

Situationsanalyse im öffentlichen Raum

Anja Bergdolt
Hannes Benne

Humboldt-Universität zu Berlin
Institut für Informatik

22. Juli 2019



Gliederung

- 1** Projektziele
- 2 Klassifikation und Situationsanalyse mit Ähnlichkeitsmaßen
- 3 Regressionsmodelle zur Vorhersage gefährlicher Situationen



Projektziele

- Automatisches Erkennen von zuvor definierten Situationen (Menschen die bestimmte Gebäude oder öffentliche Verkehrsmittel betreten)
- Erkennen von anormalen Verhaltensweise
- Erkennen von gefährlichen Situationen



Verwendete Daten

Trajektorien aus folgenden Quellen:

- Leave a Trace Datensatz
- Verkehrsdaten aus Cvat
- Verkehrsdaten von der Trackinggruppe



Gliederung

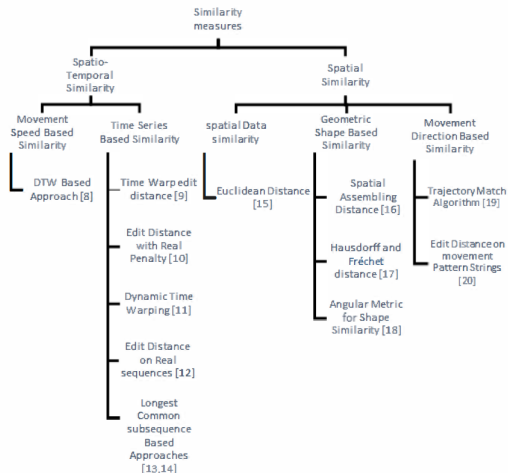
- 1 Projektziele
- 2 Klassifikation und Situationsanalyse mit Ähnlichkeitsmaßen**
- 3 Regressionsmodelle zur Vorhersage gefährlicher Situationen



Situationsanalyse mit Ähnlichkeitsmaßen

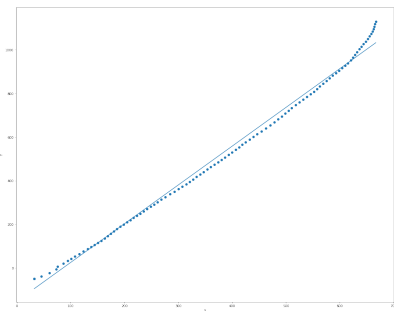
- Situationen definieren, die erkannt werden sollen
- Für diese Situationen typische Trajektorien finden
- beobachtete Trajektorien über Ähnlichkeitsmaße den Situationen zuordnen

Ähnlichkeitsmaße

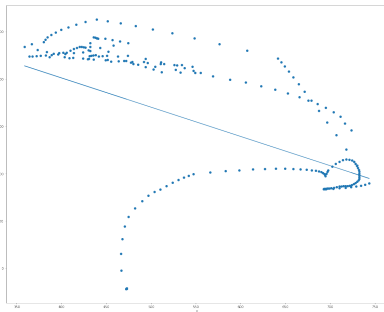


Mittlere quadratische Abweichung

Idee: Normale Trajektorien haben geringere Abweichung von Regressionsgeraden.



(a) MSE = 820.9



(b) MSE = 19987.6

Abbildung: Beispiel für normale und anormale Trajektorie



Definition: Fréchetdistanz (Dog walking distance)

Sei (V, d) ein metrischer Raum und
 $f : [a, b] \rightarrow V, g : [a', b'] \rightarrow V$ Kurven. Dann ist die Fréchet
Distanz definiert als:

$$d_F(f, g) = \inf_{\alpha, \beta} \max_{t \in [0, 1]} d(f(\alpha(t)), g(\beta(t)))$$

Wobei $\alpha : [a, b] \rightarrow [0, 1]$ und $\beta : [a', b'] \rightarrow [0, 1]$ beliebige
monoton wachsende, stetige Funktionen sind.



Diskrete Fréchetdistanz

In der Praxis:

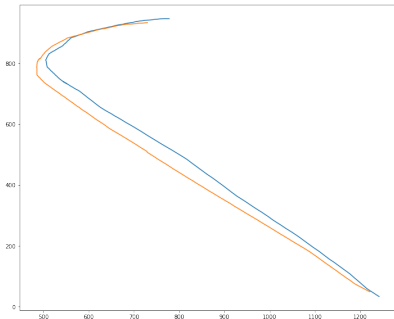
- Stetige Kurven durch Polygonzüge approximieren
- Unsere Trajektorien liegen schon in diskreter Form vor

Implementierungsdetails für diskrete Fréchetdistanz in:
Eiter, Thomas/ Mannila, Heikki: Computing Discrete Fréchet
Distance

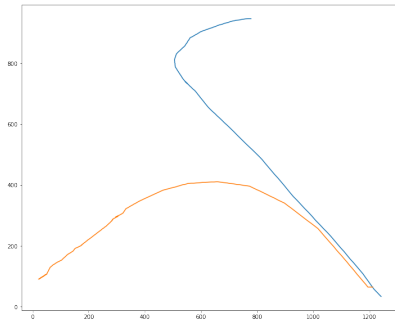
Python Implementation:

<https://pypi.org/project/similaritymeasures/>

Diskrete Fréchetdistanz



(a) $d_F = 49.73$



(b) $d_F = 1142.42$

Abbildung: Trajektorien in Bildkoordinaten und ihre Fréchetdistanz



Erkennen von Situationen

- Aus Trainingsdaten Klassen von Situationen definieren.
- Repräsentanten für diese Klassen bestimmen.
- Testdaten klassifizieren, indem wir die mit den Repräsentanten abgleichen.

Diskrete Fréchetdistanz

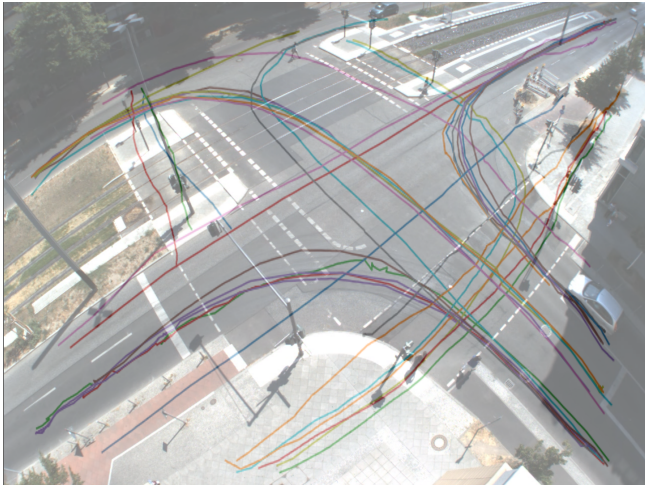


Abbildung: Kreuzung mit Trainingsdaten

Erzeugen von repräsentativen Trajektorien für Situationen

Annotierte Daten (Frame 11000 bis 12000)

Auto von S Adlershof biegt auf Groß-Berliner Damm ab

$A1_{ids} = [2,3,4,24,25]$

Auto biegt von Groß-Berliner Damm Richtung S Adlershof ab

$A2_{ids} = [5,14,18,29,31,36]$

Auto biegt von Groß-Berliner Damm Richtung Johann von Neumann-Haus ab

$A4_{ids} = [0,1,6,15,17,30,34,35]$

Fußgänger läuft von Commerzbank zu Subway

$F1_{ids} = [10,32,33]$

Fußgänger läuft von S Adlershof nach Johann von Neumann-Haus

$F2_{ids} = [11,12,13,21]$

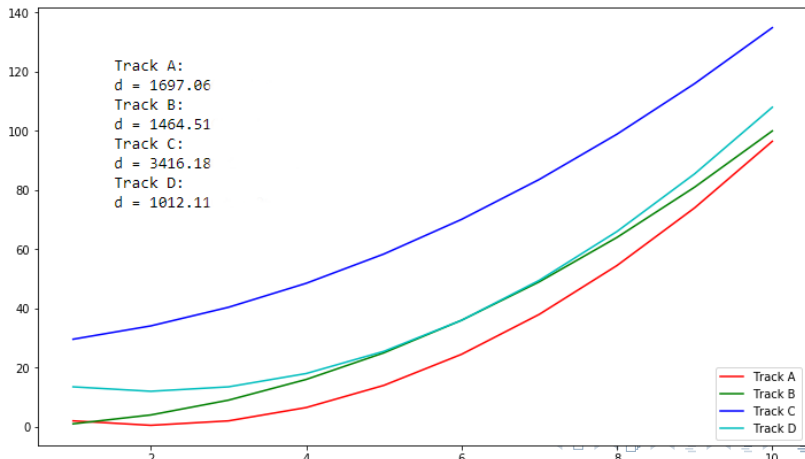


Erzeugen von repräsentativen Trajektorien für Situationen

Sei C eine Klasse von Trajektorien. Ein Repräsentant für diese Klasse ist eine Trajektorie x , deren Quadratsumme der Abstände zu allen anderen Trajektorien der Klasse minimal ist.

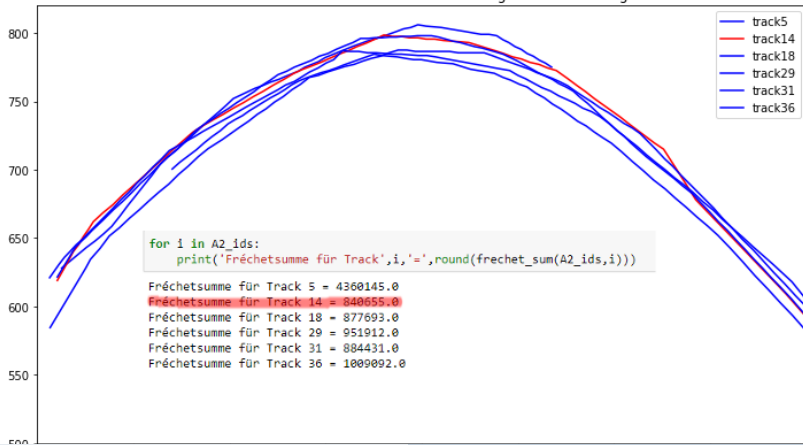
$$x \text{ ist Repräsentant für } C \text{ gdw. } \sum_{y \in C} d_F(x, y)^2 = \min_{z \in C} \sum_{y \in C} d_F(z, y)^2$$

Erzeugen von repräsentativen Trajektorien für Situationen



Erzeugen von repräsentativen Trajektorien für Situationen

Autos die vom Großberliner Damm Richtung Adlershof abbiegen





Klassifikation von Trajektorien

Wir ordnen eine beobachtete Trajektorien in diejenige Klasse ein, zu deren Repräsentanten sie die geringste Fréchetdistanz hat.



Klassifikation von atypischen Situation

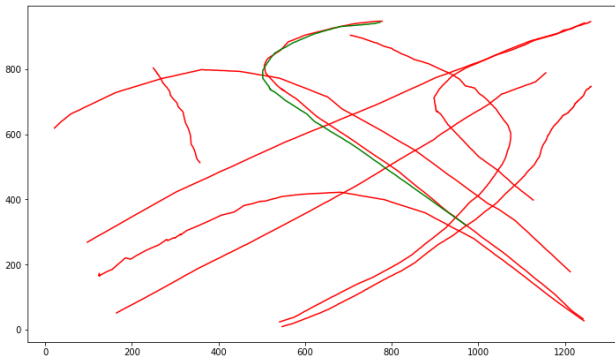
Eine Situation ist atypisch, wenn ihre Trajektorie zu allen Klassen einen großen Abstand aufweist -> Parameter für den Klassifikator

Klassifikation Beispiel

```
testdata = pd.read_csv('12000-13000.csv', header=0, delimiter= ' ')  
report(testdata, 16, 400)
```

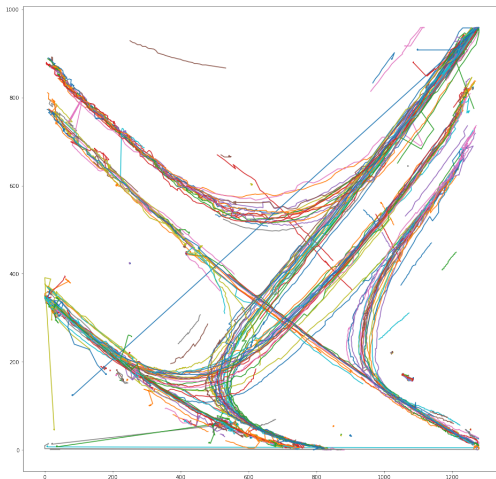
Der beobachtete Track gehört zur Klasse A3:
Auto von Johann von Neumann-Haus biegt auf Groß-Berliner Damm ab

Der Abstand zum Repräsentanten der Klasse beträgt: 394.74865420923226



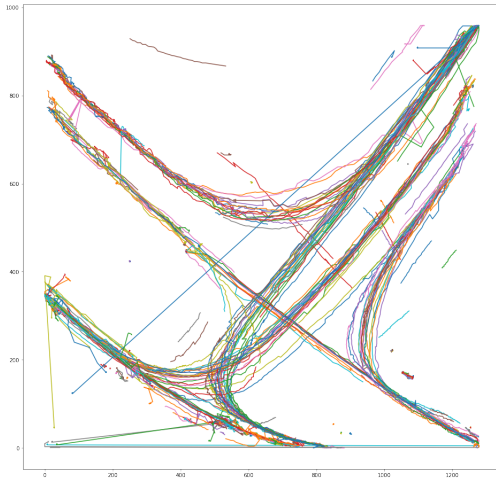
Klassifikation mit echten Daten

Frames 1 bis 6425, 893 Trajektorien



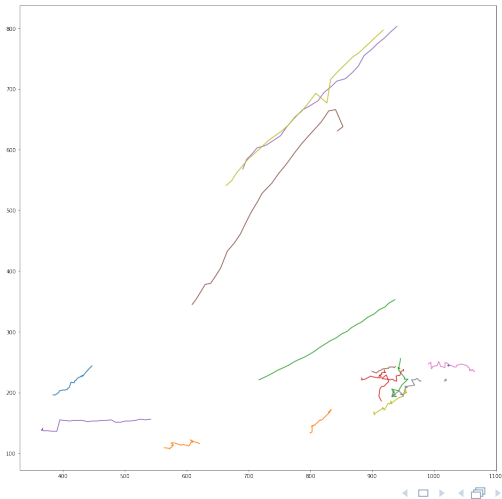
Klassifikation mit echten Daten

Nur Autos



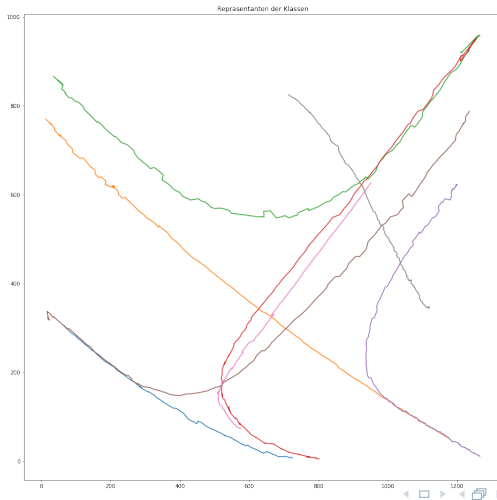
Klassifikation mit echten Daten

Nur Radfahrer



Klassifikation mit echten Daten

Repräsentanten der Klassen





Ergebnisse/ Metriken

ID	classes	classified
20	x	x
21	x	f2
22	x	x
23	f2	f2
24	a5	a5

Ergebnisse/ Metriken

Konfusionsmatrix:

[3	0	0	0	0	0	0	0	0]
[0	1	0	0	0	0	0	0	0]
[0	0	10	0	0	0	0	0	0]
[0	0	0	8	0	0	0	0	0]
[0	0	0	0	11	0	0	0	1]
[0	0	0	0	0	15	0	0	0]
[0	0	0	0	0	0	5	0	4]
[0	0	0	0	0	0	0	7	1]
[4	0	0	1	4	0	0	11	767]

(Schwellwert für atypische Tracks = 300)



Ergebnisse/ Metriken

Precision und recall für die einzelnen Klassen: Sei M die Konfusionsmatrix. Dann berechnen sich precision und recall für die i -te Klasse wie folgt:

$$\text{Precision}_i = \frac{M_{i,i}}{\sum_j M_{j,i}}$$

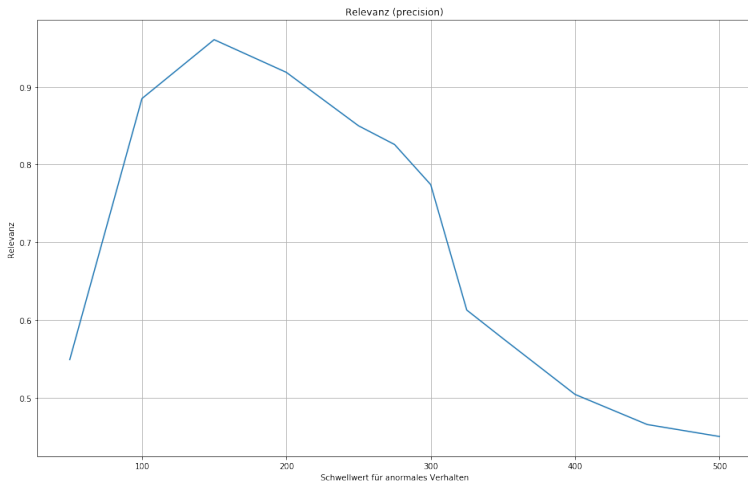
$$\text{Recall}_i = \frac{M_{i,i}}{\sum_j M_{i,j}}$$



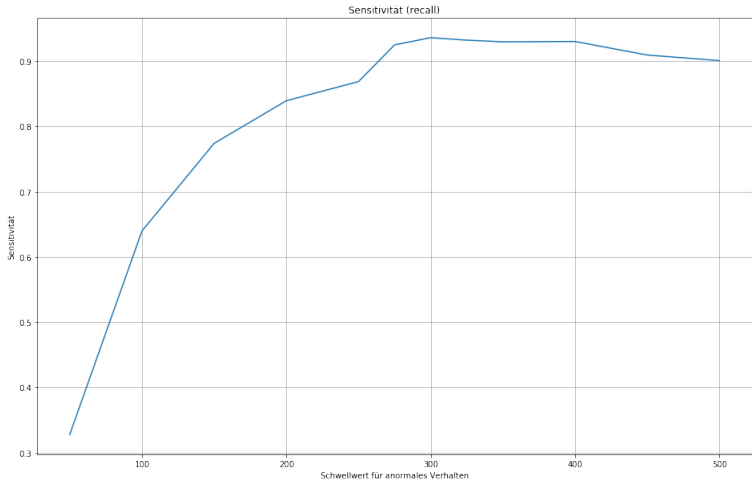
Ergebnisse/ Metriken

	Maß	a1	a2	a3	a3	a4	a5	a6	f2	b1	Mittelwert
0	recall	1.00	1.0	1.0	1.0	0.916667	1.0	0.666667	0.875000	0.963151	0.935721
1	precision	0.25	1.0	1.0	0.8	0.733333	1.0	0.857143	0.333333	0.994751	0.774284

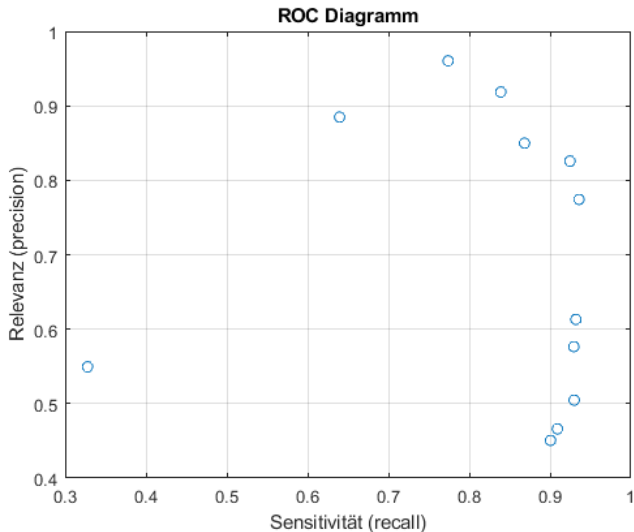
Ergebnisse/ Metriken



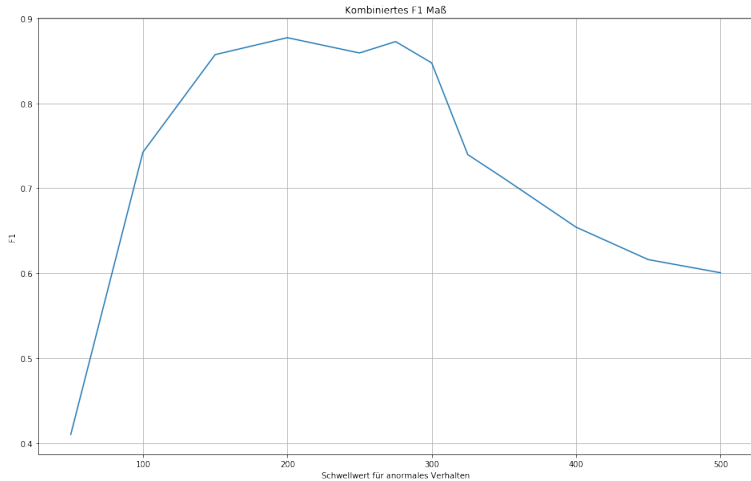
Ergebnisse/ Metriken



Ergebnisse/ Metriken



Ergebnisse/ Metriken





Ergebnisse/ Metriken

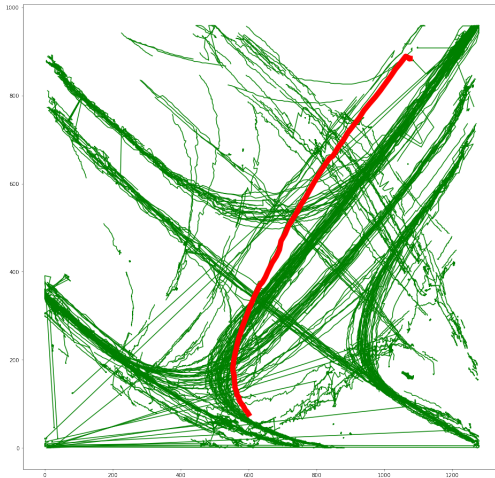
Bestes Ergebnis: Schwellwert = 200

$$\text{recall} = 84\%$$

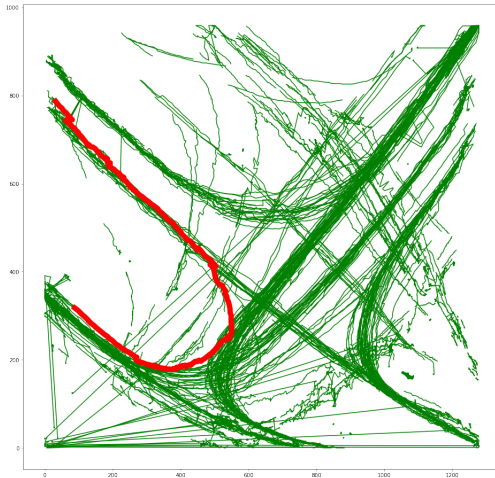
$$\text{precision} = 92\%$$

$$F = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = 88\%$$

Beispiel für atypisches Verhalten



Beispiel für atypisches Verhalten





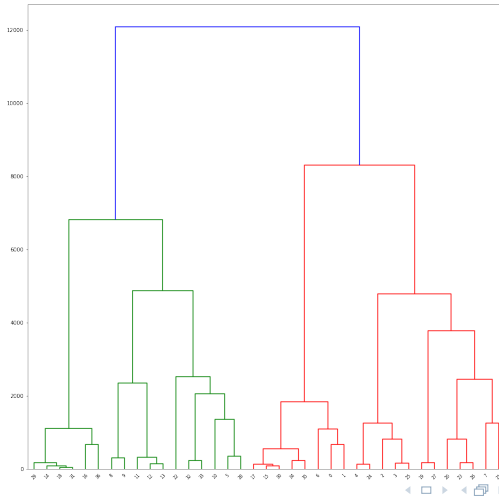
Anpassung des Systems an neue Umgebungen

Problem: Wenn wir eine neue Kreuzung oder andere Umgebung überwachen wollen, müssen wir den Klassifikator neu trainieren. Hunderte Tracks per Hand zu befunden ist anstrengend.

Lösung: Unüberwachtes Lernen (Hierarchische Clusteranalyse)

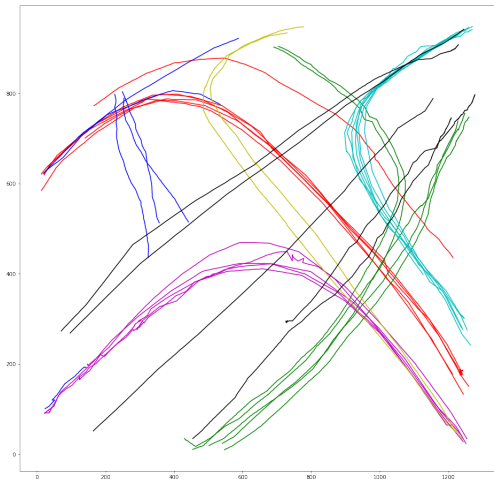
Anpassung des Systems an neue Umgebungen

Dendrogramm mit Ward linkage



Anpassung des Systems an neue Umgebungen

$k = 7$





Zeitkomplexität

Gegeben zwei Trajektorien mit n Punkten.

- Erster Algorithmus 1991 von Alt und Godau.

$$\text{Fréchet} \in \mathcal{O}(\log(n)n^2)$$

- Moderne Implementationen ebenfalls in $\mathcal{O}(\log(n)n^2)$.



Zeitkomplexität

Aus der starken Exponentialzeithypothese folgt, dass es keinen subquadratischen Algorithmus zur Berechnung der Fréchetdistanz gibt.

Bringmann, Karl: Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails, Max Planck Institute for Informatics, 2014, URL:<https://people.mpi-inf.mpg.de/~kbringma/paper/2014FOCS.pdf>



Zeitkomplexität

- Trajektorien lassen sich parallel klassifizieren.
- Die Klassifizierung einzelner Trajektorien lässt sich ebenfalls parallelisieren.



Zeitkomplexität

Eine Trajektorie im Metrischen Raum (M, d) heißt *c-packed curve*, wenn für alle Kugeln K in M gilt: Die Länge der Trajektorie in K ist kürzer als c mal der Radius von K .

Für c -packed curves lässt sich die Fréchetdistanz in $\mathcal{O}(c \cdot n)$ berechnen.

Driemel, Anne/ Har-Peled, Sariel/ Wenk, Carola: Approximating the Fréchet Distance for Realistic Curves in Near Linear Time, <https://arxiv.org/pdf/1003.0460.pdf>



Gliederung

- 1 Projektziele
- 2 Klassifikation und Situationsanalyse mit Ähnlichkeitsmaßen
- 3 Regressionsmodelle zur Vorhersage gefährlicher Situationen**



Ziel der Regressionsanalyse

Ziel: Kollisionen zwischen Verkehrsteilnehmern vorhersagen

Frage: Wann kollidieren zwei Objekte?

→ Zwei Objekte kollidieren zu einem Zeitpunkt t , wenn der Betrag des Abstands zwischen den beiden Objekten einen Schwellwert r erreicht oder unterschreitet. Der Schwellwert hängt von den Objekten ab.



Regressionsanalyse

Aufgabe: Bestimme zu einem Zeitpunkt $t = tA$, ob Objekte im Zeitraum $[tA + tP]$ kollidieren werden.

Vorgehen: Lineares Regressionmodell

$$x = v_x * t + x_0$$

$$y = v_y * t + y_0$$

Bestimme Modellparameter v_x , x_0 , v_y , y_0 , so dass Abweichung von gemessenen Positionen im Zeitbereich $[tA - tO, tA]$ minimiert wird.



Entwicklung Regressionsmodell

Genauigkeit der Vorhersage hängt unter Anderem ab von:

- tO (Observationszeitraum)
- tP (Vorhersagezeitraum)
- Art des Objekts
- Klasse der Trajektorie

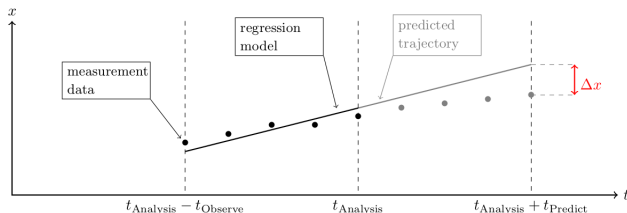
Entwicklung Regressionsmodell

Offline-Analyse: wir vergleichen unsere Vorhersage für einen zukünftigen Zeitpunkt $t' = tA + tP$ mit den bekannten, gemessenen Daten zum Zeitpunkt t' .

$\Delta x = \text{Vorhersage}(t') - \text{Messung}(t')$ und

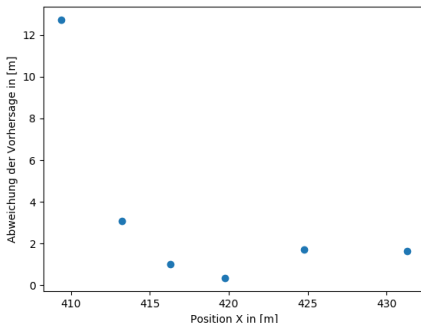
$\Delta y = \text{Vorhersage}(t') - \text{Messung}(t')$.

Untersuchen die Genauigkeit (Δx , Δy) der Vorhersage für verschiedene Zeiträume tO und verschiedene Zeiträume tP .



Wie genau ist das Regressionsmodell?

Abweichungen Δx in Abhängigkeit der Position x für Repräsentant der Äquivalenzklasse „Auto biegt vom Groß-Berliner Damm Richtung S-Adlershof ab“





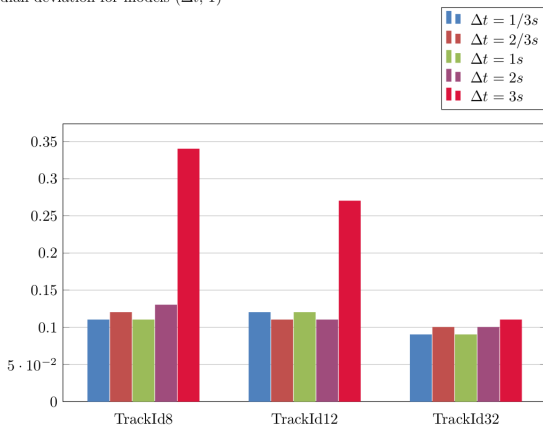
Wie genau ist das Regressionsmodell?

- die berechneten Abweichungen unterscheiden sich stark
- es kann deutliche Ausreiser geben
- bei der Medianberechnung werden Ausreiser nicht so stark gewichtet wie bei der Mittelwertberechnung
- um ein gutes Maß für den Vergleich der Abweichungen zu finden, berechnen wir jeweils den Median für alle Repräsentanten und alle Regressionsmodelle

Vergleich vers. Observationszeiträume

- Berechnete Mediane aller Abweichungen Δx für jeden Repräsentanten der Klasse Fußgänger

Median deviation for models (Δt , 1)





Gewähltes Regressionsmodell

- Verhalten ändert sich schnell, sodass ein Observationszeitraum $tO \geq 2s$ zu großen Abweichungen führt
- Verhalten ändert sich nicht so schnell, dass ein Observationszeitraum $tO < 1s$ zu besserer Genauigkeit führt
- Verhalten ändert sich schnell, sodass ein Vorhersagezeitraum $tP \geq 2s$ zu großen Abweichungen führt
- wir wählen den Observationszeitraum $tO = 1s$ und den Vorhersagezeitraum $tP = 1s$



Kollisionsberechnung

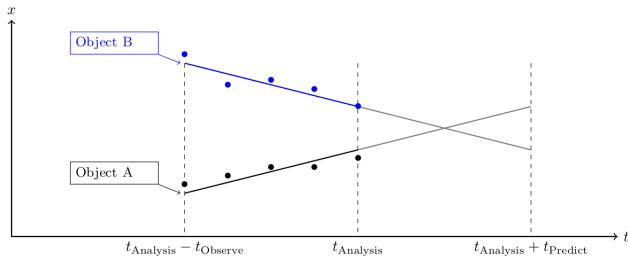
- fitten unsere Daten mit Geraden
- Geraden verlaufen entweder parallel oder schneiden sich genau ein Mal
- wir berechnen die Zeitpunkte, zu denen sich unsere Geraden schneiden:

$$t_x = \frac{x_A - x_B}{v_{xB} - v_{xA}}$$

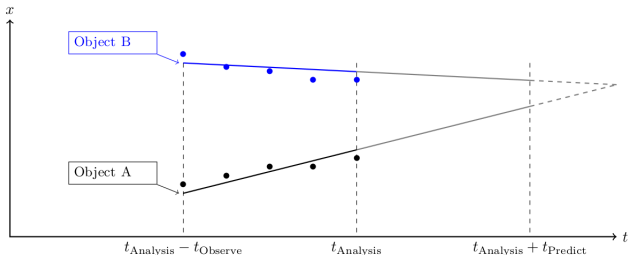
$$t_y = \frac{y_A - y_B}{v_{yB} - v_{yA}}$$

- es kommt zu einer Kollision, wenn die Zeitpunkte t_x und t_y nah beieinander liegen

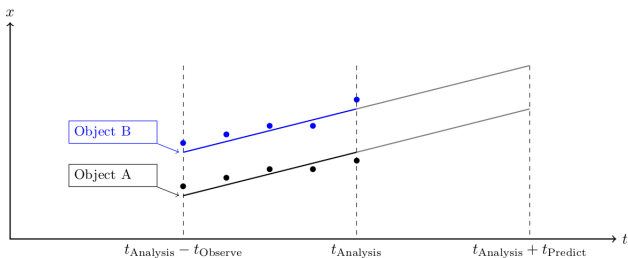
Fall 1: Geraden schneiden sich im Vorhersagezeitraum



Fall 1: Geraden schneiden sich außerhalb des Vorhersagezeitraums



Fall 3: Geraden verlaufen parallel





Stellschrauben bei der Kollisionsberechnung

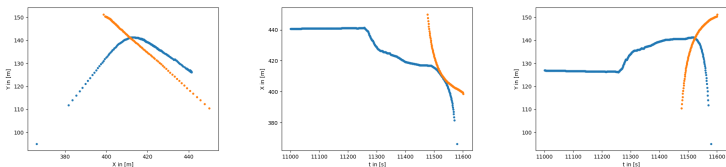
Berechnung der Kollision hängt unter Anderem ab von:

- maximale, erlaubte Differenz zwischen t_x und t_y
- Toleranzbereich um den Schnittpunkt, der die Ausdehnung der Objekte berücksichtigt
- Toleranzwert, für sich nebeneinander (parallel) bewegende Objekte

Testen der Kollisionsberechnung

Um die Berechnung möglicher Kollisionen zu testen, wurden die Zeitstempel erfasster Trajektorien so geändert, dass es zu einem Zusammenstoß kommt.

Abbildung: Kollision zwischen Auto (blau) und Fahrrad (orange)





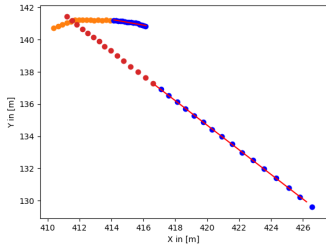
Wahl der Stellschrauben

- für den Unterschied der Kollisionszeitpunkte gilt:
 $|t_x - t_y| \leq 0,5s$
- der Toleranzbereich um den Schnittpunkt wurde mit 1m festgelegt
- für die Unterschiede der Geschwindigkeiten gilt:
 $abs(v_{x_A} - v_{x_B}) < 0,1$ und $abs(v_{y_A} - v_{y_B}) < 0,1$

Ausgabe des Algorithmus

Der Algorithmus erkennt eine drohende Kollision in folgendem Analyseschritt:

Abbildung: Erkennung einer Kollision innerhalb der nächsten Sekunde: Messdaten zum Fitten (blaue Punkte), Regressionsgerade (rote Linien), zukünftige Messdaten (rote und orangene Punkte)

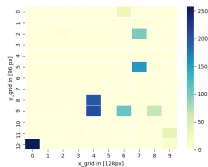
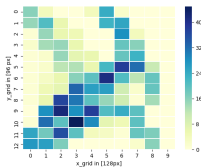


Heatmaps

Kollisionsvorhersage ist aufwändig, deshalb wollen wir das speziell für gefährliche Bereiche machen.

Annahme: gefährliche Bereiche sind Bereiche mit vielen Verkehrsteilnehmer und/oder hoher durchschnittlicher Geschwindigkeit

Abbildung: Heatmap: Anzahl der Verkehrsobjekte pro Grid (links) und durchschnittliche, absolute Geschwindigkeit pro Grid (rechts)





Ergebnisse bei der Anwendung der Kollisionserkennung auf Datensatz

- die Daten beinhalten keine Kollisionen
- die gewählten Stellschrauben für die Erkennung der Kollisionen konnten deshalb nicht auf Praxistauglichkeit geprüft werden
- im Datensatz konnten keine Kollisionen vorhergesagt werden



Verbesserungsmöglichkeiten für zukünftige Arbeiten

- Daten mit Kollisionen zur Analyse
- Schwellwerte optimieren
- Schwellwerte objektabhängig wählen
- positionsabhängiges Regressionsmodell entwickeln



Quellen

- Nehal, Magdy: Review On Trajectory Similarity Measures, 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems, URL: <https://www.researchgate.net/publication/293823957>
- Eiter, Thomas/ Mannila, Heikki: Computing Discrete Fréchet Distance, Christian Doppler Labor für Expertensysteme, Technische Universität Wien, URL: <http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdtr9464.pdf>
- Bringmann, Karl: Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails, Max Planck Institute for Informatics, 2014, URL: <https://people.mpi-inf.mpg.de/~kbringma/paper/2014FOCS.pdf>
- Driemel, Anne/ Har-Peled, Sariel/ Wenk, Carola: Approximating the Fréchet Distance for Realistic Curves in Near Linear Time, <https://arxiv.org/pdf/1003.0460.pdf>