siehe* Zählschleife
  
- k nächste Nachbarn, *siehe* kNN
- Kandidatenmenge, 34
  - Motifs, 42
  - notwendiges Kriterium, 40
- Kernfunktion, 63
- Klassen, 32
- Klassifikation, 15
  
- kNN, 58, 67
- Konfusionsmatrix, 18
- Kreuzvalidierung, 18
  
- Lernaufgabe, 3, 15
- level-of-service, 23
  
- makroskopische Modelle, 2–3, 21
  - Einsatzbereiche, 21
- maschinelles Lernen, 15–20
  - überwacht, 16
  - träges Lernen, 58
  - unüberwacht, 16
- Messquerschnitt, 27
- Metrik, 11
- mikroskopische Modelle, 2, 20
- Mittelwert, 37
- Modell, 15
- Motif, 33–36, 47
  - Algorithmus, 36–42
  - Extraktion, 42
  - notwendiges Kriterium, 34
  - Ursprung, 35
  - Verfahren, 35
  
- Nagel-Schreckenberg-Modell, 21
- Naive Bayes, 57, 67
- Neuronale Netze, 65, 67
  
- Online-Algorithmus, 14
- Ordnungsstrahl, 39–40
  - Referenzvektor, 39
  
- Parameter
  - kNN, 58
  - Motifs, 42–44
  - Neuronale Netze, 66
  - Random Forests, 61
  - Repräsentation, 49
  - SVM, 63
- Parameteroptimierung, 19–20
  - evolutionäre Algorithmen, 19
  - Rasteroptimierung, 19
- Phase, *siehe* Verkehrsphase

## Index

- Phasenübergang, *siehe* Zustandsübergang
- Quantil, 38
- Quantile, 10–11
- Random Forests, 59, 67
- RapidMiner, 51
  - Operator, 51, 56, 58, 61, 63, 66
  - Prozess, 56
- Rauschen, 37
- Regression, 15
- Repräsentation, 45–49
- SAX, 35
- Schlüsselwort, 46
- Schockwelle, 24–26, 32
- Sensitivität, 18
- Sensor, 27
- Sensornetzwerk, 13, 27
- sich bewegender breiter Stau, *siehe* Stau
- Signifikanz, 34
- Standardisierung, 12, 53
- Stau, 24, 25, 31
  - Folgen, 1
  - Gegenmaßnahmen, 1
- Staufront, 22, 24
- Stichprobe, 12
- Stratifikation, 12
- stromabwärtig, 22
- stromaufwärtig, 22
- SVM, 61, 67
- synchroner Fluss, 24, 31
- Teilfolge, 33
  - Übereinstimmung, 38–42
  - beste Übereinstimmung, 41
  - interessante Teilfolge, 38
- Testfehler, 18
- Testmenge, 12, 16
- Testphase, 16
- Text Mining, 46
- TF\*IDF, 48–49
- Trainingsmenge, 12, 15, 55
- Trainingsphase, 15
- Transition, *siehe* Zustandsübergang
- Vektor
  - spärlich besetzt, 47
- Verkehrsaufkommen, 1
- Verkehrsbeeinflussung, 1–2
- Verkehrsdichte, 21, 30
- Verkehrsdynamik, 20–26
- Verkehrsfluss, 21, 22, 30
- Verkehrsgröße, 28, 30
- Verkehrsgrößen, 21
- Verkehrsmuster, 22
- Verkehrsphase, 22–26
  - Bestimmung, 31
- Verkehrssystem, 20
- Verkehrsvorhersage, 3
- Verkehrszusammenbruch, 25, 32
- Verlustfunktion, 16
- Vorhersagegüte, 16–19
- Vorhersagehorizont, 71
- Wörterbuch, 46
- Wort, 46
- Wortvektor, 46–48
- Wortvektorrepräsentation, *siehe* Repräsentation
- z-Transformation, *siehe* Standardisierung
- Zählschleife, 27, 29, 55
- Zeitreihe, 7–10
  - Definition, 7
  - klassisches Komponentenmodell, 7
  - Rauschen, 10
  - Saison, 9
  - Trend, 8
  - Zyklus, 9
- Zustandsübergänge, 32
- Zustandsübergang, 24
  - Ursachen, 25