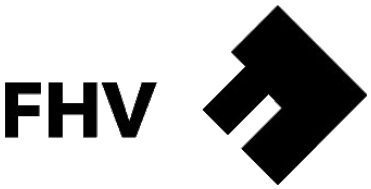
Fachhochschule Vorarlberg

University of Applied Sciences



Masterthesis

Fachhochschul-Studiengang

Master Informatik

gendering

Implementierung eines Verfahrens zur automatisierten Verkehrsmittelerkennung

Transportation Mode Detection

ausgeführt von

Michael Zangerle, BSc

1310249004

zur Erlangung des akademischen Grades

Master of Science in Engineering, MSc

Dornbirn, im Juli 2015

Betreuer: Prof. (FH) DI Thomas Feilhauer

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich vorliegende Masterthesis selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dornbirn, am 29. Juli 2015 Michael Zangerle, BSc

Zusammenfassung

Zusammen
ergänzen

Abstract

Abstract e

zen

Inhaltsverzeichnis

Todo list

Einleitung

Der Mensch produziert täglich eine extrem große Menge an Daten. Allein auf Youtube werden pro Minute 300 Stunden Video-Material veröffentlicht und täglich hunderte Millionen Stunden von Videos konsumiert. [Youtube, 2015] Hochgerechnet auf das gesamte Internet und die gesamte Bevölkerung ergibt dies eine unvorstellbar große Menge an Daten, die bewusst oder auch unbewusst generiert werden.

Sehr viel an Daten wird auch durch diverse Fitnessgadgets, Smartwatches, Smartphones sowie Navigations-Geräten und Ähnlichem generiert. Abseits von Fitnesswerten sind viele dieser Geräte im Regelfall in der Lage GPS-Spuren aufzuzeichnen. Dies bedeutet, dass man genau nachvollziehen kann, wann man wo unterwegs war. Mit ein wenig Rechenarbeit kann man auch die Geschwindigkeit und viele andere Werte berechnen, sofern dies die Geräte nicht schon selbst machen.

Genau auf diesen GPS-Daten basiert diese Arbeit. Dabei ist es nicht wichtig, von welcher Person diese Daten stammen, sondern dass sich mit Hilfe dieser aufgezeichneten Daten feststellen lässt, wann ein Individuum sich auf welcher Strecke mit welchem Verkehrsmittel fortbewegt hat.

Analysiert man viele dieser Daten, so lassen sich viele Erkenntnisse daraus gewinnen. Unter anderem lassen sich zum Beispiel viel frequentierte Strecken in der Infrastruktur

finden und mögliche Engstellen erkennen. Neben Engstellen lassen sich damit auch mögliche Verbesserungen, Einsparungen oder auch verstecktes Potential für zukünftige Projekte entdecken.

1.1 Ziele dieser Arbeit

Das Hauptziel dieser Arbeit ist es, einen Prototypen zu erstellen, welcher anhand von aufgezeichneten GPS-Spuren und in Kombination mit verschieden Methodiken das benutzte Verkehrsmittel mit einer möglichst hohen Wahrscheinlichkeit bestimmt. Diese aufgezeichneten GPS-Spuren enthalten dabei keinerlei Informationen über die jeweilige Person. Deshalb erfolgt die Auswertung ausschließlich über die jeweilige GPS-Spur sowie über öffentlich zugängliche Daten, wie zum Beispiel Busstationen und Gleise. Jede dieser Aufzeichnung kann mehrere Verkehrsmittel beinhalten und von unterschiedlicher Länge und Dauer sein.

Die in dieser Arbeit berücksichtigten Verkehrsmittel sind in folgender Liste ersichtlich. Besonders interessant ist hierbei die Unterscheidung von Bus und Auto in verkehrsabhängigen Situationen bzw. in der Stadt, da diese Transporttypen in diesen Situationen sehr ähnliche Verhalten und Werte aufweisen.

- Fußgänger
- Fahrrad
- Bus
- Auto (stellvertretend für Motorrad, Taxi, PKW etc.)
- Zug

Es soll weiters, für jede Person, die ein Gerät besitzt, das in der Lage ist, eine GPS-Spur im GPX-Format aufzuzeichnen, möglich sein, diese Spur analysieren zu lassen. Dies bedeutet, dass keine speziellen Geräte oder andere Sensoren benötigt werden und dass sich dieser Prototyp mit möglichst geringen Anpassungen (Trainingsdaten, GIS-Daten

und Grenzwerte in der Konfiguration) auch auf andere Regionen anwenden lässt. Zum Aufzeichnen der in dieser Arbeit verwendeten GPS-Spuren, wurden mehrere Smartphones mit der App “MyTrack“ sowie mehrere GPS-Geräte benutzt.

Im Zuge dieser Arbeit wird auch untersucht, welcher Grad an Genauigkeit sowohl mit als auch ohne geografische Zusatzinformationen im Raum Vorarlberg erreicht werden kann. Dabei soll es nach der Analyse die Möglichkeit geben, die automatisch bestimmten Verkehrsmittel manuell zu korrigieren, sollten diese nicht mit der Realität übereinstimmen. Weiters sollen diese manuellen Änderungen in die Auswertung mit einfließen.

Schlussendlich soll mit Hilfe der Auswertungen auch eine Aussage über die erzielte Genauigkeit mit und ohne den verwendeten Zusatzinformationen gemacht werden. Außerdem soll eine Aussage darüber gemacht werden, ob weitere Informationen (wie z.B. GPS-Daten der öffentlichen Verkehrsmittel) für eine noch genauere Bestimmung benötigt werden und welche Informationen dies sein könnten.

Nichtziele

In dieser Arbeit werden die diversen Schlussfolgerungsmodelle (neuronales Netz, Bayessches Netz, Random Forest, Support Vector Machine, ...) nicht betrachtet oder verglichen, sondern es wird auf ein Modell gesetzt, das bereits bei anderen Arbeiten wie z.B. [Stenneth et al., 2011], [Reddy et al., 2010], [Sebastian Nagel, 2011] und [Zheng et al., 2008b] vielversprechende Ergebnisse erzielt hat. Dieses Modell ist der Entscheidungsbaum. Außerdem werden die Daten zur Analyse bzw. Auswertung nicht in Echtzeit betrachtet sondern in Form einer GPS-Spur an den Prototypen übergeben.

Die Frage, an welcher Position man das Gerät zur Aufzeichnung am besten trägt, um möglichst genaue GPS-Daten zu erhalten, wird nicht weiter verfolgt, da die Daten möglichst realistisch sein sollen. Auch bleibt die Frage nach dem Energieverbrauch der App bzw. wie eine möglichst energieschonende App und die dazugehörige Kommunikation

aufgebaut sein könnten, unberücksichtigt. Schlussendlich ist eine umfassende Behandlung der Themen Sicherheit und Privatsphäre im Rahmen der vorliegenden Arbeit nicht möglich und daher bleiben auch diese Themen unberührt. An dieser Stelle soll allerdings erwähnt werden, dass keine benutzerspezifischen Information in den GPS-Spuren benötigt oder vom Prototypen generiert oder gespeichert werden.

1.2 Motivation und Nutzen

Schon früher wurde versucht, Aufzeichnungen über die Verkehrswege von verschiedenen Menschen zu sammeln. Aber die Protokolle in Papierform sowie die Telefonbefragungen waren zu aufwändig und die Menschen nicht zuverlässig genug. Darum ist es von entscheidendem Vorteil, eine App oder ein Gerät zur Verfügung zu haben, welches die Vorgänge des Aufzeichnens möglichst genau für einen übernimmt. [Zheng et al., 2010]

Wird eine Auswertung mit einer für das Zielgebiet aussagekräftigen Anzahl an Personen durchgeführt, so kann das Resultat für verschiedenste Zwecke verwendet werden. Auch ohne spezielle Analyse kann rein durch die Betrachtung der gesammelten GPS-Spuren festgestellt werden, welche Routen besonders häufig benutzt werden.

Zieht man nun verschiedene Werte aus der Auswertung hinzu, kann auch festgestellt werden, wo sich zum Beispiel verkehrstechnische Engstellen befinden und welche Routen sehr populär sind oder es können Aussagen über die allgemeine Verkehrssituation gemacht werden. Durch die gesammelten Daten könnten sich auch Simulationen für anstehende Bauvorhaben machen lassen und auch versucht werden, eine Vorhersage für bestimmte Situationen zu tätigen. Diese Aspekte können unter anderem für das Verkehrsministerium, den öffentlichen Personennahverkehr oder auch für die Stadtplanung sehr interessant sein (Optimierung von Auslastung, Einsparungspotentiale, ...).

Eine ganze Reihe von Apps lässt sich mit den Auswertungen erstellen. Unter anderem könnten diese Apps die Auswertungen in soziale Medien integrieren oder für Fitnessanalysen verwendet werden. Einen einfachen Rückblick über die eigene Fortbewegung kann man damit genauso ermöglichen wie für Umweltbewusste errechnen, wie viel CO_2 sie produziert oder gespart haben. Ein Reisetagebuch könnte daraus genauso Nutzen ziehen wie eine App (sollten die Daten in Echtzeit ausgewertet werden), die beim Autofahren Auskunft über die aktuell billigste Tankstelle in näherer Umgebung gibt oder eine App, die einfach nur Vorschläge für alternative, schnellere Routen zu einem bekannten Ziel anbietet .

Zusammenfassend kann man sagen, dass die Verwendungsmöglichkeiten für solche Daten umfangreich sind und sich am besten unter den Begriffen kontextorientierte, geographische Apps zusammenfassen lassen. Nicht zuletzt öffnen sich mit solchen Daten aber auch umfangreiche Möglichkeiten für die Werbebranche.

1.3 Weiterer Aufbau der Arbeit

Der Hauptteil der vorliegenden Arbeit gliedert sich in fünf große Abschnitte:

Im ersten Abschnitt wird auf die Akquirierung der GPS-Daten eingegangen. Dies umfasst sowohl die gesammelt GPS-Spuren und deren Struktur, sowie die verwendeten GIS-Daten. Dabei geht es einerseits um deren Herkunft als auch darum, wie diese extrahiert wurden und wie auch diese Daten aufgebaut sind. Außerdem werden auch andere Daten, wie zum Beispiel GPS-Daten von Bussen des ÖPNV in Betracht gezogen.

Der zweite Abschnitt behandelt den entwickelten Prototypen, der die übergebenen GPS-Spuren analysiert. Dabei werden einerseits dessen Funktionalitäten erklärt sowie der grundlegende Ablauf für den Benutzer/die Benutzerin dargelegt. Weiters wird auch auf die Architektur des Prototyps sowie auf dessen Konfigurationsmöglichkeiten eingegangen.

Der dritte Abschnitt befasst sich sowohl mit dem Einbinden der GIS-Daten als auch dem Aufbereiten der GPS-Daten sowie dem Bestimmen der Verkehrsmittel. Mit Aufbereiten der Daten ist gemeint, dass zusätzliche Werte zu GPS-Punkten für die spätere Analyse berechnet werden. Beispiele für diese Werte sind sowohl die Geschwindigkeit, Beschleunigung und Distanz als auch der Abstand zu der nächsten Bushaltestelle.

Weiters wird im dritten Abschnitt auch der Prozess des Aufteilens von GPS-Spuren in Teile (Segmente), in denen nur ein Verkehrsmittel verwendet wird, beschrieben. Diesem Schritt vorangegangen ist das in Anhang 1 beschriebene Filtern der GPS-Daten. Dies bedeutet in diesem Zusammenhang, dass ein Teil der fehlerhaften Ausreißer aus den GSP-Spuren entfernt wurden.

Aufbauend auf den Segmenten wird schließlich der Prozess der tatsächlichen Erkennung der Verkehrsmittel mit Hilfe eines Entscheidungsbaums und der berechneten Werte erklärt. Die aus dem Entscheidungsbaum gewonnenen Erkenntnisse werden schlussendlich ein letztes Mal überprüft, um sinnfreie bzw. sehr unwahrscheinliche Wechsel zwischen Verkehrsmitteln zu verhindern.

Der fünfte und letzte Abschnitt des Hauptteils befasst sich mit der Auswertung der gewonnenen Erkenntnisse aus den vorhergehenden Abschnitten sowie den Testläufen mit neuen GPS-Spuren mit und ohne zusätzliche GIS-Informationen.

State of the Art

Zu den Meilensteinen auf dem Forschungsgebiet der Verkehrsmittelerkennung (Transport Mode Detection) zählt die Arbeit von Yu Zheng in welcher er unter anderem auf die Erkennung von den Abschnitten mit nur einem Verkehrsmittel eingeht. Weiters verwendete er in seiner Arbeit auch einen mit dem von den GPS-Spuren gesammelten geographischen Wissen aufgebauten Graphen, welcher zur abschließenden Auswertung verwendet wurde. [Zheng et al., 2010] [Zheng et al., 2008a] [Zheng et al., 2008b]

Mit der Frage, wie geographische Daten in eine solche Analyse miteinbezogen werden können hat sich unter anderem Leon Stenneth beschäftigt. Dabei hat er nicht nur fixe Daten wie Gleise und Busstationen, sondern auch aktuelle Buspositionen miteinbezogen. [Stenneth et al., 2011]

Sowohl Stenneth als auch Zheng haben in ihren Arbeiten detailliert erklärt, wieso sie welche Attribute (Geschwindigkeit, Beschleunigung, ...) für die Bestimmung des Verkehrsmittels verwendet haben und sie haben diese auch durch Versuche nach ihrer Wichtigkeit gereiht. Außerdem haben beide und auch Sasank Reddy [Reddy et al., 2008] mehrere Schlussfolgerungsmodelle (Entscheidungsbaum, Bayessches Netz, Markov Modelle, Random Forest, ...) betrachtet und miteinander verglichen.

Wie man mit Verbindungsabbrüchen umgeht und zwischen ähnlichen Verkehrsmitteln

unterscheiden kann hat unter anderem auch Filip Biljecki beschäftigt. Des Weiteren baut er für die unterschiedlichen Kategorien von Verkehrsmitteln ein hierarchisches Modell auf, welches ihm helfen soll, bessere Entscheidungen zu treffen. [Biljecki et al., 2013]

2.1 Verwendete Daten

Ein wesentlicher Unterschied zwischen all den betrachteten Publikationen sind die verwendeten Daten. Nur die GPS-Spuren bilden eine gemeinsame Basis. Manche untersuchten die Verwendung von GSM- und Wifi-Informationen [Reddy et al., 2010], stützten sich auf Zusatzinformationen durch weitere Sensoren wie zum Beispiel einem Beschleunigungssensor [Reddy et al., 2010] [Nadine Schüssler et al., 2011]. Andere, wie Leon Stenneth, verwendeten Live-Informationen von den öffentlichen Verkehrsmitteln und kombinierten diese mit GIS-Informationen, seien es Busstationen, Bahnstrecken, das Straßennetz oder Parkplätze [Stenneth et al., 2011].

2.2 Betrachtete Verkehrsmittel

Ein weiterer Unterschied zwischen den Publikationen sind die betrachteten Verkehrsmittel. Hierbei reicht die Spanne der unterschiedenen Fortbewegungsmöglichkeiten von "gehen, laufen, Fahrrad und motorisiert" [Reddy et al., 2010] bis hin zu "gehen, Zug, U-Bahn, Rad, Auto, Straßenbahn, Bus, Fähre, Segelboot und Flugzeug" [Biljecki et al., 2013].

2.3 Analyse der Publikationen von Yu Zheng

Zu den Meilensteinen auf dem Gebiet der Transport Mode Detection zählen die Publikationen von Yu Zheng und seinem Team: "Understanding Mobility Based on GPS Data" [Zheng et al., 2008a], "Understanding Transportation Modes Based on GPS Data

for Web Applications“ [Zheng et al., 2010] und “Learning Transportation Mode from Raw GPS Data for Geographic Application on the Web“ [Zheng et al., 2008b]. In diesen Artikeln wird unter anderem auf die Erkennung von den Abschnitten mit nur einem Verkehrsmittel eingegangen. Weiters wird erklärt wie in den Projekten ein aus analysierten GPS-Spuren erstellten Graph mit geographischen Wissen zur Verbesserung der Erkennungsrate beiträgt und mit welchen Schlussfolgerungsmodellen welche Ergebnisse erzielt werden konnten.

2.3.1 Geolife

Die Applikation, welche im Zusammenhang mit den Arbeiten entstanden ist, nennt sich Geolife und ist eine Webapplikation mit den Ansätzen eines sozialen Netzwerks. Dabei ging es darum, dass BenutzerInnen GPS-Spuren in Form einer Aufzeichnung auf die Webseite laden konnten, diese Spur vom Algorithmus analysiert und die Verkehrsmittel bestimmt wurden. Dargestellt wurden die Resultate auf einer Karte und die BenutzerInnen konnten diese mit anderen teilen.

Mit Hilfe der so gesammelten Daten konnte unter anderem Datamining betrieben und populäre Strecken festgestellt sowie Verkehrssituationen beurteilt werden. Außerdem konnten auch aktuelle Positionswerte mit einem GPS-fähigen Smartphone/Handy ausgewertet werden und Informationen wie z.B. Abfahrtszeiten der öffentlichen Verkehrsmittel angeboten werden. Wurde aber z.B. eine Strecke mit dem Auto in die Stadt gesucht so konnte auf Grund der bereits analysierten Fahrten über durchschnittliche Geschwindigkeit festgestellt werden, welche Strecke am schnellsten ans Ziel führt.

In diesem von Microsoft Research geführten Projekt standen umfangreiche Test- und Trainingsdaten (1,2 Millionen Kilometer und 48.000 Stunden) zur Verfügung. [Microsoft Research, 2015] Ein Bildschirmfoto der Applikation ist in Abbildung 2.1 zu sehen.

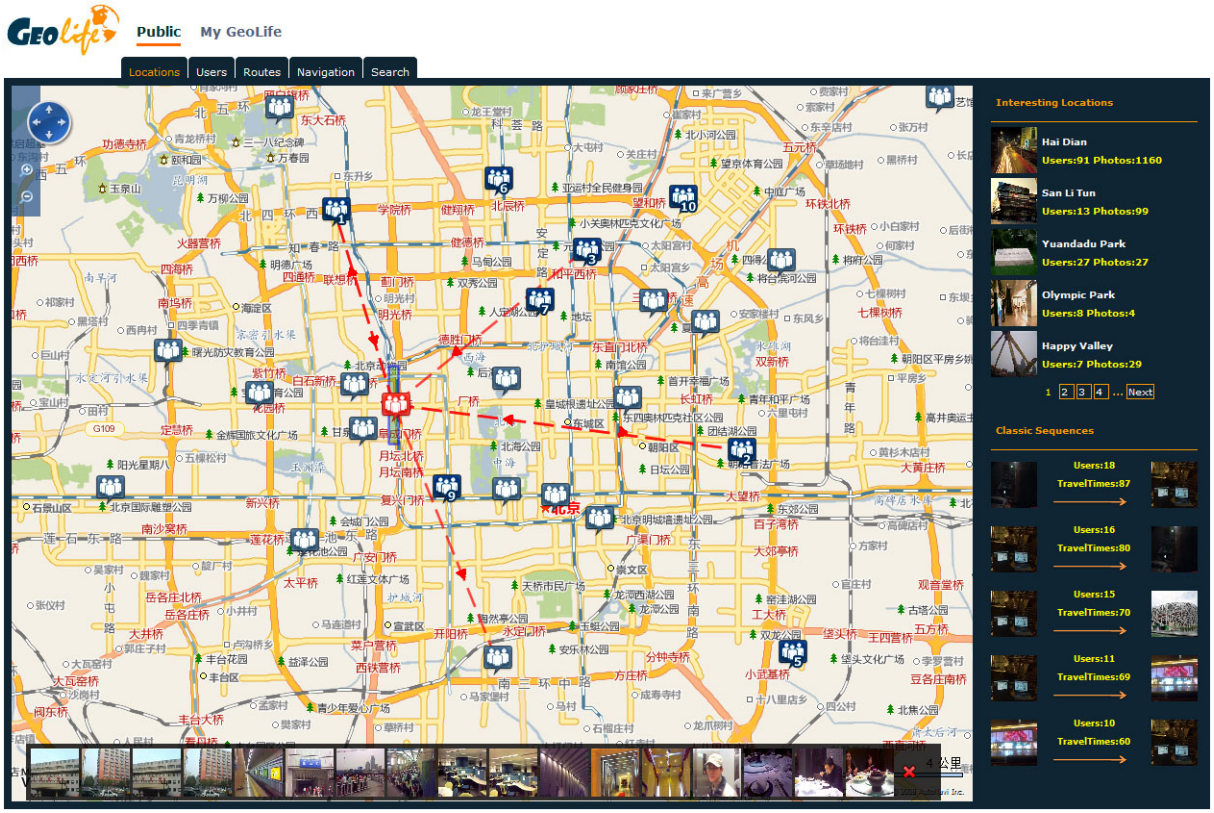


Abbildung 2.1: Geolife (Quelle: research.microsoft.com)

2.3.2 Segmentierung

Die Publikationen von Zheng beinhalten auch eine der detailiertesten Beschreibungen des Segmentierungsvorgangs. Dabei werden GPS-Spuren in Abschnitte, in welchen nur ein Verkehrsmittel verwendet wird, unterteilt. Diese Abschnitte können dann in weiterer Folge genauer analysiert und das benutzte Verkehrsmittel bestimmt werden.

Beim Segmentieren stützt sich Zheng darauf, dass Personen bei einem Wechsel des Verkehrsmittels stehenbleiben und sich ein Stück zu Fuß bewegen. Dies bedeutet, dass es einige GPS-Punkte mit einer Geschwindigkeit von genau oder beinahe 0 km/h gibt. Diese aufeinander folgenden Punkte können dann in ein “Gehen“-Segment zusammengefasst werden. Alle anderen Segmente werden vorläufig als “Nicht-Geh“-Segmente klas-

sifiziert. Zheng sagt außerdem, dass der Beginn und das Ende eines Segments mit dem Typ “Gehen“ ein wichtiger Indikator für einen Wechsel des Fortbewegungsmittels ist. Diese Aussage stützt sich auf die Erkenntnisse aus GPS-Daten, die von 65 Personen über 10 Monate gesammelt wurden.

2.3.3 Schlussfolgerungsmodelle

Um die Typen der “Nicht-Geh-Segmente“ bestimmten zu können, hat sich Zheng auf verschiedene Zusatzinformationen zu den Segmenten gestützt, darunter:

- Distanz
- Maximale Geschwindigkeit
- Maximale Beschleunigung
- Durchschnittliche Beschleunigung
- Richtungswechsel
- Stopprate
- Erwartete Geschwindigkeit
- Geschwindigkeitsänderungsrate

In weiterer Folge stellte er fest, dass die Stopprate, Richtungswechselrate und die Geschwindigkeitsänderungsrate am effektivsten und stabilsten gegenüber den verschiedenen Verkehrssituationen sind. Diese 3 Eigenschaften können auch mit ein paar der anderen kombiniert werden um noch weitere Verbesserungen zu erhalten. Werden allerdings zu viele Eigenschaften miteinbezogen, so konnte er eine Verringerung der Genauigkeit beim Bestimmen des Typs feststellen.

Für die tatsächliche Bestimmung des Typs führte Zheng verschiedene Experimente mit dem Entscheidungsbaum, der Support Vector Machine, dem Bayesschen Netz und dem Conditional Random Field durch. Dabei stellte er fest, dass der Entscheidungsbaum die

besten Ergebnisse im Zusammenspiel mit der in Abschnitt 2.3.3 beschriebenen Segmentierungsmethode liefert.

Während dem Analysieren der Trainingsdaten wurde im Hintergrund ein Graph aufgebaut, welcher Informationen über die jeweils betrachtete geografische Region widerspiegelt. In diesen geografischen Regionen waren 28 große Städte in China, mehrere Städte in den USA sowie Südkorea und Japan enthalten. Mit diesem Graph wurden die Schlussfolgerungen nochmals überprüft und es konnte eine weitere Verbesserung erzielt werden. Schlussendlich konnte der von Zheng entwickelte Algorithmus 76,2% der Testfälle ohne jegliche Zusatzinformationen im Sinne von GIS-Daten oder weitere Sensordaten korrekt identifizieren.

2.4 Analyse der Publikationen von Leon Stenneth

Leon Stenneth vergleicht in der Publikation “Transportation Mode Detection usign Mobile Phones and GIS Information“ [Stenneth et al., 2011], ähnlich wie Zheng auch, verschiede Schlussfolgerungsmodelle. Allerdings verwendete er auch zusätzliche GIS-Informationen, um eine genauere Bestimmung zu ermöglichen. Außerdem arbeitet die entwickelte Applikation nicht mit ganzen GPS-Spuren, sondern analysiert immer 2 Punkte innerhalb eines 30 Sekunden Intervalls, wodurch auch das eigentliche Segmentieren entfällt.

2.4.1 GIS-Informtionen

Die von Stenneth verwendeten GIS-Informationen beinhalten sowohl die Gleise von Zügen als auch Bushaltestellen sowie die aktuelle Position von allen Bussen in Chicago. Daraus berechnete er verschiedene Zusatzinformationen für die Bestimmung des Transporttyps:

- Durchschnittliche Nähe zu den Gleisen

- Durchschnittliche Nähe zu Bushaltestellen
- Durchschnittliche Nähe zu dem nächsten Bus

2.4.2 Schlussfolgerungsmodelle

Um den Typ des jeweils betrachteten Abschnitts zu bestimmen, zieht auch Stenneth mehrere Zusatzwerte als Kriterien zu Hilfe:

- Durchschnittliche Geschwindigkeit
- Durchschnittliche Nähe zu den Gleisen
- Durchschnittlicher Abstand zu einem Bus
- Durchschnittliche Beschleunigung
- Durchschnittlicher Richtungswechsel
- Durchschnittliche Bushaltestellennähe
- Durchschnittliche Genauigkeit der Koordinaten
- Distanz zum nächsten Bus

Wie auch Zheng stellt auch Stenneth fest, dass nur ein paar der Zusatzwerte wirklich effektiv sind. In seinem Fall waren das die durchschnittliche Geschwindigkeit, Beschleunigung und die Nähe zu den Gleisen sowie der Abstand zu anderen Bussen sowie die Distanz zum nächsten Bus.

Die von Stenneth betrachteten Modelle sind Naive Bayes, Bayessches Netz, Entscheidungsbaum, Random Forest und Multilayer Perceptron. In den Experimenten stellte er fest, dass der Random Forest mit den GIS-Informationen das vielversprechendste Modell mit mehr als 93% richtig erkannten Verkehrsmitteln ist. Ohne GIS-Informationen schneidet auch der Random Forest ähnlich wie bei Zheng der Entscheidungsbaum mit 76% ab. Mit GIS-Informationen erreicht auch der Entscheidungsbaum eine Erkennungsrate von mehr als 92%.

2.5 Analyse der Publikationen von Filip Biljecki

Filip Biljecki veröffentlichte eine Arbeit mit dem Titel “Transportation mode-based segmentation and classification of movment trajectories“ [Biljecki et al., 2013]. Darin vergleicht er nicht nur eine Vielzahl von Publikationen zu dem Thema Verkehrsmittelerkennung (unterer anderem [Schuessler and Axhausen, 2009], [Zheng et al., 2010], [Reddy et al., 2010], [Gonzalez et al., 2010]) anhand der Fortbewegungsmittel, den Zusatzwerten, ob GIS-Daten verwendet wurden und welches Resultat erzielt worden ist, sondern führt auch ein hierarchisches Modell für die Erkennung der Transportmittel ein. Weiters verwendet er zur Bestimmung des Transportmittels auch GIS-Daten wie Bus- und Straßenbahnhaltestellen. Ein Kapitel ist in der Arbeit auch der Behandlung von Störungen oder Unterbrechungen des GPS-Signals gewidmet. Darin wird aufgeführt, welche Fälle weiterhin behandelt werden können und welche sich mit diesem Ansatz nicht beheben lassen. In dieser Arbeit konnte eine Genauigkeit von 95% erreicht werden.

2.5.1 Segmentierung

Für die Segmentierung verwendet Biljecki dieselbe Methodik wie Zheng aber erweitert diese dahingehend, dass auch dann segmentiert wird, wenn ein Signalverlust (30 Sekunden keine neuen Werte) feststellt werden kann. Diese Entscheidung wird damit begründet, dass es bei einem Verkehrsmittelwechsel oft zu einem Signalverlust kommt. Überschreitet ein Stopp eine bestimmte Dauer (12 Sekunden) so wird auch segmentiert, denn dies könnte auch auf einen Wechsel hindeuten. Da die Track-Punkte nicht 100% genau sind, wird alles unter 2km/h als Stopp eingestuft. Weil nach der Erkennung alle aufeinander folgenden Segmente mit dem selben Typ zusammengefasst werden, ist eine Übersegmentierung kein Problem.

2.5.2 Schlussfolgerung

Um das Fortbewegungsmittel feststellen zu können wird ein Expertensystem verwendet. Dieses System basiert auf einer Fuzzy-Logic und klassifiziert die Segmente mit Wahrscheinlichkeitswerten. Dazu verwendet dieses System auch die hierarchische Gliederung der verschiedenen Verkehrsmittel welche in Abbildung 2.2 ersichtlich ist. Diese Gliederung soll verhindern, dass sich das System zu früh auf einen Typ festlegen muss - das System kann den Typ dadurch Schrittweise bestimmen.

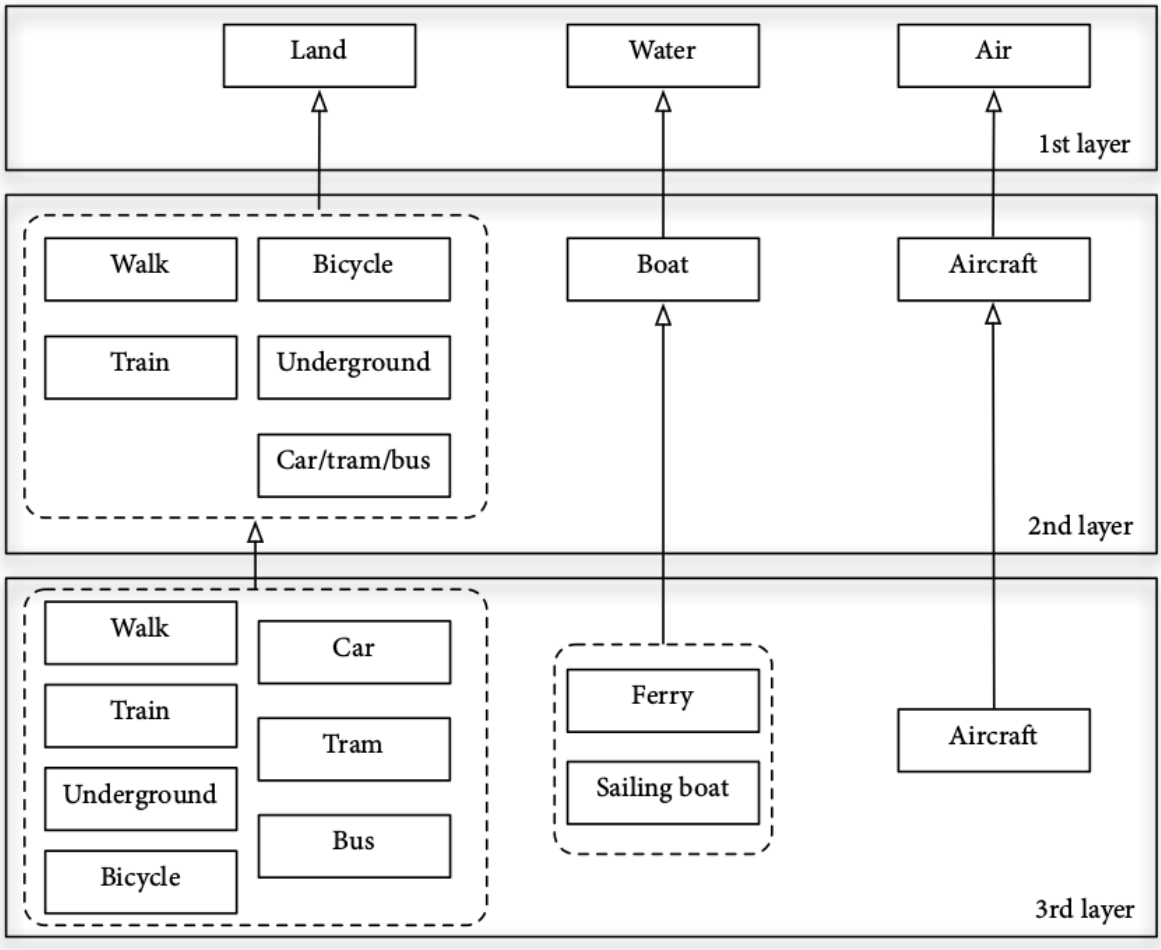


Abbildung 2.2: Fortbewegungsmittel-Hierarchie

Zusammenfassend sagt Biljecki, dass es sehr schwierig ist, alle möglichen Fälle der Realität abzubilden und wirklich zufriedenstellende Resultate zu erhalten. Dies betrifft vor allem den Segmentierungs- und Bestimmungsprozess von Daten mit sehr vielen fehlerhaften Ausreißern (Rauschen) oder jene Fällen, in denen wenige Beispieldaten vorhanden sind. Weiters meint er, dass Fehler nicht zwangsläufig auf das System zurückzuführen sind, sondern, dass es sich bei diesen Fehlern oft um spezielle Situationen handelt, welche sehr kompliziert zu modellieren sind oder drastische Auswirkungen auf die allgemeine Performanz haben.

Alles in allem kann aber gesagt werden, dass das von Biljecki vorgestellte Modell zehn verschiedene Verkehrsmittel unterscheiden kann. Dies sind mehr als in allen anderen vom Autoren betrachteten Publikationen. Außerdem konnte durch die Verwendung von GIS-Daten bessere Resultate (95% bis 100% je nach Daten) als in vielen anderen Publikationen im Bereich der Verkehrsmittelerkennung erzielt werden.

2.6 Analyse der Publikationen von Sasank Reddy

Sasank Reddy hat sich in den Publikationen “Using Mobile Phones to Determine Transportation Modes“ [Reddy et al., 2010] und “Determining Transportation Mode on Mobile Phones“ [Reddy et al., 2008] wie auch Stenneth und Zheng mit verschiedenen Schlussfolgerungsmodellen befasst. Bevor diese Modelle aber zum Einsatz kamen, wurde evaluiert, mit welchen weiteren Sensoren man sinnvolle Daten aufzeichnen könnte. Im Zuge dessen wurde auch überprüft, an welche Stelle am Körper das Smartphone die genauesten Daten liefert und wie möglichst Energieeffizient Daten aufgezeichnet und übermittelt werden können. Im Gegensatz zu anderen Publikationen wurde in diesen nicht genauer zwischen den motorisierten Verkehrsmitteln unterschieden.

2.6.1 Sensoren

Dadurch, dass die meisten Smartphones nicht nur über ein GPS-Modul sondern auch über WIFI, einen Beschleunigungssensor sowie Bluetooth und natürlich über ein GSM-Modul verfügen, wurde in diesen Arbeiten auch evaluiert, ob es möglich und sinnvoll ist Daten von diesen Geräten und Sensoren miteinzubeziehen.

Bluetooth wäre zwar interessant, aber es kommt hauptsächlich innerhalb von Gebäuden zu Einsatz (TV, Radio, Computer, ...). Darum konnte dieser Sensor nicht für die Bestimmung des Verkehrsmittels eingesetzt werden.

Wifi und GSM in den Erkennungsprozess miteinzubeziehen konnte nach einer Reihe von Versuchen ausgeschlossen werden, da der Grad der Verbesserung nur 0,6% betrug und ein wesentlich höherer Energiebedarf gegeben war. Außerdem war die Abhängigkeit von Wifi und GSM in ländlichen Gegenden mit schlechtem Empfang und / oder wenig Wifis ein weiterer Grund gegen diese Sensoren.

Die vielversprechendste Kombination war jene aus GPS-Daten und den Daten des Beschleunigungssensors, welche auch für die weiteren Experimente verwendet wurde.

2.6.2 Schlussfolgerungsmodelle

Um das Fortbewegungsmittel des jeweiligen Abschnittes zu erkennen, zieht Reddy folgende Zusatzinformationen heran:

- Geschwindigkeit
- Varianz des Beschleunigungssensorsignals
- 3 Beschleunigungssensorwerte (3Hz, 2Hz, 1Hz)

Die in dieser Publikation betrachteten Modelle sind der Entscheidungsbaum, K-Means Clustering, Naive Bayes, Nearest Neighbor, Support Vector Machine sowie ein Continuous Hidden Markov Model und ein Entscheidungsbaum in Kombination mit einem

Discrete Hidden Markov Model. Dabei konnte festgestellt werden, dass die letzte Variante die besten Resultate mit 93,6% erzielt. Wie aber auch in den anderen Publikationen war der Entscheidungsbaum mit 91,3% nicht weit abgeschlagen hinter dem kombinierten Ansatz.

2.7 Zusammenfassung

Neben den vorgestellten Publikationen zum Thema Verkehrsmittelerkennung gibt es noch weitere interessante Publikationen, deren Erfahrungen zum Teil in diese Arbeit eingeflossen sind. Unter diesen Publikationen sind unter anderem “Processing raw data from global positioning systems without additional information“ [Schuessler and Axhausen, 2009] und “Improving post-processing routines for gps oversavations using propted-recall data“ [Nadine Schüssler et al., 2011] von Nadine Schüssler sowie “Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks“ [Gonzalez et al., 2010] von Paola Gonzalez.

Für diese Arbeit wurde für den Segmentierungsprozess die Vorgehensweise von Yu Zheng [Zheng et al., 2010] verwendet. Darin sind aber auch die von Filip Biljecki [Biljecki et al., 2013] vorgeschlagenen Änderungen eingeflossen, welche eine weitere Segmentierungsregel bei Empfangsverlust beinhaltet.

Für die weitere Verarbeitung der GPS-Daten wurde der Entscheidungsbaum aufgrund des guten Abschneidens in den erwähnten Publikationen verwendet. Der Entscheidungsbaum wurde zur nachfolgenden Analyse und Auswertung der verbesserten Verkehrsmittelerkennung in zwei Varianten implementiert. In der einen Variante verwendet er nur aus den GPS-Daten berechnete Werte (Geschwindigkeit, Beschleunigung, ...), wie es auch in vielen anderen Arbeiten gemacht wurde. In der zweiten Variante greift der Entscheidungsbaum aber auch auf GIS-Informationen zurück, wie es zum Beispiel Leon

Stenneth in seiner Publikation [Stenneth et al., 2011] beschrieben hat.

Ein weiterer Grund neben dem guten Abschneiden bei der GIS-Daten-gestützten Verkehrsmittelerkennung war die eigene Zielsetzung, dass keine zusätzlichen Sensoren oder Geräte neben dem Gerät für die Aufzeichnung der GPS-Daten verwendet werden sollen.

Modellbildung und Einbindung der GPS- und GIS-Daten

Dieser Abschnitt behandelt die Akquirierung und die Struktur der verwendeten GPS-Daten für das Training des Entscheidungsbaums. Außerdem wird erläutert, wieso der Entscheidungsbaum als Schlussfolgerungsmodell ausgewählt und wie er erstellt wurde. Aufgrund der unterschiedlichen Attribute der beiden Fälle (mit und ohne GIS-Daten), sehen die zwei Entscheidungsbäume sehr unterschiedlich aus und werden daher nur oberflächlich miteinander verglichen.

Sowohl für die Trainingsdaten für den Entscheidungsbaum als auch für die Testdaten wird erläutert, wie diese aufgezeichnet wurden. Dies inkludiert sowohl die Geräte als auch die Software, welche dazu verwendet worden ist. Weiters wird auf die Struktur der GPS-Spuren und der GIS-Daten eingegangen.

Außerdem wird dargelegt, woher die verwendeten GIS-Daten stammen, wie diese extrahiert wurden und welche Rolle sie im weiteren Prozess spielen. Abschließend wird auch auf die Positionsdaten der verschiedenen Verkehrsmittel des ÖPNV eingegangen und in welcher Weise diese hätten eingesetzt werden können.

Typ	Anzahl	Distanz (km)
Auto	73	752,33
Fußgänger	53	71,73
Fahrrad	41	1.120,93
Zug	8	239,04
Bus	16	66,03
Gesamt	**190**	**2.250,07**

Tabelle 3.1: Trainingsdatenübersicht

3.1 Trainingsdaten

Für das Training der Entscheidungsbäume konnten GPS-Aufzeichnungen aus einem Projekt von Sebastian Nagel "Möglichkeitsstudie zum Projekt: Mobilitäts-Tracker"verwendet werden. Diese Daten wurden mit verschiedenen GPS-Geräten (Wintec WTB-202, Columbus V-900, photoMate 887 Lite, qStarz BT-Q1300, xaiox) und der Hilfe von mehreren Personen aufgezeichnet. [Sebastian Nagel, 2011]

Weiters wurden zum Training auch neue Datensätze verwendet, die mit zwei verschiedenen Smartphones und mit Hilfe der App “MyTrack“ aufgezeichnet wurden. Diese App wurde ausgewählt, da sie sich sehr einfach handhaben lässt und die einzelnen Aufzeichnungen komfortabel exportiert werden können.

Einen groben Überblick über die gesammelten Trainingsdaten bietet die Tabelle 3.1. Die erste Spalte enthält den Verkehrsmitteltyp, die Zweite die Anzahl der Segmente und die dritte Spalte enthält die Gesamtdistanz in Kilometern von dem jeweiligen Typ. Insgesamt beinhalten die Trainingsdaten 187 Segmente der verschiedenen Transportmittel und erstrecken sich über annähernd über 2.000 Kilometer.

3.1.1 Struktur der GPS-Daten

Diese Trainingsdaten sind in einzelnen Dateien als XML abgelegt und entsprechen dem gängigen GPX-Format, wie es im Listing 3.1 ersichtlich ist. Die Spezifikation für GPX kann auf der Webseite von Topografix unter `http://www.topografix.com/GPX/1/1/` gefunden werden. Ein zugehöriges XML-Schema findet man hier `http://www.topografix.com/GPX/1/1/gpx.xsd`. [Topografix, 2004]

Track (trk) Üblicherweise beginnt eine solche Datei mit einem gpx-Element, welches wiederum einen Track enthält. Ein Track repräsentiert eine Aufzeichnung oder Spur und enthält eine Folge aller aufgezeichneten Trackpoints. Diese sind aber wiederum in ein oder mehrere Tracksegmente gegliedert.

Tracksegment (trkseg) Ein Track kann aus einem oder mehreren Tracksegmenten bestehen. Die Tracksegmente enthalten wiederum beliebig viele aufeinander folgende Trackpoints. Mit diesen Segmenten kann ein Track in logische Abschnitte unterteilt werden. Außerdem kann ein neues Segment begonnen werden, wenn zum Beispiel die Verbindung verloren oder der GPS-Empfänger aus- und wieder eingeschaltet wurde.

Trackpoint (trkpt) Ein Trackpoint entspricht einem Punkt des aufgezeichneten Tracks und enthält Koordinate (Längen- und Breitengrad) sowie einen Zeitstempel und die Höhenmeter. Es gibt aber auch Fälle, in welchen bei einem Trackpoint auch die Geschwindigkeits- oder Beschleunigungsdaten abgelegt werden.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gpx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http:
    //www.topografix.com/GPX/1/1" ...>
    <metadata>
        <name>Badgasse - FH</name>
        <desc></desc>
    </metadata>
    <trk>
        <name>Badgasse - FH</name>
        <trkseg>
            <trkpt lat="47.39786" lon="9.735109">
                <ele>475.0</ele>
                <time>2015-02-19T07:20:18.156Z</time>
            </trkpt>
            ...
            <trkpt lat="47.405439" lon="9.744841">
                <ele>492.0</ele>
                <time>2015-02-19T07:24:35.160Z</time>
            </trkpt>
        </trkseg>
    </trk>
</gpx>
```

Listing 3.1: GPX-Datei

3.1.2 Entscheidungsbaum als Schlussfolgerungsmodell

Aufgrund des guten Abschneidens des Entscheidungsbaums in verschiedenen Arbeiten [Stenneth et al., 2011], [Reddy et al., 2010], [Sebastian Nagel, 2011]und [Zheng et al., 2008b], wurde auch in dieser Arbeit ein Entscheidungsbaum als Schlussfolgerungsmodell verwendet. Um einen Entscheidungsbaum erstellen zu können, werden Trainingsdaten benötigt. Daraus werden dann die Regeln für den Entscheidungsbaum bzw. die jeweiligen Entscheidungen des Baums abgeleitet.

Deshalb wurde für diese Trainingsdaten ein Teil der gesammelten GPS-Daten manuell segmentiert und mit dem benutzten Verkehrsmittel ergänzt. Mit Hilfe des Prototyps wurden die Trainingsdaten eingelesen, gefiltert und mit zusätzlichen Informationen bereichert. Diese zusätzlichen Informationen sind bei der Variante ohne GIS-Daten zum Beispiel Geschwindigkeit und Beschleunigung. Bei der Variante mit GIS-Daten wird unter anderem die Nähe zu Bushaltestellen oder zu Schienen ergänzt. Danach wurde für alle eingelesenen Segmente die berechneten Werte und das dazugehörige Verkehrsmittel in einer Datei im CSV-Format abgelegt.

Für die konkrete Generierung der Entscheidungsbäume wurde das Tool Rapidminer verwendet. Dabei handelt es sich um eine Open-Source Datamining Software, welche sowohl verschiedenste Datenquellen unterstützt (Datenbanken und auch einzelne Dateien in unterschiedlichen Dateiformaten) als auch die Generierung von Entscheidungsbäumen über eine komfortable grafische BenutzerInnenoberfläche erlaubt. In der freien Version ist jedoch nur der CSV-Import verfügbar. Deshalb wurde vom Prototypen auch eine CSV-Datei generiert, welche in der Folge einfach in Rapidminer eingebunden und ausgewertet werden konnte.

Schlussendlich wurden aus den eingebundenen Daten Entscheidungsbäume für die Variante mit GIS-Daten und ohne generiert. Diese können sowohl als Bilder als auch in Textform exportiert werden. In weiterer Folge wurde aus diesen Entscheidungsbäumen

in Textform PHP-Klassen generiert (siehe Abschnitt “5.5.1 Erstellen der Entscheidungsbäume“).

Überanpassung (Overfitting)

Beim Erstellen von Entscheidungsbäumen wie auch bei anderen von Trainingsdaten lernenden Algorithmen muss beachtet werden, dass das Ergebnis nicht zu sehr auf die Trainingsdaten zugeschnitten ist. Dies bedeutet, man möchte, dass mit Hilfe der Trainingsdaten ein Modell generiert wird, welches auch für Nichttrainingsdaten ein akzeptables Resultat liefert und nicht zu sehr auf die Gegebenheiten / Ungenauigkeiten in den Trainingsdaten spezialisiert ist. Ist dies jedoch der Fall so spricht man von einer Überanpassung (Overfitting) des Modells auf die Daten. [Tom Dietterich, 1995]

Zurückschneiden (Pruning)

Ist ein Entscheidungsbaum zu sehr auf die Trainingsdaten angepasst, so muss dieser wieder zurückgeschnitten werden. Dies bedeutet, dass einzelne Blätter oder auch Teilbäume wieder entfernt werden, um ein bestmögliches Resultat (geringe Anzahl an Fehlern) für Nichttrainingsdaten zu erhalten. Für diese Aufgabe wurden verschiedene Algorithmen entwickelt, welche unter anderem in der Publikation “Pruning Decision Trees with Misclassification Costs“ von Jeffrey Bradford beschrieben werden. [Jeffrey P. Bradford et al., 1998]

Generierung eines Entscheidungsbaumes

Um einen Entscheidungsbaum generieren zu können gibt es verschiedene Algorithmen und Ansätze die im folgenden Abschnitt kurz erklärt werden. Die betrachteten Algorithmen sind:

- ID3
- C4.5

- CART
- CHAID

ID3

Der ID3 Algorithmus (Iterative Dichotomiser 3) wurde von John Ross Quinlan in 1986 entwickelt und ist der Vorgänger C4.5. ID3 wurde entworfen um nicht alle möglichen Entscheidungsbäume generieren zu müssen, wenn man nur einen möglichst guten Entscheidungsbaum für die gegebenen Daten benötigt. Im Allgemeinen generiert ID3 einfache Entscheidungsbäume, aber es kann nicht garantiert werden, dass es keine bessere Entscheidungsbäume gibt. Der allgemeine Algorithmus ist iterativ und geht dabei wie folgt vor: [Quinlan, 1986]

- Es wird eine Teilmenge der Trainingsdaten per Zufall ausgewählt (Window) und für dieses ein Entscheidungsbaum generiert. Dieser Entscheidungsbaum ist für alle Elemente innerhalb der Teilmenge gültig und kann diese korrekt klassifizieren.
- Nachfolgend werden alle anderen Elemente mit diesem Entscheidungsbaum klassifiziert. Kann der Baum alle Elemente korrekt klassifizieren, so ist ein ausreichend guter Entscheidungsbaum gefunden und das Ziel erreicht. Wenn nicht alle Elemente korrekt klassifiziert werden können, wird eine Auswahl der falsch klassifizierten Elemente in die Teilmenge (Window) übernommen und erneut ein Entscheidungsbaum gesucht.

Für das Erstellen des Entscheidungsbaumes selbst wird dabei auf das finden des Attributs mit der niedersten Entropie für den aktuellen Knoten gesetzt. Entropie ist in diesem Zusammenhang ein Maß für die Reinheit bzw. die Verunreinigung. [Howard Hamilton, 2009] Ein hoher Reinheitswert bedeutet zugleich ein Gewinn an Information (information-gain) da sich der Informationsgewinn aus den berechneten Entropien-Werten berechnen lässt. [Thomas Mitchell, 1997]

Dies bedeutet, man sucht ein Attribut, welches für eine möglichst reine Menge an Objekten in diesem Knoten sorgt. Hat man zum Beispiel eine Menge von männlichen und

weiblichen Personen so sucht man ein Attribut, welches bewirkt, dass im nächsten Knoten eine möglichst hohe Wahrscheinlichkeit für männlich oder weiblich Person gegeben ist und somit die Anzahl nicht ausbalanciert ist. Wäre die Anzahl genau ausbalanciert, so würde man durch diesen Knoten keinen Schritt näher an die schlussendliche Klassifizierung kommen.

Mit diesem Ansatz können Entscheidungsbäume bereits nach wenigen Iterationen für ein Trainingsset von bis zu 30.000 Objekten und 50 Attributen gefunden werden. [Quinlan, 1986]

ID3 macht kein Backtracking und überdenkt sozusagen seine Auswahl an Attributen zu keinem späteren Zeitpunkt. Dies kann auch dazu führen, dass ein lokales aber kein globales Optimum gefunden wird. [Howard Hamilton, 2009] Weiters kann es vorkommen, dass der erstellte Entscheidungsbaum überangepasst ist, wenn eine zu kleine Menge an Trainingsdaten verwendet worden ist. [RapidMiner, 2015]

C4.5

Der C4.5 Algorithmus ist eine Erweiterung des ID3 Algorithmus, welche versucht unter anderem folgende Probleme des ID3 Algorithmus zu adressieren: [Howard Hamilton, 2009]

- Überanpassen der Entscheidungsbaums an die Daten
- Vermeiden von fehlerhaften Zurückschneiden
- Verbesserung de Effektivität der Berechnung
- Handhabung von nicht diskreten Werten durch Umformen der Bedingung für das Attribut auf größer oder einen kleiner gleich
- Handhabung von Trainingsdaten mit fehlenden Attributen
- Nachträgliches Zurückschneiden von Teilbäumen und ersetzen dieser durch Blätter-Knoten

CART

Der CART-Algorithmus benutzt im Gegensatz zu C4.5 und ID3 den Gini-Index als Entscheidungskriterium zur Bestimmung des nächsten Attributs. [RapidMiner, 2015] Der Gini-Index ist ein weiteres statistisches Maß für Angabe der Reinheit. Information-gain und der Gini-Index unterscheiden sich in der Berechnung aber liefern laut Laura Raileanu in nur zwei Prozent der Fälle tatsächlich ein unterschiedliches Ergebnis. [Raileanu and Stoffel, 2004]

Der Algorithmus wählt die Attribute auch nach dem Kriterium der Reinheit für die daraus folgenden Knoten. Das bedeutet, dass er die Daten so aufzuteilen versucht, dass möglichst reine Ergebnisse in den neuen Knoten sind. An dieser Stelle stopp der Algorithmus jedoch nicht, sondern er wählt jenes Attribut welches die Reinheit maximiert und erzeugt eine Reihe weiterer Teilbäume. Diese Teilbäume werden dann bis zur Wurzel zurückgeschnitten. Dabei wird abgeschätzt wie hoch die Kosten für eine falsche Klassifizierung sind und jener Teilbaum mit den geringsten Kosten wird gewählt. [Wei-Yin Loh, 2008]

CHAID

CHAID steht für CHi-squared Automatic Interaction Detection und verwendet statt der Entropie und dem Informationsgewinn den Chi-Quadrat-Test für die Auswahl des nächsten Attributs. Eine Erklärung zu diesem Test liefert zum Beispiel Dr. Chandran in der Präsentation mit dem Titel “Chi-Square Test“ `http://de.slideshare.net/syamchandran3/chi-squared-test-2`. Weiters arbeitet dies Algorithmus nicht mit nummerischen Werten und stoppt das Wachstum des Baumes bevor der Baum zu groß wird. CHAID benötigt eine große Menge an Daten um verlässliche Ergebnisse / Entscheidungsbäume erzeugen zu können. [RapidMiner, 2015]

Die Entscheidungsbaum-Operatoren von RapidMiner

RapidMiner bietet verschieden Operatoren für die verschieden Algorithmen an. Darunter einen ID3-Operator für den ID3 Algorithmus und einen CHAID-Operator für den

CHAID-Algorithmus. Weiters gibt es einen Decision-Tree-Operator welcher abhängig vom ausgewählten Klassifizierungskriterium den CART-Algorithmus (“gini-index“) oder C4.5-Algorithmus (“gian-ratio“, “information-gain“) für den Entscheidungsbaum auswählt. [RapidMiner, 2015]

Das Klassifizierungskriterium “gain-ratio“ bedeutet in diesem Zusammenhang, dass möglichst reine Knoten generiert werden sollen, aber nicht zu viele. [RapidMiner, 2015]

Neben den erwähnten Algorithmen gibt es noch den C5.0 und MARS-Algorithmus, welche aber nicht in Rapidminer verfügbar sind und deshalb auch nicht weiter betrachtet werden.

Konfigurationsmöglichkeiten für die Generierung der Entscheidungsbäume

Rapidminer bietet sowohl die Möglichkeit Einfluss auf das Zurückschneiden als auch auf das Überanpassen von Entscheidungsbäumen zu nehmen. Grundsätzlich kann ein Entscheidungsbaum mit verschiedenen Parametern beeinflusst werden (siehe Abbildung 3.1). Als Kriterium für das Wachsen des Baumes wurde der Informationsgewinn gewählt. Die für diese Arbeit relevanten Parameter waren die maximale Tiefe des Baumes, der Zuversichtswert sowie der minimale Informationsgewinn mit jedem neuen Knoten und, dass das Zurückschneiden angewandt wird.

Die maximale Tiefe des Baumes ist dabei ein maximaler Wert der das Wachstum für den Baum einschränkt und dadurch die Optimierung/Überanpassung an die Testdaten ab einem gewissen Punkt beschränkt. Der minimale Informationsgewinn hat dabei einen ähnlichen Effekt wie die maximale Tiefe. Kann beim Aufteilen des aktuellen Knotens ein weiterer Informationsgewinn über dem angegebenen Wert erreicht werden, so wird dieser aufgeteilt. Ansonsten wird der Baum an diesem Punkt nicht mehr wachsen, was wieder eine Überanpassung verhindern kann. Der Zuversichtswert ist ein Wert der für das Zurückschneiden des Baumes verwendet wird und angibt welches Level an Zuversicht gegeben sein muss um das Zurückschneiden tatsächlich anzuwenden (steht

im Zusammenhang mit der durch das Zurückschneiden verursachten Fehlentscheidungen). [RapidMiner, 2015]

gini index
informatio
und strukt
punkte in
subsection

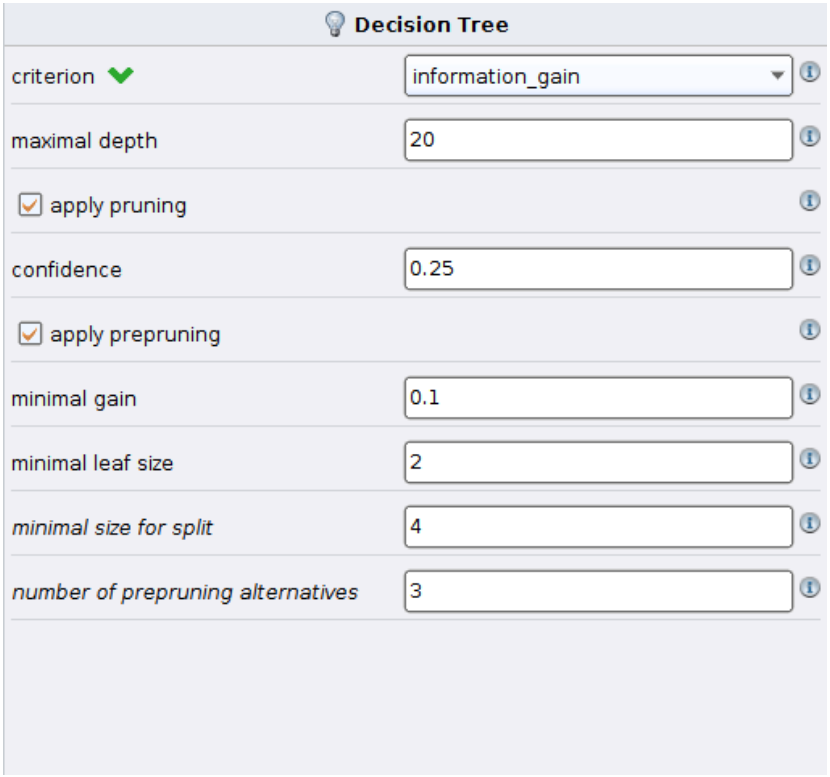


Abbildung 3.1: Konfigurationsmöglichkeiten für Entscheidungsbäume

Entscheidungsbaum ohne GIS-Daten

Der Entscheidungsbaum ohne GIS-Daten ist in Abbildung 3.2 zu sehen. Bei diesem Entscheidungsbaum wurden mittlere und maximale Geschwindigkeit, Stopprate sowie mittlere und maximale Beschleunigung als Indikatoren für den Entscheidungsprozess gewählt. Wie diese Werte berechnet und ergänzt werden wird im Abschnitt “5.2 Schlussfolgerungsvariablen“, erklärt.

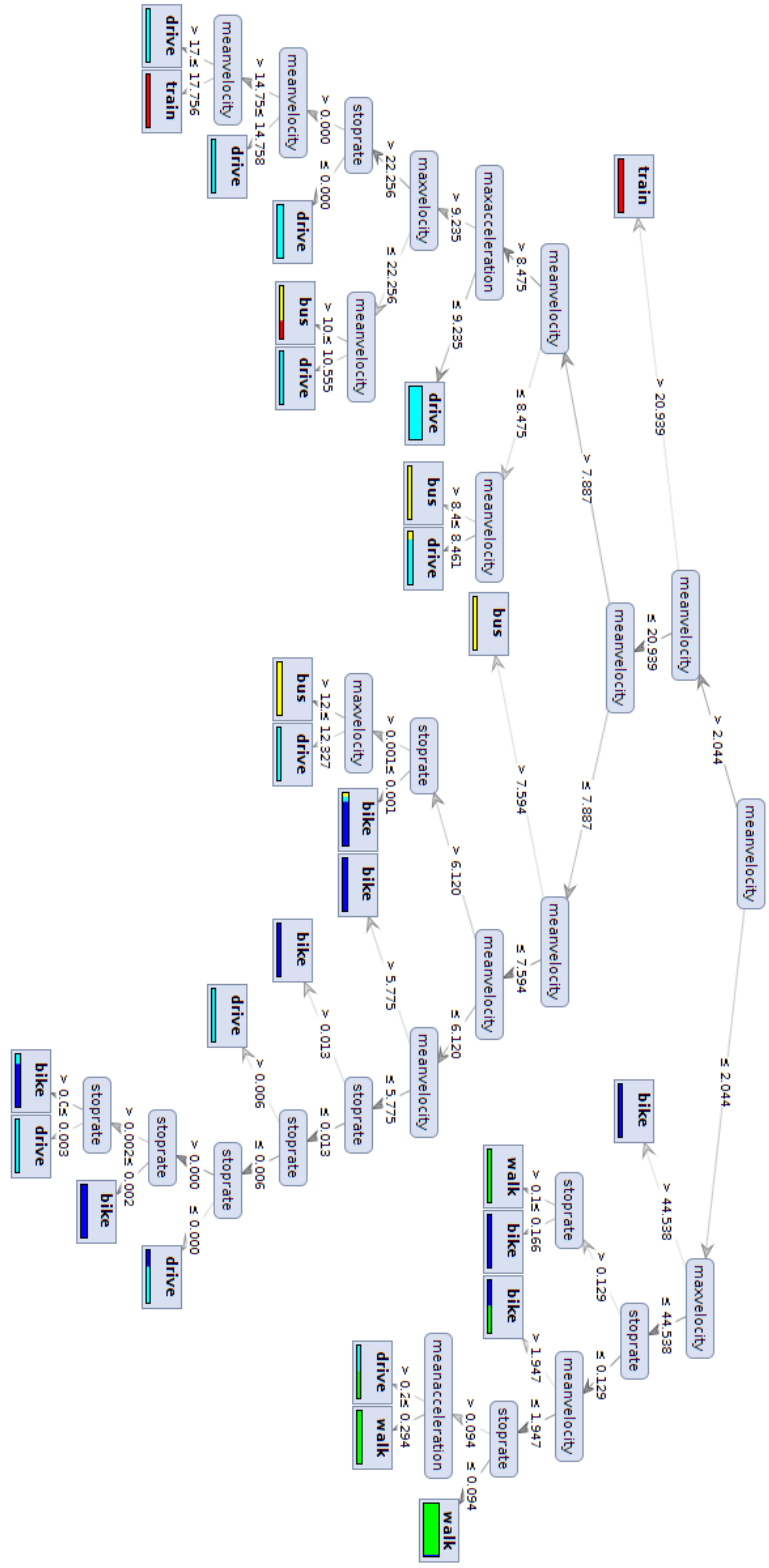


Abbildung 3.2: Entscheidungsbaum ohne GIS-Daten

Entscheidungsbaum mit GIS-Daten

Der Entscheidungsbaum mit GIS-Daten ist in Abbildung 3.3 zu sehen. Bei diesem Entscheidungsbaum wurde zusätzlich zu den Geschwindigkeits-, Beschleunigungs- und Stoppwerten auch Werte für die verwendeten GIS-Daten gesammelt. Die zusätzlichen Werte beinhalten einen Wert (ptscloseness) welcher die Stopps in der Nähe von Bushaltestellen und Bahnhöfen wiederspiegelt, sowie zwei Werte für die Nähe zur Autobahn (highwaycloseness) und zu Gleisen (railcloseness). Wie diese Werte berechnet und ergänzt werden, wird im Abschnitt “5.2 Schlussfolgerungsvariablen“, erklärt.

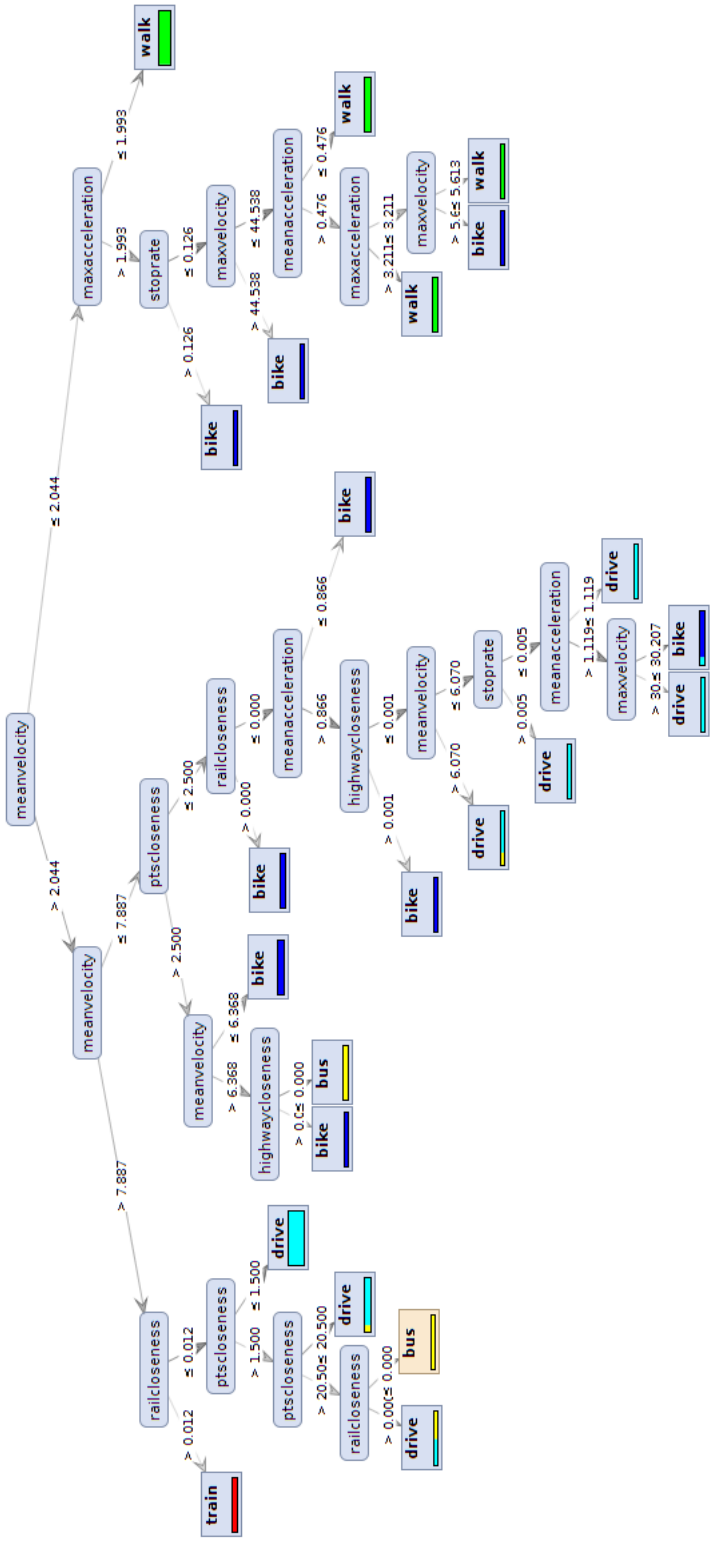


Abbildung 3.3: Entscheidungsbaum mit GIS-Daten

3.2 Neue Aufzeichnungen

Wie bereits im Abschnitt “3.1 Trainingsdaten“ erwähnt, wurden die neuen Daten mit zwei Smartphones (Samsung Galaxy S und einem LG Nexus 5) und der App “MyTrack“ (sieh Abbildung 3.4) aufgezeichnet. Neben der einfachen Handhabung bietet diese App auch an nur Punkte mit einer Mindestgenauigkeit aufzuzeichnen was in Folge beim Filtern ein wenig Arbeit abnimmt. Diese neuen Daten werden hauptsächlich zum Testen verwendet werden und nur ein Teil davon ist in die Trainingsdaten eingeflossen. Weiters kann man zwar ein Fortbewegungsmittel pro Aufzeichnung angeben, aber dies hat keinerlei Einfluss auf die Aufzeichnung selbst oder die Daten - es dient lediglich der visuellen Darstellung/Unterscheidung der einzelnen Aufzeichnungen.

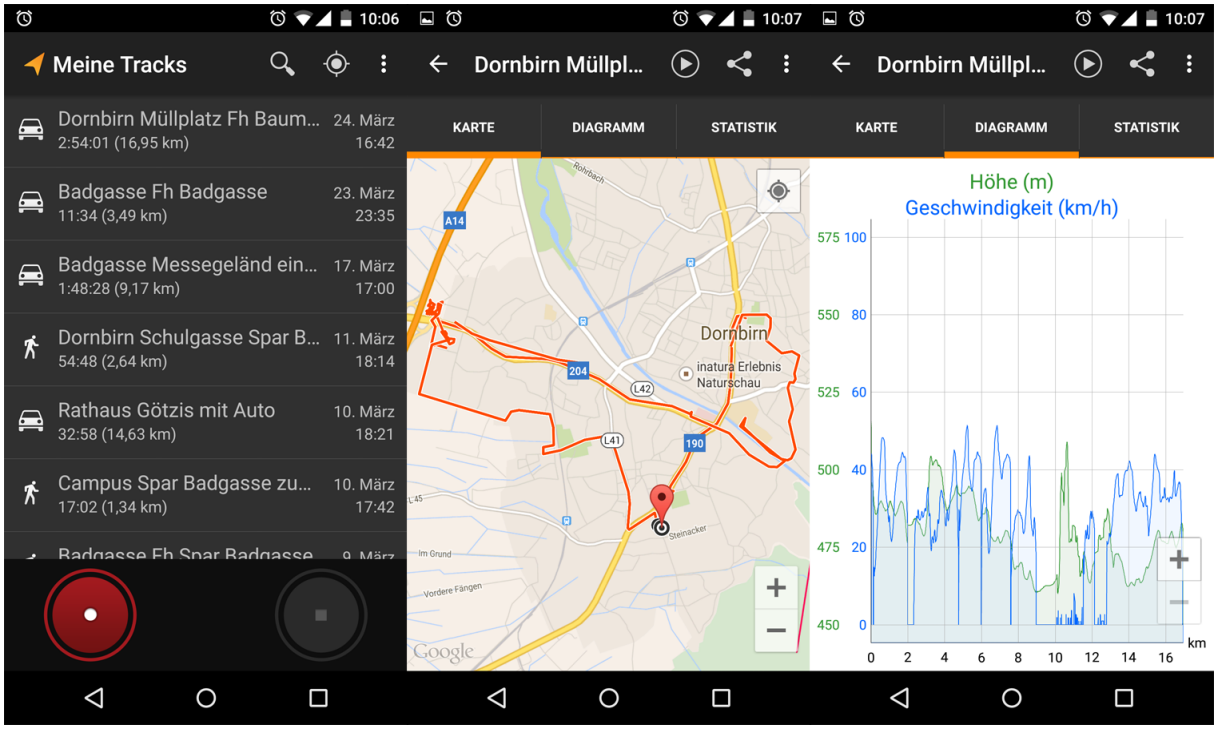


Abbildung 3.4: Die App myTrack

3.3 GIS-Daten

Als relevante GIS-Daten kommt laut den erwähnten Publikationen zum Thema Verkehrsmittelerkennung einiges in Frage wie z.B. Parkplätze, Busstationen, Gleise, Bahnhöfe und das gesamte Straßennetz. All diese Daten mögen zwar relevant sein, aber es handelt sich auch um sehr viele Daten was wiederum bedeutet, dass die Bearbeitungszeit einer Aufzeichnung rasch ansteigt. Da der Prototyp auch für konkrete EndbenutzerInnen interessant sein soll, wird auf die Verwendung des gesamten Straßennetzes und der Parkplätze verzichtet, um die Bearbeitungszeit möglichst gering zu halten. Deshalb wird auf die Verwendung von Busstationen und der Gleise gesetzt da dies schon in der Arbeit von Stenneth [Stenneth et al., 2011] zu guten Resultaten geführt hat. Dieser Ansatz wird ergänzt mit den GIS-Daten des Autobahnnetzes, da in diesen viel Potential vermutet wird und es sich um eine überschaubare Menge an Daten handelt.

Akquirierung
Allgemein sind GIS-Daten z.B. via OpenStreetMaps oder Google-Maps verfügbar, aber in einzelnen Stichproben hat sich herausgestellt, dass die Zusatzinformation in OpenStreetMaps wesentlich detaillierter und einfacher zum Extrahieren sind. Dafür wurde in Kauf genommen, dass diese Daten nicht standardisiert eingetragen wurden.

Die Österreich-Daten von OpenStreetMaps wurden als Archiv heruntergeladen und mit Hilfen von JOSM auf den relevanten Bereich eingegrenzt. JOSM ist ein Tool mit welchem die Daten von OpenStreetMaps gepflegt werden können (zu finden unter `https://josm.openstreetmap.de/`). Nachdem der Bereich auf Vorarlberg eingegrenzt worden ist, konnte dieser mit Hilfe von osmosis (weiteres Tool der OpenStreetMaps-Community `http://wiki.openstreetmap.org/wiki/Osmosis#Downloading`) nach bestimmten Punkten und Verbindungen gefiltert werden. Dadurch war es möglich, das Schienennetz von Vorarlberg sowie die Busstationen und das Autobahnnetz von Vorarlberg als XML-Datei zu exportieren. Diese XML-Dateien für Busstationen, das Autobahnnetz und das Schie-

nennetz wurden anschließend eingelesen und in die Datenbank importiert.

In weiterer Folge können diese Daten zur genaueren Bestimmung eines Verkehrsmitteltyps verwendet werden. Dazu werden Werte für die Nähe zu Bushaltestellen sowie Bahnhöfen bei Trackpoints ohne bzw. mit kaum Bewegung ermittelt. Außerdem wird in regelmäßigen Abständen die Nähe zu den Schienen und der Autobahn für jedes Segment berechnet. Diese Werte fließen dann in den Bestimmungsprozess ein und werden im Abschnitt “5.4 Berechnung der GIS-Entscheidungsvariablen“ genauer beschrieben.

3.4 Weitere Daten

Neben den zusätzlichen Werten, die aus den GPS-Spuren berechnet werden können und den GIS-Daten, wurde auch überlegt, Daten des öffentlichen Personennahverkehrs einzubinden, da diese in Vorarlberg die GPS-Tracks eines jeden Busses enthalten. Die Verwendung dieser Daten wäre insofern vielversprechend gewesen, als dass man einen ähnlichen Ansatz wie Stenneth verfolgen hätte können. Man hätte dadurch überprüfen können ob an der jeweiligen Stelle gerade ein Bus steht und darüber Rückschlüsse treffen können. Da diese Daten aber zum Zeitpunkt dieser Arbeit weder für diese Arbeit noch für die Öffentlichkeit verfügbar sind, konnte dieser Ansatz nicht weiter verfolgt werden.

3.5 Zusammenfassung

menfassung

Der Prototyp

Im Rahmen dieser Thesis wurde ein Prototyp für die Erkennung von Verkehrsmitteln aus GPS-Daten entwickelt. In Abstimmung mit den Zielen, dass es im Prinzip jeder Person (welche ein Gerät besitzt, welches GPS-Tracks aufzeichnen kann und Zugang zu einem Computer mit Internetzug hat) möglich sein soll, ihre eigenen Daten mit diesem Prototypen analysieren zu lassen, wurde eine Webapplikation entwickelt. Diese Applikation verwendet serverseitig das PHP-Framework Symfony als Basis und stellt eine REST-Schnittstelle zur Kommunikation zu Verfügung. Für die Persistierung wurde das Doctrine-ORM in Verbindung mit einer mySQL-Datenbank verwendet.

Der Prototyp bietet neben dem Filtern, Segmentieren und Analysieren der GPS-Daten auch eine Visualisierung der analysierten Daten an. Dies bedeutet, dass das Resultat der Verarbeitung auf einer Karte dargestellt wird. Zusätzlich zu der Darstellung der Karte wird dem Benutzer/den Benutzerinnen durch auch die Möglichkeit geboten, die Resultate des Prozesses zu bearbeiten und somit gegebenenfalls das Verkehrsmittel zu korrigieren.

Sowohl der vom System bestimmte als auch der vom Benutzer/von der Benutzerin korrigierte Verkehrsmitteltyp wird in der Datenbank abgelegt, um Aussagen über die Ge-

nauigkeit beider Verfahren (mit und ohne GIS-Daten) treffen zu können. Diese Auswertungen werden wiederum durch verschiedene Diagramme und Statistiken visualisiert.

4.1 Aufbau und Architektur

Der Kern der Webapplikation ist in einer Pipes-and-Filter Architektur aufgebaut. Diese Architektur bietet sich an, wenn man einen Datenstrom verarbeiten will. Dabei werden die einzelnen Schritte der Verarbeitung in sogenannte Filter unterteilt und diese werden mit Kanälen (Pipes) verbunden. Die Daten fließen somit in einen Filter, werden dort bearbeitet und schlussendlich über die verbunden Kanäle zu einem oder mehreren nachfolgenden Filtern weitergeleitet. Am Ende einer solchen Kette steht eine Senke (sink), in welcher die Daten schlussendlich abgelegt werden (z.B. eine Datei). Durch dieses Architekturmuster wird eine Codegliederung in einzelnen Komponenten/Arbeitsschritte in gewisser Weise bereits vorgegeben. Aber es ergeben sich auch andere Vorteile, wie z.B. Flexibilität bezüglich der Datenquelle, Austauschbarkeit von einzelnen Filterimplementierungen oder auch der Repräsentation und Persistierung der Endergebnisse. Weiters müssen bei diesem Ansatz auch keine Zwischendateien o.Ä. geschrieben werden. [Buschmann, 1998]

Diese Flexibilität, welches es ermöglicht, Filter bis zu einem gewissen Grad zu kombinieren und in verschieden Reihenfolgen zu verbinden, wurde für den Prototypen benötigt, um z.B. beide Analysemethoden möglichst einfach abbilden zu können. Außerdem werden im Falle dieses Prototyps die Trainingsdaten in eine Datei geschrieben und bei einer Anfrage über die Webapplikation werden die Daten in der Datenbank abgelegt. Hierbei unterscheiden sich nicht nur die Persistierungsarten sondern auch die Darstellung der Daten selbst, da in die Datei im CSV-Format geschrieben wird. Durch diesen Ansatz konnten viele Komponenten in anderen Benutzungsfällen wiederverwendet werden.

4.1.1 Pipes und Filter-Architektur für Trainingsdaten

Die Anordnung der einzelnen Filter für die Verarbeitung der GPS-Daten ist im Falle der Trainingsdaten in Abbildung 4.1 ersichtlich. Hierbei wird das Verzeichnis mit den Trai-

ningsdaten definiert und dem ersten Filter (FileReader) werden nach und nach die einzelnen Dateien übergeben. Nachdem die Daten den Trackpoint-, und Tracksegmentfilter (grün ohne GIS-Daten, orange mit GIS-Daten) durchlaufen haben werden sie schließlich von der letzten Komponente (FileWriter) in eine Datei geschrieben.

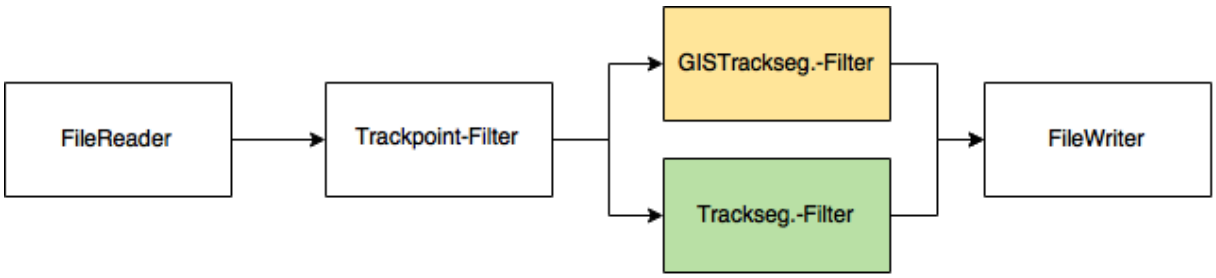


Abbildung 4.1: Pipes- und Filter- Struktur für Trainingsdaten

FileReader-Filter

Der FileReader-Filter liest die Datei mit dem angegebenen Dateinamen als XML-Datei ein, sucht sich alle Segmente (trkseg) aus dieser Datei und gibt diese schließlich an den nächsten Filter weiter. Als eingelesene Datei erwartet sich dieser Filter eine XML-Datei im GPX-Format.

Trackpoint-Filter

Der Trackpoint-Filter ist nicht nur ein Filter im Sinne des Architekturmusters, sondern auch im Sinne seiner Aufgabe. Er versucht all jene Trackpunkte (trkpt) herauszufiltern die nicht den konfigurierten Grenzwerten entsprechen. Dies bedeutet, dass all jene Trackpunkte gefiltert werden bei welchen der Zeitabstand, die Geschwindigkeit oder die Änderung im Bereich der Höhenmeter unter/über den minimalen/maximalen Grenzwerten liegen. Dadurch werden die einzelnen Segmente von den gröbsten fehlerhaften Ausreißern bereinigt, bevor sie an den nächsten Filter weitergeben werden.

Tracksegment-Filter

Der Tracksegment-Filter ist einer der wichtigsten Filter in dem Verarbeitungsprozess. Er berechnet für jedes Segment die für die weitere Verarbeitung benötigten Zusatzwerte. Dies bedeutet, dass er die mittlere und maximale Beschleunigung, die mittlere und maximale Geschwindigkeit, sowie die Stopprate berechnet. Weitere Informationen über die gewählten Zusatzwerte sowie die Auswahl selbst sind im Abschnitt “5.2 Schlussfolgerungsvariablen“ zu finden. Im Falle der Trainingsdaten ergänzt er die Segmentdaten noch mit dem in den Trainingsdaten definierten Verkehrsmitteltyp.

GISTracksegment-Filter

Der GISTracksegment-Filter macht im Wesentlichen das Selbe wie der normale Tracksegment-Filter, allerdings berechnet er zusätzlich zu den Geschwindigkeits- und Beschleunigungswerten noch die Abstände zu den verschiedenen Infrastrukturen wie Busstationen, Schienen und Autobahnen. Diese Werte werden bei der Analyse mit GIS-Daten benötigt um beim TravelMode-Filter eine genauere Bestimmung zu ermöglichen.

FileWriter

Der FileWriter übernimmt das Schreiben der übergebenen Daten in eine Datei. Hierbei werden die vom Tracksegment-Filter übergeben Segment in einer Datei im CSV-Format abgelegt, welche in weiterer Folge im Rapidminer verwendet wird.

4.1.2 Pipes und Filter-Architektur für die Webapplikation

Die Anordnung der einzelnen Filter für die Verarbeitung der GPS-Daten, die über die Webapplikation verarbeitet werden sollen, ist in Abbildung 4.2 ersichtlich.

Der Ablauf ist hierbei ähnlich wie jener bei den Trainingsdaten, und es wird wiederum zwischen den zwei Modi mit GIS-Daten (orange) oder ohne GIS-Daten (grün) unterschieden. Ausgehend vom FileReader- und Trackpoint-Filter kommen die Daten in den Segmentation-Filter. Von dort kommen sie, je nach Analyse-Art, in den Tracksegment-Filter oder in den GISTracksegment-Filter. Das Ergebnis dieser Filter kommt danach in den TravelModel-Filter, wo die eigentlichen Verkehrsmittel bestimmt werden. Anschließend werden die Ergebnisse dem Postprocess-Filter übergeben, welcher die bestimmten Verkehrsmitteln ein letztes Mal überprüft, um schlussendlich an den Database-Filter übergeben zu werden, welcher die Ergebnisse, für die Datenbank aufbereitet, in dieser ablegt.

Segmentation-Filter

Der Segmentation-Filter teilt die Tracksegmente der einzelnen Tracks in Teile, welche mit hoher Wahrscheinlichkeit nur mit einem Verkehrsmittel bewältigt wurden. Dazu verwendet dieser ermittelte Geschwindigkeits- sowie Beschleunigungswerte. Dieser Vorgang stützt sich auf die Ergebnisse aus den Publikationen von Zheng [Zheng et al., 2010] und Biljecki [Biljecki et al., 2013]. Der genaue Ablauf dieses Prozesses ist im Abschnitt 5.1 Segmentierung eines Tracks beschrieben.

TravelMode-Filter

In diesem Filter wird der eigentliche Verkehrsmitteltyp anhand der berechneten Werte bestimmt. Dies geschieht in beiden Fällen (mit und ohne GIS-Daten), mit Hilfe des Entscheidungsbaumes (siehe Abschnitt “3.1.2 Entscheidungsbaum ohne GIS-Daten“ und “3.1.2 Entscheidungsbaum mit GIS-Daten“) welcher mit Hilfe der jeweiligen Trainingsdaten generiert worden ist. Der genaue Ablauf dieses Prozesses ist im Abschnitt “5.5.2 Verwendung der Entscheidungsbäume“ beschrieben.

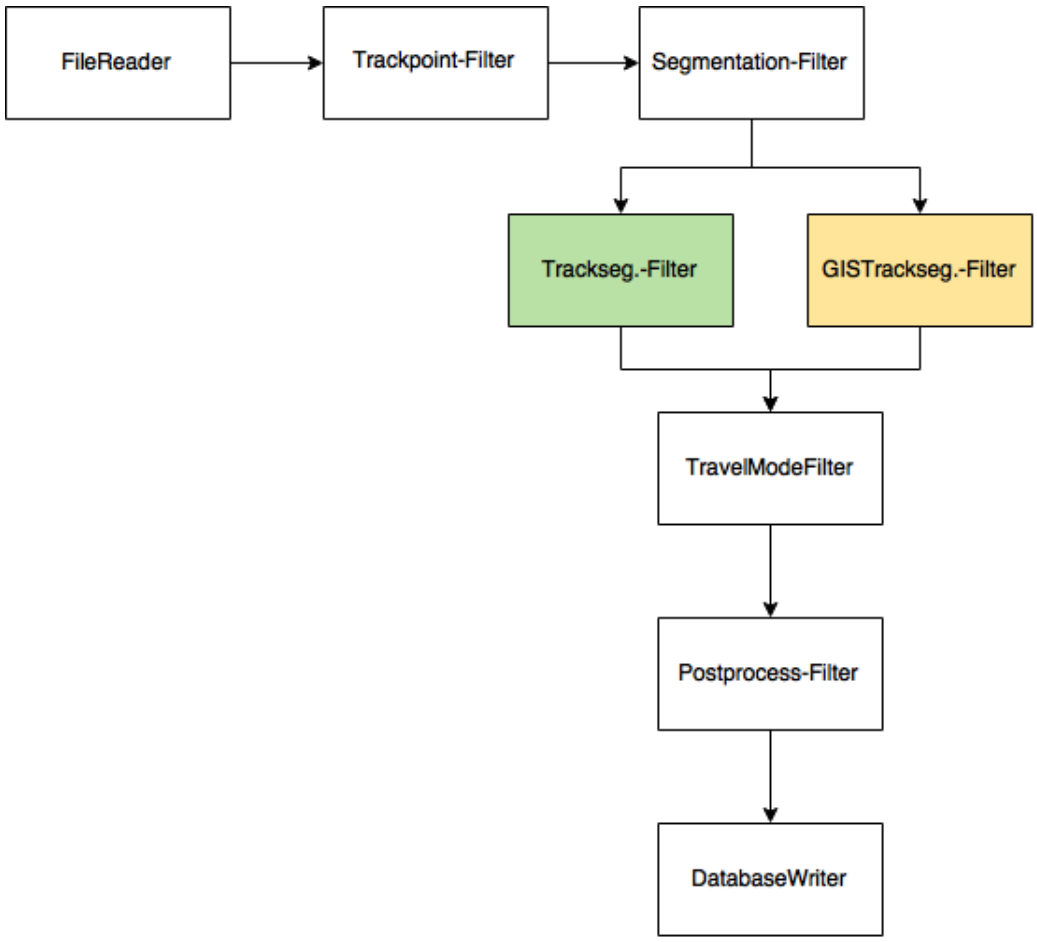


Abbildung 4.2: Pipes- und Filter- Struktur der Webapplikation

Postprocessing-Filter

Bei diesem Filter geht es darum, dass wenn alle Verkehrsmittel aller Segmente eines Tracks bestimmt worden sind, die Plausibilität der Wechsel nochmals überprüft wird. Diese letzte Überprüfung der Resultate der Entscheidungsbäume basiert auf Zheng [Zheng et al., 2010] und soll verhindern, dass sehr unwahrscheinliche Verkehrsmittelwechsels wie zum Beispiel “Auto->Bus->Auto->Bus->Auto“ entstehen. Basierend auf der Aussage von Zheng [Zheng et al., 2010], dass sich zwischen allen Verkehrsmittelwechsel ein (kleines) Segment vom Typ “zu Fuß“ befinden muss, können Wechsel wie

im obigen Beispiel verhindert werden. Somit wird das Resultat von “Auto->Bus->Auto->Bus->Auto“ zu “Auto->Auto->Auto->Auto->Auto“ korrigiert. Der genaue Ablauf dieses Prozesses ist im Abschnitt “5.5.3 Nachbearbeitung“ beschrieben.

DatabaseWriter

Da das Ablegen der ermittelten Ergebnisse in der Datenbank nur eine Variante (neben z.B. dem Ablegen in einer Datei) von vielen ist kümmert sich diese Komponente darum, dass die übergebenen Daten für die Datenbank aufbereitet und schlussendlich persistiert werden.

4.2 Verwendung der Applikation

Die Oberfläche der Webapplikation ist sehr einfach gehalten und besteht im wesentlich aus drei verschiedenen Seiten. Einer Startseite (Home), welche das Projekt kurz vorstellt, einer Seite (Create) auf welcher ein Track analysiert werden kann und einer Seite (Results), auf welcher die Resultate aller Auswertungen angezeigt werden.

4.2.1 Create-Seite

Auf dieser Seite wird dem Benutzer/der Benutzerin die Möglichkeit geboten, einen Track hochzuladen und diesen mit einem von den zwei Varianten (mit und ohne GIS-Daten) analysieren zu lassen. Ist dies geschehen, so wird eine Karte angezeigt, auf welche die ausgelesenen Segmente dargestellt werden. Diese Segmente werden, je nach ermitteltem Verkehrsmitteltyp, in verschiedenen Farben visualisiert. Per Klick auf eines dieser Segmente kann der Verkehrsmitteltyp des Segments korrigiert werden (siehe Abbildung 4.3). Diese Korrektur hat wiederum Einfluss auf die Auswertung der Ergebnisse, welche in Abbildung 4.4 ersichtlich sind.

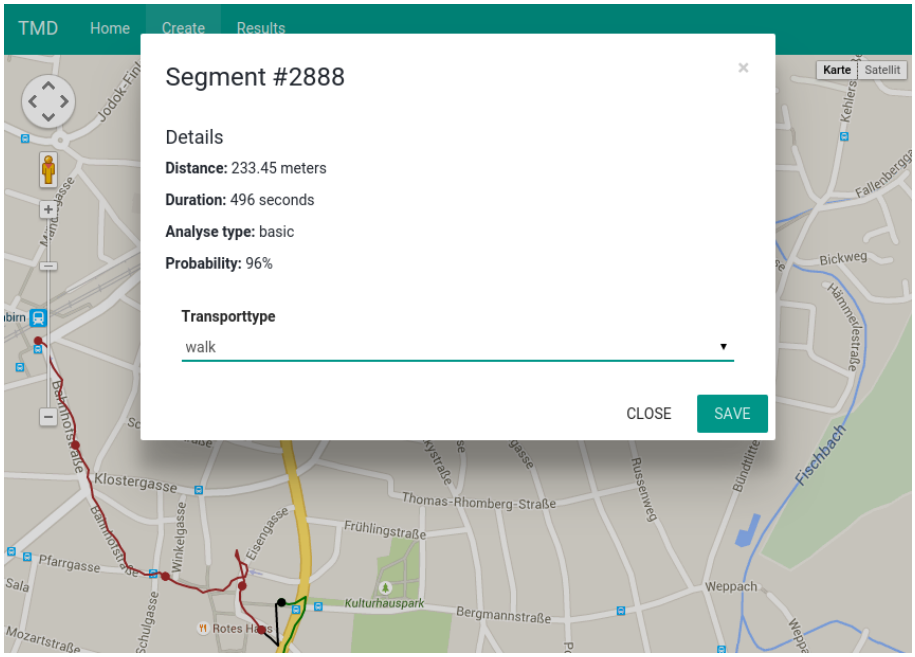


Abbildung 4.3: Korrigieren eines Verkehrsmitteltyps eines analysierten Tracks

4.2.2 Results-Seite

Die Resultate aller vorgenommenen Analysen und Änderungen werden auf dieser Seite als Diagramme dargestellt. Dabei zeigt das erste Diagramm, das Verhältnis aller analysierten und der korrekt erkannten Segmente. Das zweite Diagramm zeigt die Anzahl der korrekt analysierten Segmente und die Gesamtanzahl der Segmente je nach Verkehrsmittel. Alle weiteren Diagramme betreffen ein spezifisches Verkehrsmittel und geben darüber Auskunft, welche Verkehrsmittel eigentlich richtig gewesen wären.

Alle Diagramme zeigen dabei die Werte für beide verfügbaren Analysemethoden an, um einen direkten Vergleich zu ermöglichen. Ein Ausschnitt dieser Seite ist in Abbildung 4.4 zu sehen.

bild updat

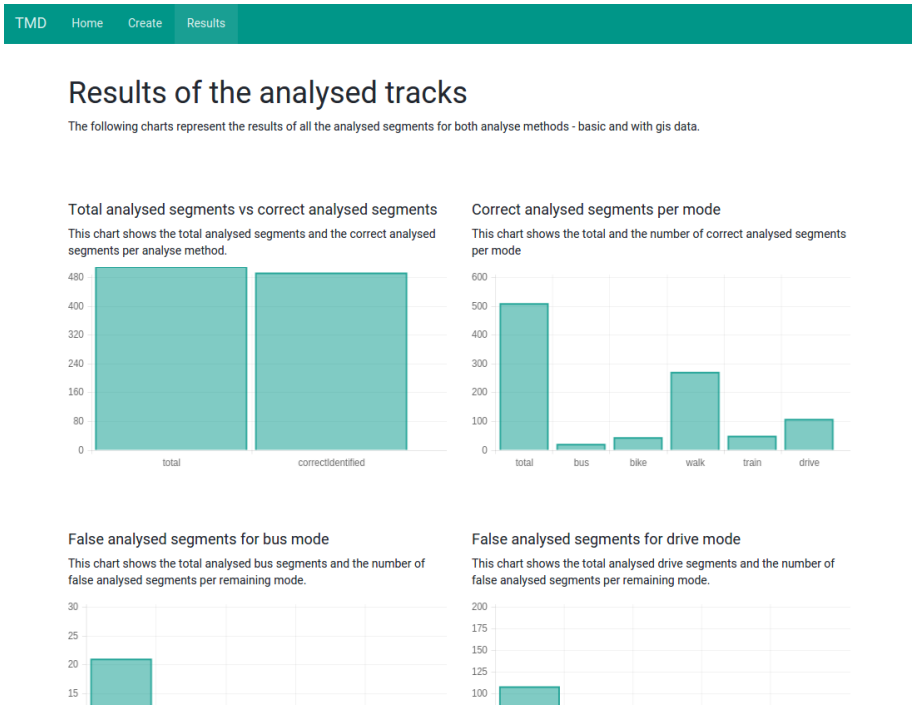


Abbildung 4.4: Ausschnitt zu den Resultaten der analysierten Tracks

4.3 Konfiguration des Prototyps

Der Prototyp lässt sich in den verschiedenen Bereichen weitestgehend konfigurieren. Diese Konfiguration kann in einer separaten Datei (app/config/config.yml) vorgenommen werden. Die Konfigurationsmöglichkeiten beinhalten unter anderem:

- Konfiguration des Default-Namespaces für das Parsen der GPX-Dateien, wenn keiner in der Datei gefunden wurde.
- Verschiedenste Grenzwerte zum beeinflussen des Filterns von fehlerhaften Ausreißern in den Trackpunkten.
- Diverse Grenzwerte für den Segmentierungsprozess, welcher die einzelnen Tracks in Segmente mit nur einem Verkehrsmittel unterteilt.
- Definitionsmöglichkeiten für alle Analysemethoden sowie deren Eigenschaften.

```
filter:
  min_distance: 0.1
  max_distance: 50
  max_altitude_change: 25
  min_trackpoints_per_segment: 2
  min_time_difference: 2
  points_to_skip_from_start: 2
  min_valid_points_ratio: 0
```

Listing 4.1: Filterkonfiguration

4.3.1 Konfiguration des Filters für fehlerhafte Ausreißer

In diesem Konfigurationsbereich (siehe Listing 4.1) inkludiert sind sowohl eine minimale als auch maximale Distanz in Metern pro Zeiteinheit sowie ein Maximalwert für die Änderung der Höhenmeter. Weiters kann hier definiert werden, wie hoch die minimale Anzahl der Trackpunkte pro Segment sein muss, wie groß die minimale Zeitdifferenz (in Sekunden) zwischen zwei Trackpunkten sein soll und wie viele Punkte am Start übersprungen werden sollen. Außerdem kann ein Prozentwert für die Anzahl der validen Trackpunkte im Verhältnis zu allen Trackpunkten eingestellt werden. Eine genauere Erklärung zu dem Filterprozess ist im Anhang 1 zu finden.

4.3.2 Konfiguration des Segmentierens

In dem Abschnitt der Segmentierungskonfiguration (siehe Listing 4.2) kann festgelegt werden, bis zu welcher Geschwindigkeit (m/s) und Beschleunigung (m/s^2) ein Wegpunkt als Geh-Punkt gilt. Weiters kann festgelegt werden, welches die minimale Zeitspanne (in Sekunden) und die minimale Distanz (in Metern) ist, die ein Segment haben muss, um nicht mit dem vorangegangenen Segment vereint zu werden. Schlussendlich gibt es drei Werte, welche zu dem Beenden eines Segments führen können. Darunter ist sowohl ein Stopp, welcher durch eine sehr geringe oder gar keine Bewegung über

```
segmentation:
  max_walk_velocity: 2.78
  max_walk_acceleration: 1.5

  min_segment_time: 20
  min_segment_distance: 50
  max_time_difference: 30

  max_time_without_movement: 10
  max_velocity_for_nearly_stoppoints: 0.55

  certain_segments_min_time: 60
  certain_segments_min_distance: 100
```

Listing 4.2: Segmentierungskonfiguration

eine bestimmte Zeitspanne (in Sekunden) definiert wird oder durch eine Zeitspanne (in Sekunden), in welcher keine neuen Trackpunkte gefunden werden. Wofür diese Werte benötigt werden, wird im Abschnitt “5.1 Segmentierung eines Tracks“ detailliert erklärt. Die letzten zwei Konfigurationsvariablen beschreiben die minimale Zeit und Distanz die ein Segment haben muss um vom Status eines ungewissen Segments in den Status eines sicheren übergehen zu können.

4.3.3 Konfiguration der Analysemethoden

Für die Analyse können verschiedenste Methoden definiert werden. In Listing 4.3 ist die Konfiguration für die Analysemethoden ohne GIS-Daten abgebildet. Hierbei werden unter einem Namen für die Analysemethode (basic) verschiedene Konfigurationsvariablen abgelegt. Darunter zum einen Variablen für die Erstellung des Entscheidungsbaums sowie der Generierung der Datei mit den Trainingsdaten. Die Konfiguration für den Entscheidungsbaum umfasst einen Klassennamen (zugleich auch Dateiname), das

```
1     basic:
2       class: "BasicDecisionTree"
3       cacheDir: "%kernel.root_dir%/../decisionTrees/basic"
4       txtFilePath: "%kernel.root_dir%/../decisionTrees/basic"
5       txtFileName: "basicDecisionTree.txt"
6       csv_columns:
7         - "stoprate"
8         - "meanvelocity"
9         - "meanacceleration"
10        - "maxvelocity"
11        - "maxacceleration"
```

Listing 4.3: Analysekonfiguration

Verzeichnis, in welchem die generierte Entscheidungsbaum-Datei abgelegt werden soll, sowie den Pfad wo die Textdatei mit der Definition des Entscheidungsbaums gefunden wird. Wozu diese Werte benötigt werden wird im Abschnitt “5.5.1 Erstellen der Entscheidungsbäume“ genauer erklärt. Hinter der Konfigurationsvariable “csv_columns“ verstecken sich die Spalten welche beim Erstellen der Trainingsdatendatei verwendet werden und diese definieren somit auch welche Entscheidungsvariablen im Entscheidungsbaum verwendet werden.

Segmentierung und Klassifizierung

In diesem Abschnitt wird der Prozess der Klassifizierung von einzelnen Segmenten erklärt und er inkludiert den Segmentierungsvorgang sowie die Berechnung der Zusatzinformation, welche für die Segmentierung benötigt werden. Weiters wird dargelegt, wie die Entscheidungsvariablen für die Entscheidungsbäume ausgewählt worden sind.

Unter dem Segmentierungsvorgang versteht man das Aufteilen einer GPS-Spur in Abschnitte, in welchen mit hoher Wahrscheinlichkeit nur ein Verkehrsmittel benutzt worden ist. Dabei wird versucht, die Segmente zu finden, in welchen man zu Fuß unterwegs war. Diese werden dann als Geh-Segmente gekennzeichnet. Die anderen Segmente werden als nicht Geh-Segmente klassifiziert. Dadurch, dass hierbei nur zwischen zwei temporären "Verkehrsmitteln" unterschieden wird, vereinfacht sich die Verkehrsmittelbestimmung drastisch.

Nach der Segmentierung folgt die konkrete Bestimmung des jeweiligen Verkehrsmittels für die einzelnen Abschnitte, welche aus dem Segmentierungsvorgang hervorgegangen sind. In dieser Arbeit wurde die Bestimmung mit Hilfe von Entscheidungsbäumen realisiert. Dabei gibt es einen Entscheidungsbaum für die Klassifizierung mit und einen für die Klassifizierung ohne GIS-Daten. Für diese Entscheidungsbäume wurden verschiedene Entscheidungsvariablen ausgewählt. Sowohl die Auswahl als auch die Berechnung

dieser Werte wird erklärt.

Schlussendlich werden die klassifizierten Abschnitte ein letztes mal auf ihre Plausibilität in dieser Reihenfolge überprüft. Dabei stützt sich dieser Prozess auf die Aussage von Zheng [Zheng et al., 2010], nach welcher sich immer - wenn mit unter auch kurze - Geh-Segmente zwischen Verkehrswechsel befinden müssen und man nicht von einem in ein anderes Verkehrsmittel wechseln kann ohne eine kurz Strecke zu Fuß bewegen zu müssen. Durch miteinbeziehen des Kontexts lassen sich Abfolgen wie z.B. “Auto->Bus->Auto->Bus->Auto“ verhindern.

5.1 Segmentierung eines Tracks

Bei der Segmentierung geht es in erster Linie darum, den Umfang des Problems der Verkehrsmittelbestimmung zu verkleinern. Dies bedeutet, man versucht das angegebene GPX-Tracksegment in Geh-Segmente und nicht Geh-Segmente aufzuteilen und dadurch statt zwischen mehreren Verkehrsmitteln (Gehen, Bus, Zug, Auto, Fahrrad) nur mehr zwischen zwei unterscheiden muss. Die genauere Unterscheidung kann dann in einem weiteren Schritt folgen, in welchem jedoch schon klar ist, wo es sich um ein Segment handelt in welchem eine Person zu Fuß unterwegs war oder nicht.

Zheng sagt in diesem Zusammenhang, dass es zwischen jedem Wechsel eines Verkehrsmittels einen Abschnitt gibt in welchem man zu Fuß unterwegs war und ein Stopp stattgefunden hat, auch wenn dieser Abschnitt sehr klein ist. Ein Wechsel erfolgt nie direkt wie z.B. von einem Zug in den Bus ohne Anzuhalten und über einen Bahnsteig gehen zu müssen. Dies bedeutet, dass wenn ein Verkehrsmittelwechsel stattgefunden hat, dann gibt es neben Geh-Punkten auch Trackpunkte, in welchen eine Geschwindigkeit und eine Beschleunigung von nahezu 0 zu erwarten ist (=> Stopp). Diese Aussage hat sich aus seinen umfangreichen Testdaten ableiten lassen. Daraus hat er folgenden Algorithmus abgeleitet: [Zheng et al., 2010]

- Finde alle Geh-Punkte und nicht Geh-Punkte des betrachteten Abschnitts oder Tracks anhand von Grenzwerten für Geschwindigkeit und Beschleunigung und fasse die aufeinander folgenden Punkte vom selben Typ in Segmente zusammen.
- Liegt die Distanz oder die Zeit eines solchen Segmente unterhalb einer definierten Grenze so vereine dieses Segment mit dem Vorherigen.
- Überschreitet ein Segment eine bestimmte Länge (200 Meter), ist es ein “sicheres Segment“. Liegt die Distanz eines Segments jedoch unterhalb dieses Werts so ist es ein “unsicheres Segment“. Überschreitet die Anzahl der aufeinander folgenden “unsicheren Segmente“ einen bestimmten Grenzwert, so werden die aufeinander

folgenden “unsicheren Segmente“ vereint und als nicht-geh-Segment betrachtet.

Biljeki ergänzt diesen Ansatz von Zheng mit seinen Erfahrungen in welchen er feststellte, dass bei einem Wechsel des Verkehrsmittels auch oft das Signal verloren geht. Aus diesem Grund beendet Biljecki ein Segment, wenn die Verbindung verloren wurde und startet ein Neues. Den Verbindungsverlust interpretiert Biljecki so, dass er für mindestens 30 Sekunden keinen weiteren Trackpoint findet. Weiters segmentiert Biljecki dann, wenn er für mehr als 12 Sekunden keine bzw. wenig Bewegung (< 2km/h) feststellen konnte. Diese beiden Änderungen an dem Algorithmus von Zheng begründet Biljecki damit, dass eine Übersegmentierung besser ist als eine Untersegmentierung. Aufeinander folgende Segmente mit demselben Typ kann man immer noch in einem nächsten Schritt zusammenlegen. [Biljecki et al., 2013]

Für den Prototypen wurde im Wesentlichen der durch Biljecki erweiterte Ansatz von Zheng gewählt mit der Einschränkung, dass die Klassifizierung in “sichere“ und “unsichere“ Segmente nicht übernommen wurde. Dies wurde mit der Aussage von Biljecki begründet, dass eine Übersegmentierung besser ist als eine Untersegmentierung. Das Resultat eines Segmentierungsvorgangs kann man in Abbildung 5.1 sehen. Jeder Kreis auf der eingezeichneten Route zeigt den Start bzw. das Ende eines Segments. Auf Basis dieser ersten Aufteilung kann nun die genauere Bestimmung des Verkehrsmittels erfolgen.

/ uncer-
gments

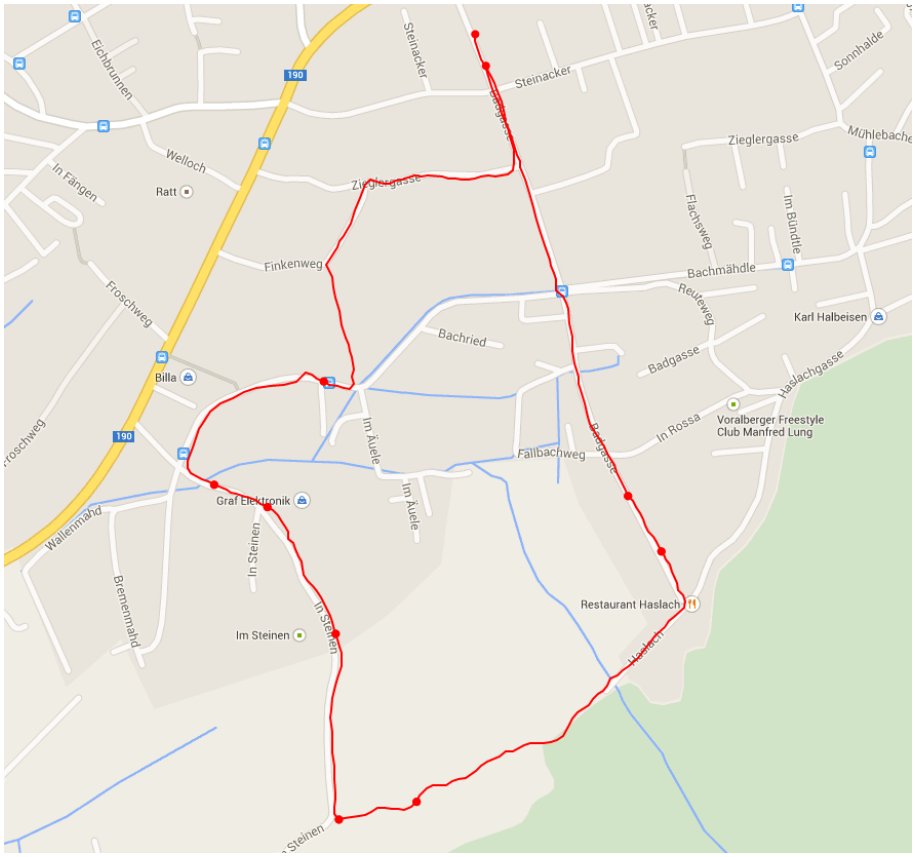


Abbildung 5.1: Grundlegende Segmentierung

5.2 Schlussfolgerungsvariablen

nd varia-
n diesem
menhang

Als Grundlage für die Schlussfolgerung durch die Entscheidungsbäume müssen zuerst Variablen, die für die Schlussfolgerung möglichst ausschlaggebend für die betrachteten Verkehrsmittel sind, festgelegt werden. Dies Variablen können dabei diverse geschwindigkeitsabhängige Werte, wie durchschnittliche und maximale Geschwindigkeit oder auch Beschleunigungswerte und Abstände zu bestimmten Infrastrukturen, sein. Eine Übersicht über die Schlussfolgerungsvariablen in den betrachteten Publikationen ist in der Tabelle 5.1 abgebildet. Diese Werte sind nach der Anzahl der Vorkommnisse in den Publikationen gereiht und bildet eine Grundlage für die in dieser Arbeit verwendeten Variablen.

Aus Gründen der Vollständigkeit muss noch erwähnt werden, dass aufgrund eines anderen Segmentierungsverfahrens nicht alle Variablen von Gonzales [Gonzalez et al., 2010] aufgeführt sind. Außerdem handelt es sich bei der maximalen Geschwindigkeit nicht um das absolute Maximum sondern um 95% davon bzw. ist es die dritt höchste Geschwindigkeit die gemessen wurde.

Weiters muss noch erwähnt werden, dass sowohl Zheng [Zheng et al., 2010] als auch Stenneth [Stenneth et al., 2011] nicht alle Variablen schlussendlich verwendet haben. Sie haben allerdings evaluiert, welche Variablen in ihrem Projekt die größte Wirkung zeigen. Dies war bei Zheng [Zheng et al., 2010] die Kombination von der Stopprate, der Geschwindigkeitsänderungsrate und der Richtungsänderungsrate. Fügte er weitere Variablen hinzu, konnte er eine Verschlechterung der Ergebnisse beobachten. Bei Stenneth [Stenneth et al., 2011] hat die Evaluierung ergeben, dass die durchschnittliche Geschwindigkeit und Beschleunigung kombiniert mit verschiedenen GIS-Werten das beste Ergebnis geliefert.

Die betrachteten GIS-Werte waren bei Stenneth [Stenneth et al., 2011] der durchschnittliche Abstand zu Gleisen, Bussen und dem Buskandidat (jener Bus in dem sich die Person am ehesten Befand). Biljecki verwendete den Abstand zu Gleisen, Bushaltestellen, Buslinien, U-Bahn und Straßenbahn als Indikatoren für die Schlussfolgerung. Schüssler [Nadine Schüssler et al., 2011] inkludiert von den möglichen GIS-Werten nur den Abstand zu öffentlichen Verkehrsmitteln aller Art.

5.2.1 Reihung der allgemeinen Variablen

Die in dieser Arbeit verwendeten Schlussfolgerungsvariablen basieren auf der Reihung welche in Tabelle 5.1 ersichtlich ist. Die allgemeine Geschwindigkeit an fünfter Stelle wurde deshalb übersprungen, weil sie bereits zwei Mal vertreten ist und bereits Zheng gesagt hat, dass die Geschwindigkeit allein kein aussagekräftiger Indikator für ein Verkehrsmittel ist [Zheng et al., 2010]. Weiters hat Zheng für die Auswahl der Stopprate eine interessante Erklärung, welche im Abschnitt “5.3.3 Stopprate“ genauer erklärt wird und weshalb die Stopprate auch als fünfte Variable ausgewählt wurde.

1. durchschnittliche Geschwindigkeit
2. maximale Geschwindigkeit
3. durchschnittliche Beschleunigung
4. maximale Beschleunigung
5. Stopprate

5.2.2 Reihung der GIS-Variablen

Bei der Auswahl der GIS-Variablen fallen jene mit U-Bahn und Straßenbahn weg, da es diese im Raum Vorarlberg nicht gibt. Jene Variablen, die spezifische GPS-Daten von einzelnen Verkehrsmitteln benötigen, konnten auch Aufgrund von fehlenden Schnittstellen

zu den Daten der öffentlichen Verkehrsbetriebe nicht verwendet werden. Somit wurden folgende Variablen für die GIS-gestützte Analyse verwendet:

- durchschnittliche Nähe zu Busstationen und Bahnhöfen
- durchschnittliche Nähe zu Gleisen
- durchschnittliche Nähe zu Autobahnen

	Zheng[1]	Stenneth[2]	Reddy[3]	Biljecki[4]	Gonzales[5]	Schüssler[6]	**Gesamt**
durchschn. Geschwindigkeit	x	x		x	x	x	**5**
max. Geschwindigkeit *	x			x	x		**3**
durchschn. Beschleunigung		x	x		x		**3**
verwendet GIS Daten		x		x		x	**3**
Geschwindigkeit			x			x	**2**
max. Beschleunigung	x				x		**2**
Richtungswechselrate	x	x					**2**
Stopprate	x						**1**
Distanz des Segments	x						**1**
Distanz des Tracks					x		**1**
Geschwindigkeitswechselrate	x						**1**
durchschn. Varianz d. Beschl.						x	**1**
durchschn. beweg. Geschw.				x			**1**
durchschn. Genauigkeit		x					**1**
erwartete Geschwindigkeit	x						**1**
Varianz d. Geschwindigkeit	x						**1**

Tabelle 5.1: Entscheidungsvariablenübersicht

[1] [Zheng et al., 2010], [2] [Stenneth et al., 2011], [3] [Reddy et al., 2010], [4] [Biljecki et al., 2013], [5] [Gonzalez et al., 2010], [6] [Nadine Schüssler et al., 2011]

5.3 Berechnung der allgemeinen Entscheidungsvariablen

Das Hinzufügen und Berechnen (z.B. Geschwindigkeit aus Weg und Zeit) der Entscheidungsvariablen ohne GIS-Daten, wird von dem Tracksegment-Filter (siehe Abschnitt "4.1.1 Tracksegment-Filter"), realisiert. Dabei ist die Berechnung der grundlegenden Geschwindigkeit zwischen zwei GPS-Punkten essentiell, da alle weiteren Entscheidungsvariablen darauf aufbauen.

5.3.1 Geschwindigkeit

Berechnet wurde die Geschwindigkeit (v) als Durchschnittsgeschwindigkeit, die als Verhältnis vom zurückgelegten Weg (s) zu der Zeit (t) ausgedrückt wird, wie es z.B. im Buch "Physik" von Douglas Giancoli definiert wird [Douglas C. Giancoli, 2010, S. 27] und in 5.1 ersichtlich ist.

$$v = \frac{s}{t} \tag{5.1}$$

Die Zeit ist dabei die Differenz zwischen den Zeitstempeln von zwei GPS-Trackpunkten und der Weg die Differenz zwischen den zwei Koordinaten. Zur Berechnung des Weges (der Luftlinienentfernung) zwischen zwei Koordinaten gibt es laut [Movable Type Ltd, 2015] mehrere Möglichkeiten. Welche man verwendet hängt von den Längen der betrachteten Strecken, der gewünschten Genauigkeit sowie der benötigten Performanz ab (schnell aber ungenau vs. langsam aber genau). Da es sich bei den hier betrachteten Distanzen immer um sehr kleine Distanzen im Verhältnis zur Erde selbst handelt, aber diese aufgrund ihrer weiteren Verwendung zur Berechnung der Geschwindigkeit möglichst genau sein soll wird in diesem Prototyp die Haversine-Formel (siehe die Gleichungen 5.2) verwendet. Dabei sind φ die Breitengrade, λ die Längengrade, R der Erdradius und $\triangle\varphi$ bzw. $\triangle\lambda$ die Differenz der Breiten- bzw. Längengrade. [Movable Type Ltd, 2015]

$$
\begin{aligned}
a &= sin^2(\triangle\varphi/2) + cos\varphi_1 * cos\varphi_2 * sin^2(\triangle\lambda/2) \\
c &= 2 * atan2(\sqrt{a}, \sqrt{(1-a)}) \\
d &= R * c
\end{aligned}
\tag{5.2}
$$

Durchschnittliche Geschwindigkeit Die durchschnittliche Geschwindigkeit wird über alle Geschwindigkeitswerte eines Segments berechnet.

Maximale Geschwindigkeit Die maximale Geschwindigkeit wird aus allen Geschwindigkeitswerten des Segments ermittelt.

5.3.2 Beschleunigung

Die Beschleunigung ist als Geschwindigkeitsunterschied zwischen zwei Punkten pro Zeiteinheit definiert [Douglas C. Giancoli, 2010, S. 51]. Somit wurde diese auf Basis der ermittelten Geschwindigkeit berechnet und dadurch folgende zwei Entscheidungsvariablen bestimmt.

- **Durchschnittliche Beschleunigung** Die durchschnittliche Beschleunigung für das betrachtete Segment.
- **Maximale Beschleunigung** Die maximal gemessene Beschleunigung für das betrachtete Segment.

5.3.3 Stopprate

Wie Zheng in [Zheng et al., 2010] beschrieben hat, ist die Geschwindigkeit als Basis für Entscheidungsvariablen nur bedingt geeignet, da diese sehr von der aktuellen Verkehrssituation abhängig ist. Deshalb setzt Zheng auf die Kombination von Richtungswechsel,

Geschwindigkeitswechsel und der Stopprate. Dieser Aussage widerspricht wiederum indirekt die Auswahl der Entscheidungsvariablen von anderen Autoren wie Biljecki [Biljecki et al., 2013], Gonzales [Gonzalez et al., 2010], Schüssler [Nadine Schüssler et al., 2011] und Reddy [Reddy et al., 2010] (siehe Tabelle 5.1).

Die Stopprate ist für Zheng insofern sehr aussagekräftig, da er an spezifischen Verkehrsmitteln auch eine spezifische Stopprate bzw. ein Stoppverhalten festmachen kann. Dies ist Beispielhaft in Abbildung 5.2 ersichtlich. Dabei stoppt eine Person im Auto (a) wesentlich weniger oft wie zum Beispiel ein Bus (b) da dieser nicht nur bei Kreuzungen sondern auch bei Bushaltestellen anhalten muss. Noch öfter stoppt laut Zheng nur ein Fußgänger (c) [Zheng et al., 2010]. Im Falle des Prototyps wird die Erkennung von Stopps bereits beim Segmentieren benötigt und dadurch, dass die anderen Entscheidungsvariablen sehr auf der Geschwindigkeit basieren, wurde die Stopprate als weitere Entscheidungsvariable, die nicht abhängig von der Geschwindigkeit ist ausgewählt. Als Stopp gelten dabei eine bestimmte Anzahl von Trackpunkten (z.B. über eine Zeit von 5 Sekunden), bei welchen die Geschwindigkeit unterhalb einem Grenzwert (z.B. unter 2km/h) liegt. Diese Werte können wiederum in der Konfiguration eingestellt werden.

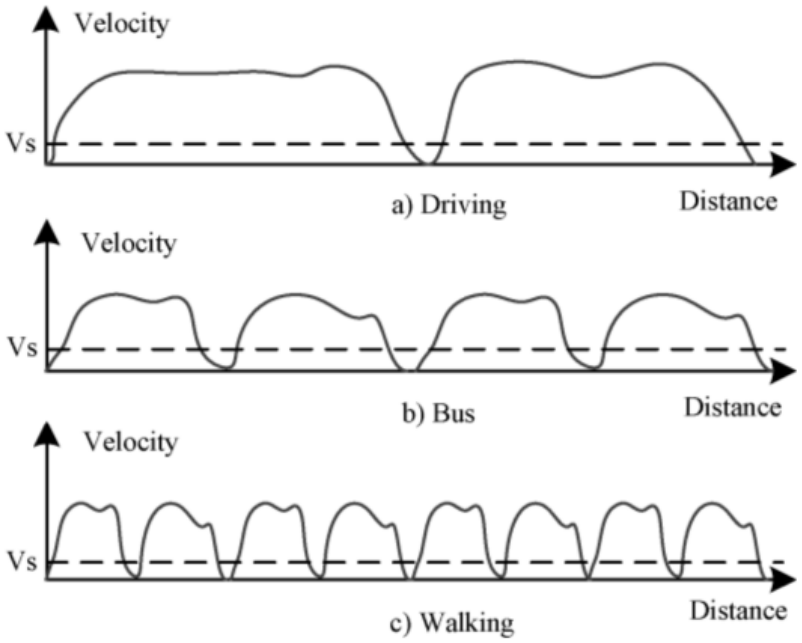


Abbildung 5.2: Stopprate laut Zheng (Quelle: [Zheng et al., 2010])

5.4 Berechnung der GIS-Entscheidungsvariablen

Das Hinzufügen und Berechnen der Entscheidungsvariablen mit GIS-Daten, wird von dem GISTracksegment-Filter (siehe Abschnitt “4.1.1 GISTracksegment-Filter“), realisiert. Dabei erweiter der GISTracksegment-Filter den Tracksegment-Filter und ergänzt diesen um die folgenden weiteren GIS-Werte. Bei allen GIS-Werten wird eine Bounding-Box berechnet und auf die zuvor in die Datenbank importierten GIS-Daten zugegriffen. Der Radius der Boundingbox lässt sich in der Konfiguration festlegen.

5.4.1 Abstand zu Bushaltestellen

Da der Beginn und das Ende eines Segments immer ein Stopp ist, wird hierbei auch überprüft ob sich eine Bushaltestelle in der Nähe befindet. Wird eine Bushaltestelle gefunden, so wird diese in den GIS-Wert für die Bestimmung des Verkehrsmittels miteinbezogen. Ist sowohl Beginn als auch Ende eines Segments in der Nähe einer Bushaltestelle so wird dies höher gewichtet. Innerhalb eines Segments können sich weitere kürzere Stopps befinden. Diese werden zusätzlich verwendet um festzustellen ob Bushaltestellen in der Nähe sind, und es sich im Endeffekt um einen Bus handelt er von Station zu Station fährt. Im Gegensatz zu den Haltestellen am Beginn und am Ende werden diese, allerdings nicht höher gewichtet, da sich Bushaltestellen oft in der Nähe von Ampeln oder Kreuzungen befinden. Dies stellte auch Biljecki [Biljecki et al., 2013] schon fest.

5.4.2 Abstand zu Gleisen und zur Autobahn

Ähnlich wie bei den Bushaltestellen werden auch die GIS-Werte für die Nähe zu Gleisen und der Autobahn am Beginn und am Ende eines jeden Segments berechnet. Allerdings

wird für die Überprüfung innerhalb eines Segments ein konfigurierbarer Zeitwert verwendet. Im Prototypen wird alle 20 Sekunden überprüft ob sich ein Punkt in der Nähe eines Gleises oder der Autobahn befindet.

5.5 Klassifizierung der Verkehrsmittel

Die Klassifizierung der Segmente bzw. das Bestimmen des Verkehrsmittels für ein Segment basiert bei beiden Analysearten (mit und ohne GIS-Daten) auf Entscheidungsbäumen. Wie bereits in Abbildung 3.2 und in Abbildung 3.3 ersichtlich und im Abschnitt “5.2 Schlussfolgerungsvariablen“ genau beschrieben, verwendet der Entscheidungsbaum ohne GIS-Daten ausschließlich Geschwindigkeits- und Beschleunigungswerte sowie die Stopprate als Indikatoren. Der Entscheidungsbaum mit GIS-Daten fügt diesen aber noch einen Wert für die Nahe zu verschieden Infrastrukturen hinzu.

5.5.1 Erstellen der Entscheidungsbäume

Wie bereits im Abschnitt “3.1.2 Entscheidungsbaum als Schlussfolgerungsmodell“ beschrieben, konnte in der freien Version von Rapidminer der generierte Entscheidungsbaum als Bild oder Text exportiert werden. Um jedoch bei dem Erstellen des Entscheidungsbaums möglichst flexibel zu sein (z.B. erweitern der Trainingsdaten) wurde ein Parser für die textuelle Darstellung des Entscheidungsbaums implementiert.

Ein Ausschnitt des Entscheidungsbaums als Text ist in Listing 5.1 zu sehen. Dabei stellt jede Zeile mindestens einen Knoten im Baum dar. Eine Zeile kann mit Indikatoren für die Tiefe des Knotens im Baum beginnen. Dabei steht jedes “|“ für eine Ebene tiefer im Baum. Jene zwei Zeilen am Beginn ohne Tiefenangabe bilden somit den Wurzelknoten, da sie über keine Tiefenangabe verfügen. Außerdem gibt es jeden Knoten, der kein Blatt ist, zwei Mal in dieser Darstellung. Diese beiden Knoten unterscheiden sich dabei nur durch ihren Vergleichsoperator, denn dieser ist genau das Gegenteil des Anderen. Dadurch können beide Fälle einer Bedingung, abgebildet werden.

Jede Zeile besteht aus dem Namen der Entscheidungsvariable, einem Vergleichsoperator und einem Wert. Hat ein Knoten nur mehr einen Kind-Knoten mit dem Resultat so folgt hinter dem Wert das Resultat. Das Resultat besteht dabei aus dem bestimmten

```
1 meanvelocity > 20.830: train {bike=0, walk=0, car=0, bus=0, train=5}
2 meanvelocity <= 20.830
3 |   meanvelocity > 2.041
4 |   |   meanvelocity > 7.837
5 |   |   |   meanvelocity > 8.772: car {bike=0, walk=0, car=46, bus=1,
    train=2}
6 |   |   |   meanvelocity <= 8.772
7 |   |   |   |   meanacceleration > 2.728: bus {bike=0, walk=0, car=0,
    bus=2, train=0}
8 |   |   |   |   meanacceleration <= 2.728: car {bike=0, walk=0, car=3,
    bus=0, train=0}
9 ...
```

Listing 5.1: Entscheidungsbaum in Textform

Verkehrsmittel und einer Übersicht über die Vorkommnisse aller Verkehrsmittel für die getroffenen Entscheidungen aus den Trainingsdaten.

Alle Kind-Knoten sind jeweils dem letzten Knoten auf dem vorherigen Level zuzuordnen.

Parsen und Cachen der Entscheidungsbäume

Die von Rapidminer gelieferten Entscheidungsbäume wurden im Prototypen als Teil der Konfiguration hinterlegt. Damit die Entscheidungsbäume nicht für jeden Anfrage geparst und erstellt werden müssen, bietet das Framework Symfony die Möglichkeit, die aus Konfigurationsdateien entstandenen Resultate zu cachen. Dies bedeutet in diesem Fall, dass die Entscheidungsbäume eingelesen sowie geparst werden und danach mit Hilfe des Resultats eine PHP-Datei erstellt wird. Darin wird die Initialisierung des Entscheidungsbaums als Code abgelegt (siehe Listing 5.2). Dadurch muss der Entscheidungsbaum nicht jedes mal neu geparst sondern nur ein Objekt mit genau diesem Baum instanziiert werden.

Damit dieser Vorgang nur gemacht wird, wenn sich die Datei mit dem Entscheidungs-

```
1  ...
2  class BasicDecisionTree implements DecisionTreeInterface
3  {
4      protected $tree;
5
6      function __construct()
7      {
8          $node0 = new Node();
9          $node0->setDecision(new Decision('meanvelocity', '>', 20.83));
10         $node1 = new Node();
11         $node1->setResult(new Result(0,0,0,0,5));
12         ...
13         $node0->setRight($node2);
14         $node1->setParent($node1);
15         $node2->setParent($node0);
16         $node2->setLeft($node3);
17         ...
```

Listing 5.2: Ausschnitt des generierten Entscheidungsbaums als PHP-Klasse

baum in Textform verändert, merkt Symfony sich das Änderungsdatum und entscheidet basierend darauf, ob die PHP-Datei neu generiert werden muss.

5.5.2 Verwendung der Entscheidungsbäume

Die jeweiligen Entscheidungsbäume werden basierend auf der Analysemethode im TravelMode-Filter verwendet. Dabei wird für jedes Segment der Entscheidungsbaum traversiert und das Resultat bei dem jeweiligen Segment hinterlegt. Schlussendlich hat jedes Segment einen Verkehrsmitteltyp zugeordnet bekommen und wird zur Nachbearbeitung weitergegeben.

5.5.3 Nachbearbeitung

Die Nachbearbeitung der Segmente mit dem bestimmten Verkehrsmittel wurde sowohl von Zheng [Zheng et al., 2010] als auch Biljecki [Biljecki et al., 2013] vorgeschlagen bzw. beschrieben. Dabei geht es darum die klassifizierten Segment auch in einem Kontext zu sehen. So kann laut Zheng [Zheng et al., 2010] zum Beispiel nicht ein “Auto“-Segment auf ein “Bus“-Segment folgen, ohne dass sich zwischen ihnen ein Segment zu Fuß befindet (auch wenn dieses sehr kurz ist). Durch diese Aussage können sich Verkehrsmittelwechsel die zum Beispiel aufgrund der Testdaten entstanden aber nicht plausibel sind, beseitigen lassen. Dies bedeutet, dass das vorherige korrekt identifizierte Segment (im Sinne des Kontexts) als Ausgang für das nächste Segment verwendet wird und dabei auch der Verkehrsmitteltyp des vorherigen Segments übernommen wird.

Ein Beispiel dafür ist in Abbildung 5.3 zu sehen. Links sieht man dabei den analysierten Track ohne Nachbearbeitung und rechts mit Nachbearbeitung. Die Abfolge der Verkehrsmittel links ist dabei Bus->Auto->Fahrrad->Auto->Fahrrad->Bus->Auto was schlicht nicht möglich ist und aufgrund der Trainingsdaten und den fehlenden Kontext entstanden ist. Nach der Nachbearbeitung ist das korrekte Resultat eine durchgängige Busfahrt.

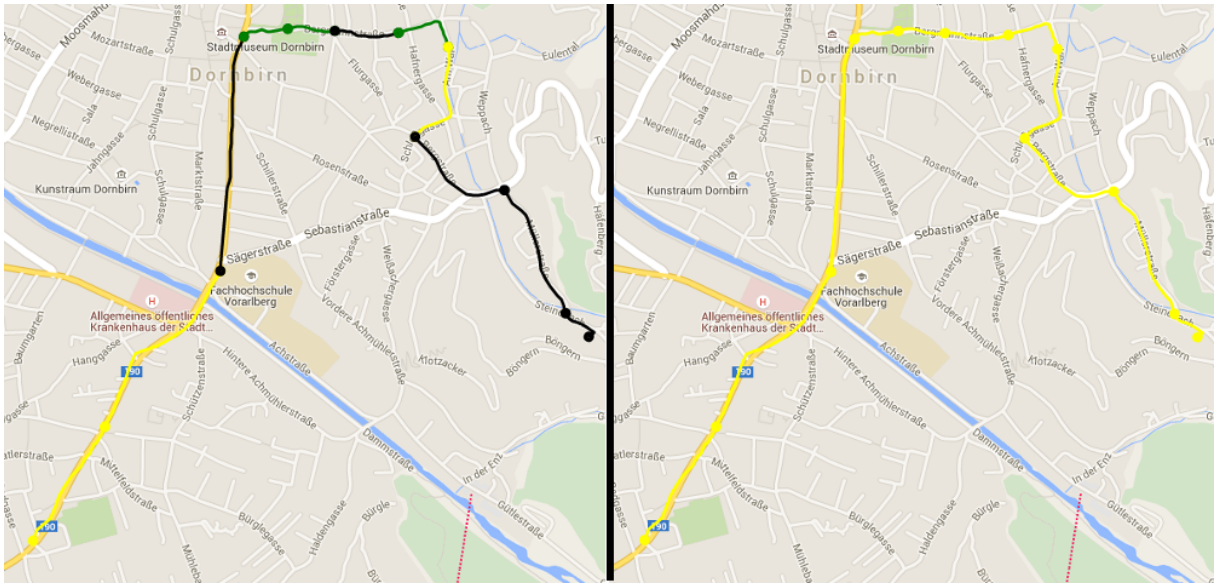


Abbildung 5.3: Kein Nachbearbeiten vs. Nachbearbeiten

Eine weitere Aufgabe die das Nachbearbeiten übernimmt und aufgrund der vielen unterschiedlichen Fahrrad-Trainingsdaten entstanden ist, ist jene vereinzelte Fahrradsegment herauszufilter. In einigen Fällen konnte es vorkommen, dass einzelne Segmente als Fahrrad-Segment identifiziert worden sind, diese aber für sich alleine standen wie z.B. zu Fuß->Fahrrad->Bus->Bus->Bus. Aus diesem Grund wurde das Nachbearbeiten dahingehend erweitert, dass einzelne Fahrradsegmente herausgefiltert werden bzw. der Typ des Segments mit dem des vorherigen oder nächsten Segments abgestimmt wird.

5.6 Zusammenfassung

zusammen
processing

Auswertung

zu wenig
nings) dat
und zu we
personen
wäre unte
schiedlich
sonen für
unterschli
daten um
zu spezifis
der generi
der entsch
dungsbäu
werden

mtb track
gen unter
ständen m
irrtümer i
dell

6.1 Genauigkeit ohne GIS-Daten

6.2 Genauigkeit mit GIS-Daten

Ausblick

- GIS-Daten für Ortsgebiet - Entscheidungsbaum wirklich das Richtige? - Qualität und Quanität der Test/Trainingsdaten - Trainingsdaten MTB Berg (langsam) vs Stadt/Eben (schnell) - Absatz über Automatisierung von RapidMiner –> DecisionTree - Genauigkeit der Entscheidungsbäume integrieren (Performance-Tab in Rapidminer) –> Abhängig von Trainingsdaten zwischen 62% und 72% Genauigkeit

Literaturverzeichnis

[Biljecki et al., 2013] Biljecki, F., Ledoux, H., and van Oosterom, P. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2).

[Buschmann, 1998] Buschmann, F. (1998). Pipes and Filters. In *Pattern-orientierte Software-Architektur: ein Pattern-System*, Professionelle Softwareentwicklung, pages 54–71. Addison-Wesley.

[Caron et al., 2006] Caron, F., Duflos, E., Pomorski, D., and Vanheeghe, P. (2006). GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. *Information Fusion*, 7(2):221–230.

[Douglas C. Giancoli, 2010] Douglas C. Giancoli (2010). *Physik: Lehr-und Übungsbuch*. Pearson Deutschland GmbH, 3 edition.

[Gonzalez et al., 2010] Gonzalez, P. A., Weinstein, J. S., Barbeau, S. J., Labrador, M. A., Winters, P. L., Georggi, N. L., and Perez, R. (2010). Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. *IET intelligent transport systems*, 4(1):37–49.

[Howard Hamilton, 2009] Howard Hamilton (2009). Machine Learning/Inductive Inference/Decision Trees/Overview. `http://www2.cs.uregina.ca/~dbd/cs831/index.html`.

[Jeffrey P. Bradford et al., 1998] Jeffrey P. Bradford, Clayton Kunz, Ron Kohavi, Cliff Brunk, and Carla E. Brodley (1998). Pruning Decision Trees with Misclassification Costs. Technical Report 51, Purdue University.

[Jun et al., 2006] Jun, J., Guensler, R., and Ogle, J. H. (2006). Smoothing methods to minimize impact of Global Positioning System random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record: Journal of the Transportation Research Board*, 1972(1):141–150.

[Microsoft Research, 2015] Microsoft Research (2015). GeoLife GPS Trajectories - Microsoft Research. `http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/`.

[Movable Type Ltd, 2015] Movable Type Ltd (2015). Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript. `http://www.movable-type.co.uk/scripts/latlong.html`.

[Nadine Schüssler et al., 2011] Nadine Schüssler, Lara Montini, and Christoph Dobler (2011). Improving post-processing routines for gps oversavations using propted-recall data. In *9th International conference on survey methods in transport*.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Raileanu and Stoffel, 2004] Raileanu, L. E. and Stoffel, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93.

[RapidMiner, 2015] RapidMiner (2015). RapidMiner Studio - RapidMiner Documentation. `http://docs.rapidminer.com/studio/`.

[Reddy et al., 2008] Reddy, S., Burke, J., Estrin, D., Hansen, M., and Srivastava, M.

(2008). Determining transportation mode on mobile phones. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pages 25–28. IEEE.

[Reddy et al., 2010] Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., and Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2).

[Schuessler and Axhausen, 2009] Schuessler, N. and Axhausen, K. W. (2009). Processing raw data from global positioning systems without additional information. *Transportation Research Record: Journal of the Transportation Research Board*, 2105(1):28–36.

[Sebastian Nagel, 2011] Sebastian Nagel (2011). Möglichkeitsstudie zum Projekt: Mobilitäts-Tracker.

[Stenneth et al., 2011] Stenneth, L., Wolfson, O., Yu, P. S., and Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 54–63. ACM.

[Thomas Mitchell, 1997] Thomas Mitchell (1997). Which Attribute Is the Best Classifier? In *Machine Learning*, pages 55–59. Mcgraw-Hill Publ.Comp., international edition edition.

[Tom Dietterich, 1995] Tom Dietterich (1995). Overfitting and Undercomputing in Machine Learning. 27(3):326–327.

[Topografix, 2004] Topografix (2004). GPX 1.1 Schema Documentation. `http://www.topografix.com/GPX/1/1/`.

[Wei-Yin Loh, 2008] Wei-Yin Loh (2008). Classification and Regression Tree Methods. In *Encyclopedia of Statistics in Quality and Reliability*, pages 315–323. Department of Statistics, University of Wisconsin.

[Youtube, 2015] Youtube (2015). Youtube statistics. `http://www.youtube.com/yt/press/statistics.html`.

[Zheng et al., 2010] Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y. (2010). Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web (TWEB)*, 4(1).

[Zheng et al., 2008a] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y. (2008a). Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM.

[Zheng et al., 2008b] Zheng, Y., Liu, L., Wang, L., and Xie, X. (2008b). Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th international conference on World Wide Web*, pages 247–256. ACM.

Listings

Tabellenverzeichnis

Abbildungsverzeichnis

Anhang 1

Wie auch bei vielen anderen Publikation die sich mit GPS-Daten beschäftigen, konnte auch bei dieser Arbeit festgestellt werden, dass sich in den GPS-Spuren einige Ausreißer befanden. Dies konnte vor allem dann beobachtet werden, wenn man sich in einem Zug befand, durch einen Tunnel fuhr oder auch wenn man sich auf einem überdachten Bahnsteig befand. Die Ausreißer werden durch unrealistisch große Distanzabstände oder Sprünge in den Höhenwerten bemerkt. Da diese Werte das Ergebnis verfälschen würden, mussten verschiedene Filter implementiert werden.

In verschiedensten Publikation zum Thema GPS wird beim Filtern von fehlerhaften Außreisern auf den Kalman-Filter gesetzt (z.B. [Caron et al., 2006] und [Jun et al., 2006]). Da das Filtern an sich aber nicht im Fokus dieser Arbeit stand und es zur Zeit dieser Arbeit keine Implementation des Kalmanfilters für die gewählte Programmiersprache PHP gab wurde auf einfachere Filtermöglichkeiten, wie sie in [Schuessler and Axhausen, 2009] grob beschrieben werden, gesetzt.

Konkret wurden Filter für Zeit, Distanz und Höhenmeter implementiert. Die somit bereinigten Resultate hatten bereits auf den Entscheidungsbaum ohne GIS-Daten große Auswirkungen. Die Grenzwerte für diese Filter können in einer Konfigurationsdatei festgelegt und je nach geografischer Region und Testdaten angepasst werden. Diese Filter können auch als konfigurierbares Regelwerk verstanden und wie folgt beeinflusst werden:

- Minimale Zeitspanne zwischen 2 Punkten
- Maximale Distanz pro Sekunde
- Minimale Distanz pro Sekunde
- Maximale Änderung der Höhenmeter zwischen 2 Punkten pro Sekunde
- Minimaler Wert für Verhältnis zwischen gültigen und aufgezeichneten Punkten
- Anzahl der zu überspringenden Punkte am Start
- Minimale Anzahl von Punkten pro Segment

GPS-Punkte, welche nicht den definierten Grenzwerten entsprechen werden im der weiteren Verarbeitung nicht betrachtet. Abgesehen vom Zeitfilter betrachten alle anderen Filter die gemessenen Werte in Relation zur gemessenen Zeit, was bedeutet, dass der Zeitwert größer als 0 sein muss und von mehren GPS-Punkten mit der selben Zeit nur einer betrachtet wird. In den nächsten Absätzen findet sich eine genauere Beschreibung der einzelnen Parameter und ihrer Auswirkungen.

Minimale Zeitspanne zwischen 2 Punkten Mit Hilfe dieses Konfigurationsparameters kann zum einen sichergestellt werden, dass sich nicht mehrere Trackpunkte mit dem selben Zeitstempel eingeschlichen haben und zum anderen kann man damit auch die Genauigkeit bzw. die Anzahl der zu verarbeitenden Punkte steuern.

Maximale/minimale Distanz pro Sekunde Durch diesen Parameter kann sichergestellt werden, dass man sich innerhalb einer gewissen Zeitspanne nur eine gewisse Strecke zurücklegen kann beziehungsweise auch Punkte filtern in denen man sich nicht wirklich bewegt hat. Dadurch können die richtig großen Sprünge gefiltert werden und Punkte ohne Bewegung herausgefiltert werden..

Maximale Änderung der Höhenmeter zwischen 2 Punkten pro Sekunde Hierbei wird überprüft ob es große Sprünge im Bereich der Höhenmeter gibt.

Minimaler Wert für Verhältnis zwischen gültigen und aufgezeichneten Punkten Durch dieses Verhältnis kann entschieden werden ob sich überhaupt noch genug gültige GPS-Informationen für den weiteren Bestimmungsprozess in einem Track.

Anzahl der zu überspringenden Punkte am Start Durch diesen Konfigurationsparameter kann gesteuert werden wie viele Punkte am Beginn einer Aufzeichnung übersprungen werden. Dies rührt daher, dass sich beim Start einer Aufzeichnung gerne Ausreiser einschleichen bis sich der Positionsbestimmungsvorgang eingependelt hat. Durch diesen Parameter können einige dieser Trackpoints übersprungen werden.

Minimale Anzahl von Punkten pro Segment Über diesen Parameter kann gesteuert werden, wie viele Punkte sich in einem Tracksegment befinden, damit es überhaupt weiterverarbeitet wird. Der minimale Wert für diesen Parameter ist 2.

Verschiedene Fälle beim Filtern

Es gibt drei verschiedene Fälle, welche beim Filtern von Ausreißern abgedeckt werden sollten. Der grundlegende Algorithmus, welcher vom aktuellen Punkt ausgehend einen neuen gültigen Punkt sucht und alle ungültigen überspringt, funktioniert in den ersten zwei Fällen. Im dritten Fall muss noch eine zusätzliche Überprüfung stattfinden.

1. Fall

Beim ersten Fall befinden sich ein oder mehrere Ausreißer am Ende der GPS-Spur wie es in Abbildung 7.1 bei dem letzten Punkt der Fall ist. Dies bedeutet, dass ab einem gewissen Punkt keine weiteren validen Punkte gefunden und alle folgenden Punkte übersprungen werden.

2. Fall

Beim zweiten Fall befinden sich ein oder mehrere Ausreißer zwischen validen voran-

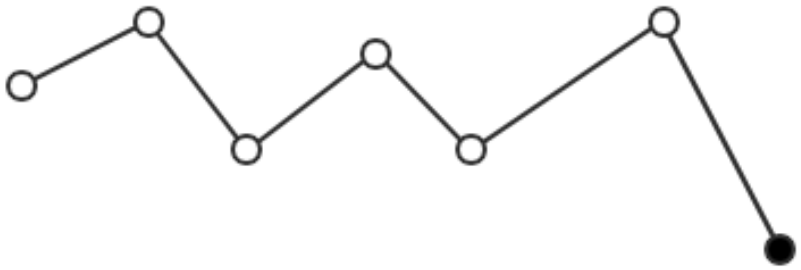


Abbildung 7.1: Filtern - 1. Fall

gegangenen und nachfolgenden Punkten. Ein Beispiel ist in Abbildung 7.2 mit dem vierten Punkt als Ausreißer abgebildet. Dies bedeutet, dass ein oder mehrere Punkte übersprungen werden und danach mit den gültigen Punkten weitergearbeitet werden kann.

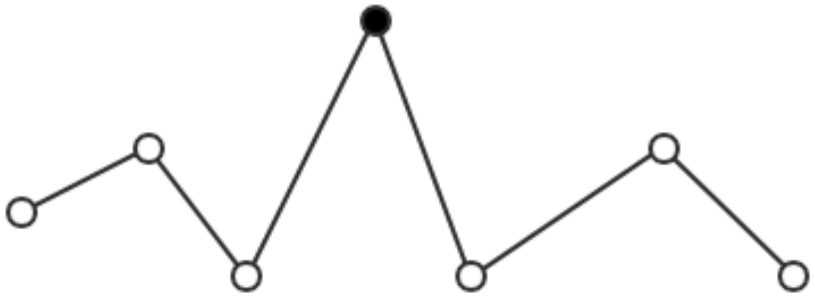


Abbildung 7.2: Filtern - 2. Fall

3. Fall

Im dritten Fall befinden sich ein oder mehrere Ausreißer am Beginn der GPS-Spur. Damit ist gemeint, dass vom Start weg keine gültigen Punkte vorhanden sind und erst im Laufe der Aufzeichnung gültige Punkte aufgezeichnet werden. Dies kann vorkommen, wenn die Aufzeichnung der GPS-Spur sofort nach dem Aktivieren des GPS-Moduls startet. Die Position konnte noch nicht mit ausreichender Genauigkeit bestimmt werden

und es wird mit einer niederen Genauigkeit gestartet. Im Laufe der Aufzeichnung steigt die Genauigkeit und es kann zu einem Sprung von ungenauen zu den genauen Punkten kommen. Ein Beispiel hierfür ist in Abbildung 7.3 mit den ersten 3 Punkten als Ausreißern ersichtlich.

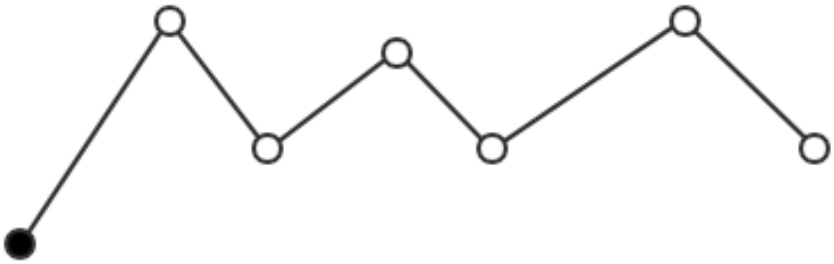


Abbildung 7.3: Filtern - 3. Fall

Da keinerlei Information über die Genauigkeit der aufgezeichneten GPS-Punkte gesammelt wurde, ist es eine komplexe Aufgabe einen gültigen Startpunkt zu finden ohne unnötig viele GPS-Daten zu überspringen. In einzelnen Testfällen kam es vor, dass ein großer Teil der Strecke einfach weggelassen wurde, weil kein gültiger Startpunkt bzw. nicht genügend valide und aufeinander folgende Punkte gefunden werden konnten. Deshalb wurde eine konstanter Parameter für die Anzahl der zu überspringenden GPS-Punkte am Anfang eines Tracks festgelegt.

Zeitfilter

Der Zeitfilter überprüf,t ob der Abstand zwischen zwei GPS-Punkten größer gleich einem minimalen Wert (in diesem Fall 0) ist. Dadurch wird verhindert, dass zwei Punkte mit demselben Zeitstempel verarbeitet werden und bei den zeitabhängigen Berechnungen durch 0 dividiert wird. Außerdem kann man dadurch auch steuern, wie viele Punkte pro GPS-Spur überprüft werden (z.B. nur jeder 2. Punkt), beziehungsweise welche

Punkte ausgelassen werden sollen um den Prozess zu beschleunigen oder weil sich der Grad an Genauigkeit nicht wesentlich verbessert.

Distanzfilter

Der Distanzfilter kontrolliert, ob sich der Abstand zwischen zwei Punkten im Verhältnis zur Zeit in einem gewissen Bereich befindet. In dieser Arbeit wurde größer 0 m pro Zeiteinheit als minimale und kleiner 50 m pro Zeiteinheit als maximale Distanz festgelegt. Liegt ein Punkt nicht innerhalb dieser Grenzen so wird der aktuelle Punkt mit dem Punkt nach dem Ausreißer verglichen. Dies wird solange gemacht bis wieder ein Punkt mit valider Distanz gefunden wird oder keine GPS-Punkte mehr vorhanden sind.

Höhenfilter

Der Höhenfilter filtert, ähnlich wie der Geschwindigkeitsfilter, jene GPS-Punkte, bei welchen die Differenz der Höhenwerte zu groß ist. Im Fall der hier verwendeten Trainingsdaten wurde 25 m/s für diesen Filter festgelegt und alle Punkte mit einen größeren Differenz werden herausgefiltert.

Resultat der Filter

Ein postitives Resultat der angewandten Filter auf die GPS Daten sind in Abbildung 7.4 und Abbildung 7.5 zu sehen. Es konnte jedoch festgestellt werden, dass zum Beispiel eine zu hohe Mindestdistanz bei Testdaten einer Mountainbike-Tour zu einem Verlust von sehr vielen Trackpunkten führen würde. Darum sollte die Mindestdistanz eher kleiner gewählt werden um auch bei solchen Tracks gute Resultate und eine gewisse Detailtreu zu erhalten.

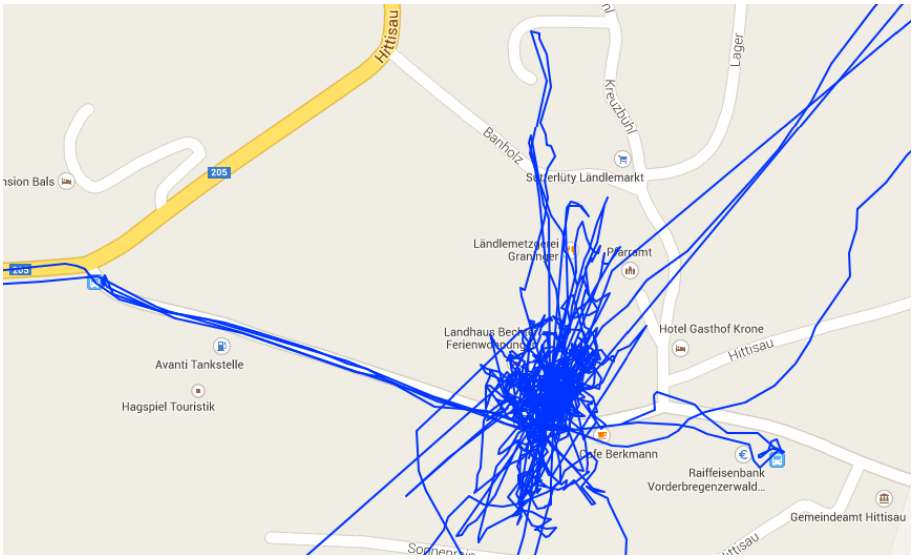


Abbildung 7.4: GPS-Track ohne Filter

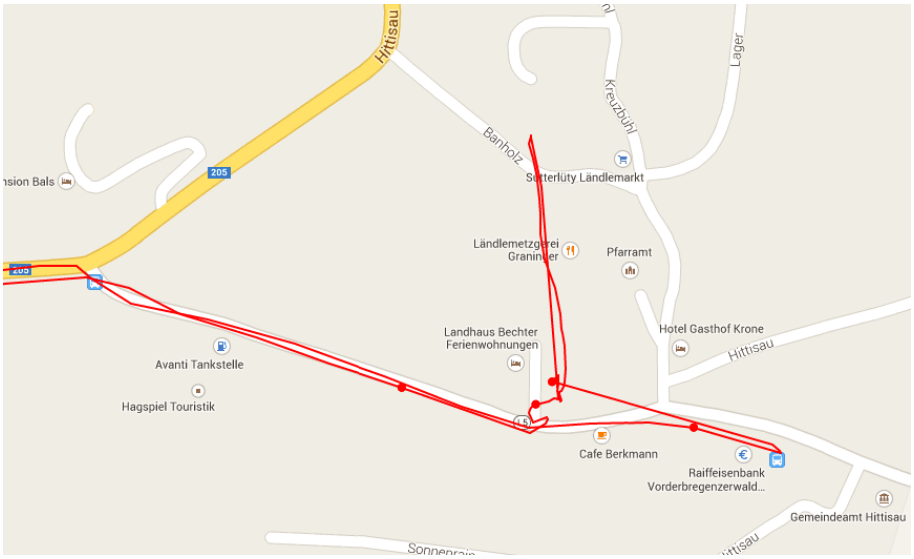


Abbildung 7.5: GSP-Track mit Filter

Anhang 2

Im folgenden Abschnitt werden zwei Vorgänge beschrieben die für das Hinzufügen von weiteren Analysemethoden (neben der Basis- und GIS-Variante) oder das Erstellen von Entscheidungsbäumen anhand neuer Daten essentiell sind. Der erste Abschnitt beschreibt dabei, welche Schritte auszuführen sind um mit Hilfe neuer/geänderter Rohdaten neue Trainingsdaten zu erstellen. Der zweite Abschnitt geht dann auf die Erstellung eines Entscheidungsbaums mit RapidMiner und auf Basis der Trainingsdaten, ein.

Ermittlung der Trainingsdaten

Grundsätzlich unterscheiden sich Trainingsdaten im Sinne von GPX-Dateien nicht wesentlich von den Rohdaten die man von dem GPX-Gerät bekommt. Da die anschließende Erstellung eines Entscheidungsbaums auf Basis dieser Daten geschieht, wird eine zusätzliche Information über das tatsächlich verwendete Verkehrsmittel für das jeweilige Segment benötigt. Dies bedeutet, dass jede GPX-Trainingsdatei den selben Aufbau wie eine normale Datei hat, aber bei jedem Tracksegment (trkseg) ein Verkehrsmitteltyp mitangegeben werden muss (siehe Listing 7.1).

Dies bedeutet, dass bei all jenen Dateien die als Trainingsdaten verwendet werden sollen, die Tracksegmente korrekt gesetzt werden müssen und das Verkehrsmittel im Attribut “type“ hinzugefügt werden muss.

```
1 ...
2     <trk>
3         ...
4         <trkseg type=‘‘bike‘‘>
5             <trkpt lat=‘‘47.39786‘‘ lon=‘‘9.735109‘‘>
6                 <ele>475.0</ele>
7                 <time>2015-02-19T07:20:18.156Z</time>
8             </trkpt>
9             ...
10        </trkseg>
11        ...
12    </trk>
13 ...
```

Listing 7.1: GPX-Datei für Trainingsdaten

```
1 app/console tmd:generate:trainingdata
```

Listing 7.2: Kommando für Erstellen der Trainingsdaten

Auf Basis dieser Trainingsdaten kann dann mit folgendem Kommando (siehe Listing 7.2) eine CSV-Datei generiert werden. Dieses Kommando muss im Wurzelverzeichnis des Projekts ausgeführt werden und akzeptiert drei Argumente in folgender Reihenfolge:

- Der **Verzeichnisname** bzw. Pfad zu dem Verzeichnis mit den Trainingsdaten
- Der **Dateiname** für die generierte Datei
- Die **Analysemethode** die zum Generieren verwendet werden soll

Generierung des Entscheidungsbaums

Die im vorherigen Abschnitt generierte CSV-Datei mit den Trainingsdaten kann nun in RapidMiner importiert werden. Aus diesen kann dann schlussendlich mit folgenden Schritten ein Entscheidungsbaum generiert werden:

1. Nachdem das Programm geöffnet worden ist kann über das Hauptmenü File -> Import Data -> Import CSV File die generierte Trainingsdaten-Datei als Datenresource importiert werden.

2. Dafür wählt man nun die entsprechende Datei im Dialog aus und folgt anschließend dem Dialog bis Schritt 4 (bei den Schritten 2 und 3 mussten für den Prototypen keine Änderungen vorgenommen werden).

3. In Schritt 4 ist es wichtig für die Resultat-Spalte (jene Spalte in, welcher die tatsächlichen Verkehrsmittel stehen) als Datentyp “text“ und als allgemeinen Typ “label“ auswählen.

4. Im nächsten Schritt kann man diese importierten Dateien zur Verwendung in RapidMiner unter dem gewünschten Namen ablegen.

5. Für die Generierung des Entscheidungsbaums wird als erstes ein neuer Prozess benötigt. Diesen kann man bequem über das Hauptmenü anlegen (File -> New Process).

6. Als nächstes kann man die zuvor importieren Daten aus dem Repository-Bereich (links unten) zu dem neuen Prozess per Drag-and-Drop hinzufügen.

7. Im nächsten Schritt zieht man den “Decision-Tree“-Operator aus dem Operatoren-Bereich (links oben) in den Prozess. Diesen Verknüpft man nun mit den Daten (out/tra) sowie mit dem Ende des Prozesses (mod/res).

Jetzt kann man mit Hilfe des “Run“-Buttons (blaues Dreick) unterhab des Hauptmenüs, den Prozess ausführen und bekommt den Entscheidungsbaum sowohl in grafischer als auch textueller Darstellung zu sehen. Eine etwas ausführlichere Anleitung ist auch auf Youtube unter `https://www.youtube.com/watch?v=Vf6G1HNdBoI` zu finden.

Der Entscheidungsbaum in Textform kann nunmehr in die in der Konfiguration des Projekts festgelegten Datei abgelegt werden. Beim nächsten Analyseprozess wird erkannt, dass es sich um eine geänderte/neue Datei handelt und automatisch eine neue PHP-Datei mit dem Baum als PHP-Code dafür generiert.