Big data needs big brains: using new data sources in traffic engineering

Victor L. Knoop 12 December 2023

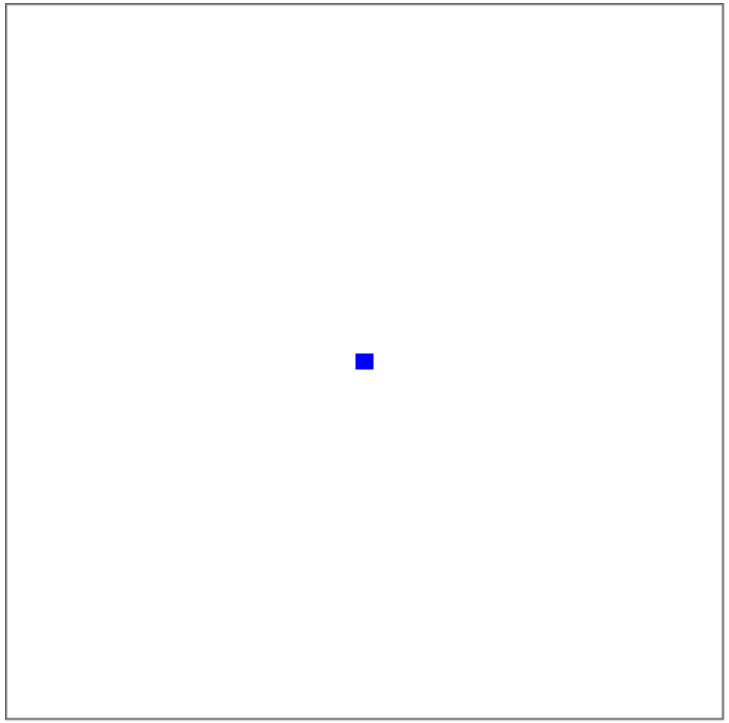
TU Delft

Introduction

- More urbanisation
- (Hence) more traffic and traffic jams
- Automated driving
  (requires and generates data)
- More data and more computing power
- Can data be the solution to the main questions in transport?

How to study traffic

- Theoretical approaches
- Cellular automata
- Simulation models
  - (simplified to have the right mathematical properties)
- First data points
- “We need more data”
- We now have
- Data driven methods
  - No need for domain knowledge

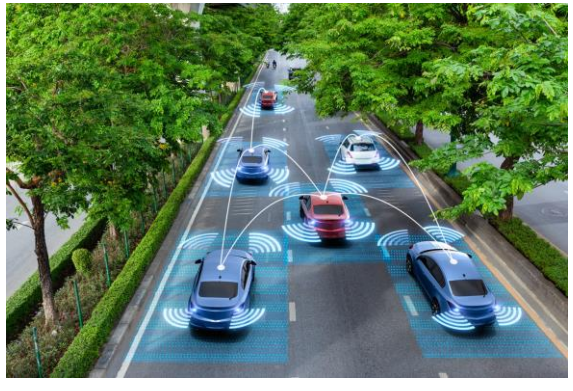


From data poor to data rich

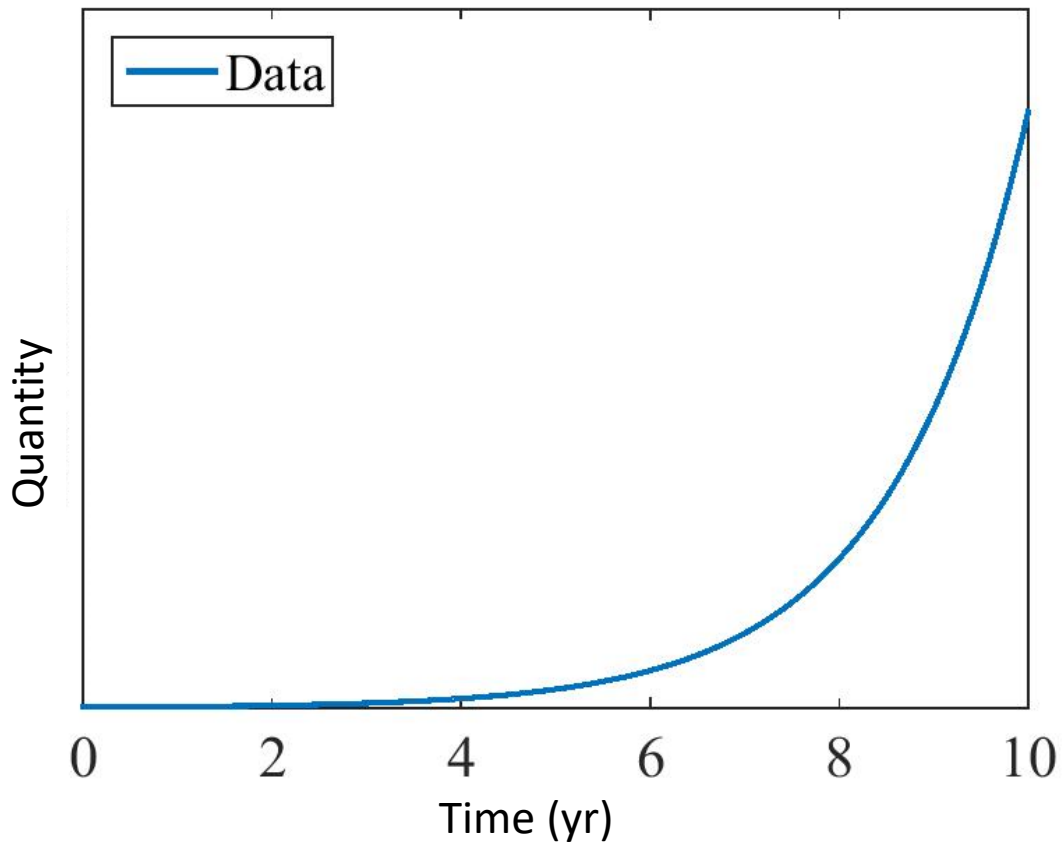
- When I started:

“Our field is data poor and assumption rich”

- This has changed in the meantime...
- All traffic is observed by loop detectors or cameras
- Helicopters and drones
- Vehicles share their position (and speed)
- Vehicles sense their surroundings

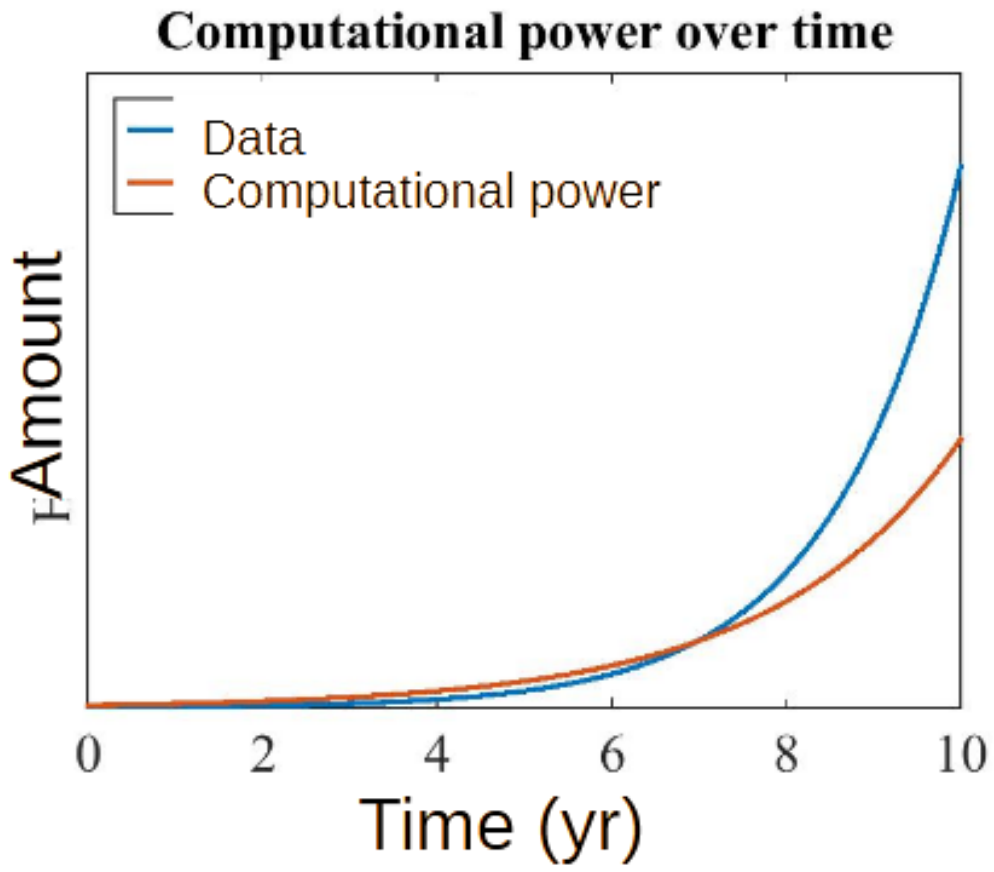


Data quantities

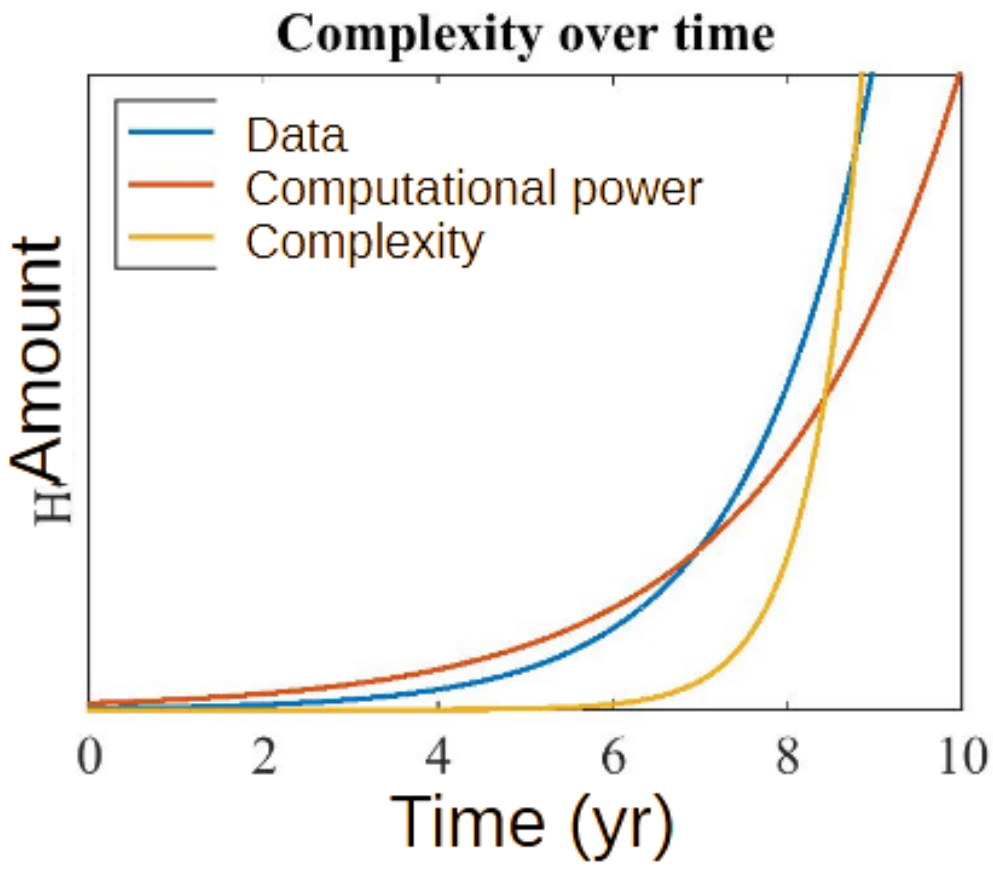


- Data

Data quantities



Data quantities



Value of data

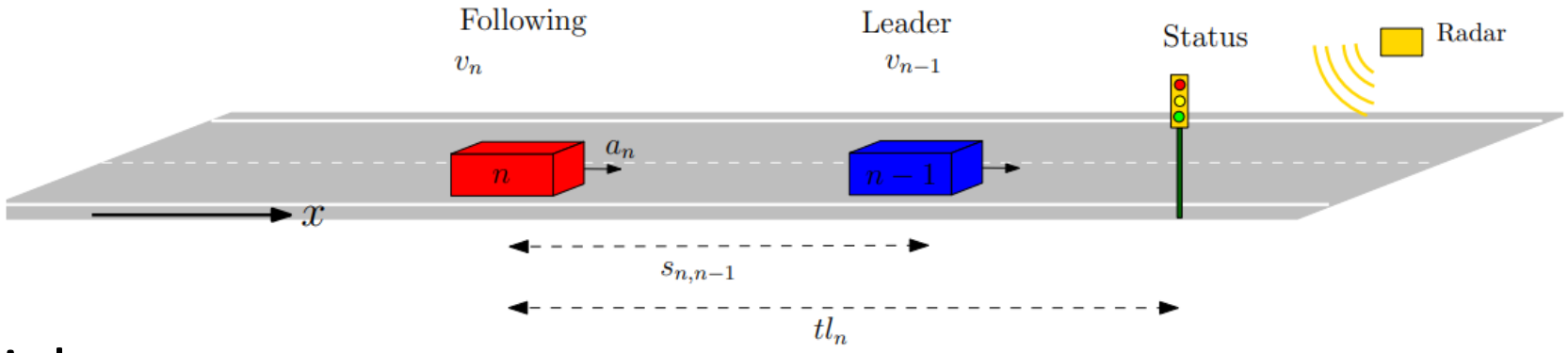
- Only what is new (speed sampling at 100MHz?)
- Only what adds to the knowledge base (another day of the same data?)

- What can we forget?
- What should we value
- What should we collect more

Example 1: car-following near traffic lights

- How do people approach traffic lights?
- Data: radars near a traffic light, detecting individual cars

Available data

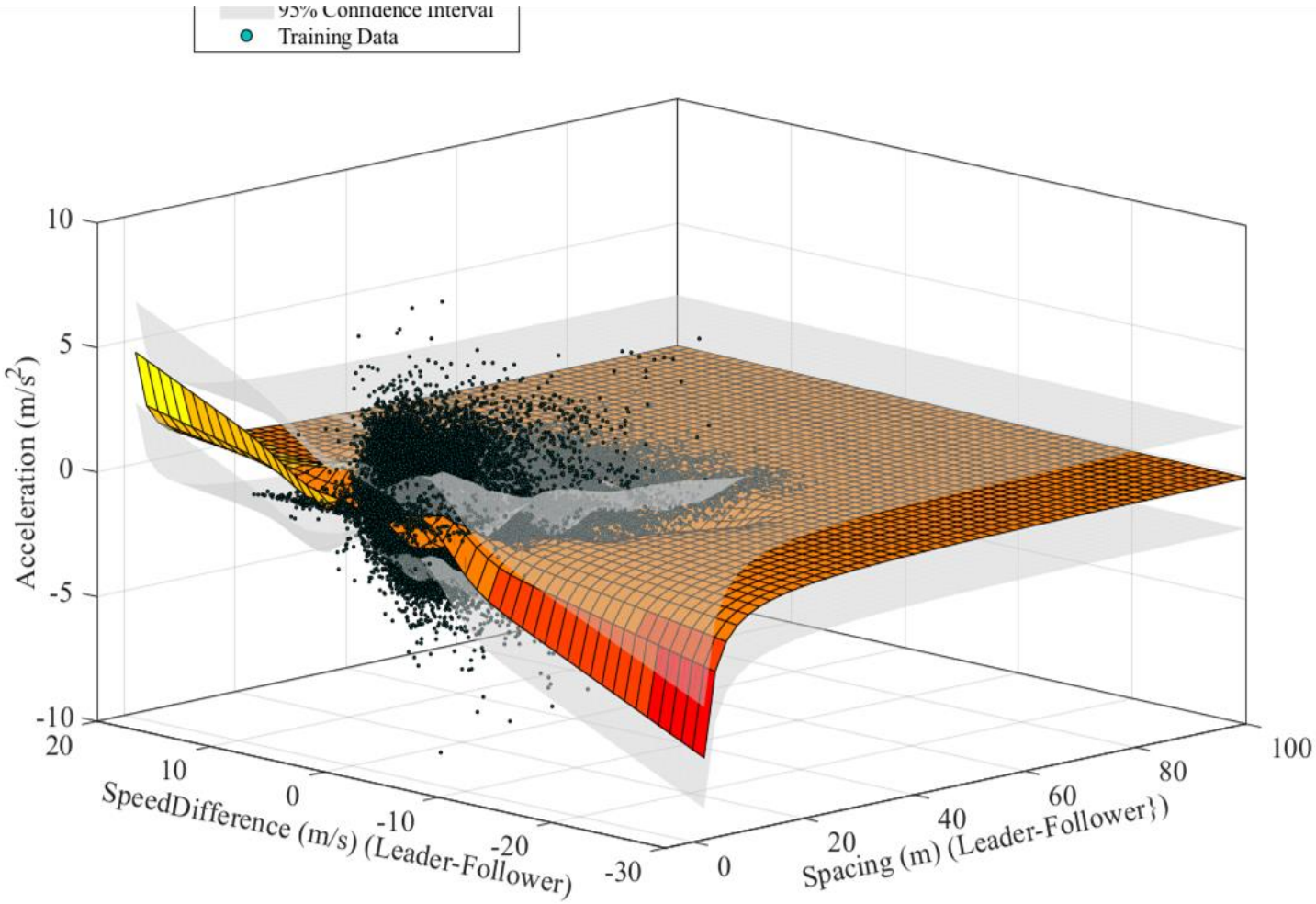


- Radar data at traffic light,
  used to analyze how people interact
  - the vehicle in front
  - the traffic light
- Various “known” models and a data-driven model
- Enabled us to find most relevant parameters
- And a mixture model

TU Delft

Smart use of big data

- Most important:
  - Speed difference
  - Spacing
  - Traffic light color
- Model decision on the use of data prediction or rule based model



TU Delft

Macroscopic traffic predictions and the limit of predictability

The role of traffic predictions

- Informing people on the traffic state is useful
- Different compared to weather predictions: the weather is not influenced by the predictions
- Predictions can be used to
  - Advise travellers on postponing/cancelling trip
  - route traffic
  - Advise drivers on unsafe situations
  - Advise drivers to actively do something (e.g., change lanes)
  - Intervene in automated vehicle (predictions on a different scale?)

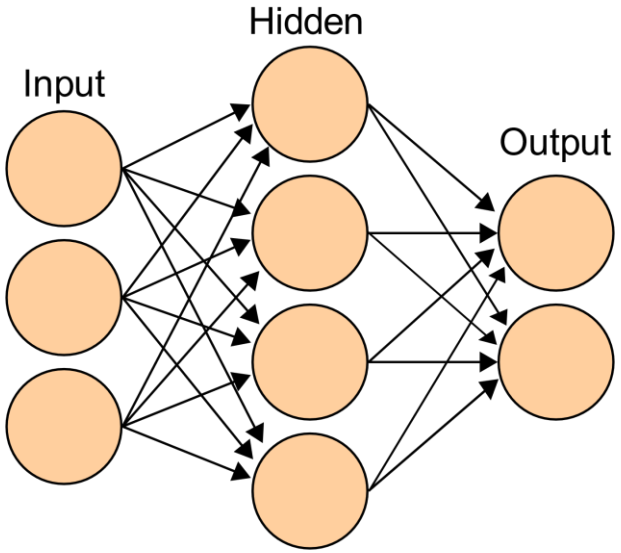
TU Delft

Common traffic prediction

- Up to 15 years ago: traffic model
- Since then: data-driven methods
- E.g., train a neural network

(because it can do anything)

  - Problem: it learns what we already knew
  - Or if designed poorly, it does worse



TU Delft

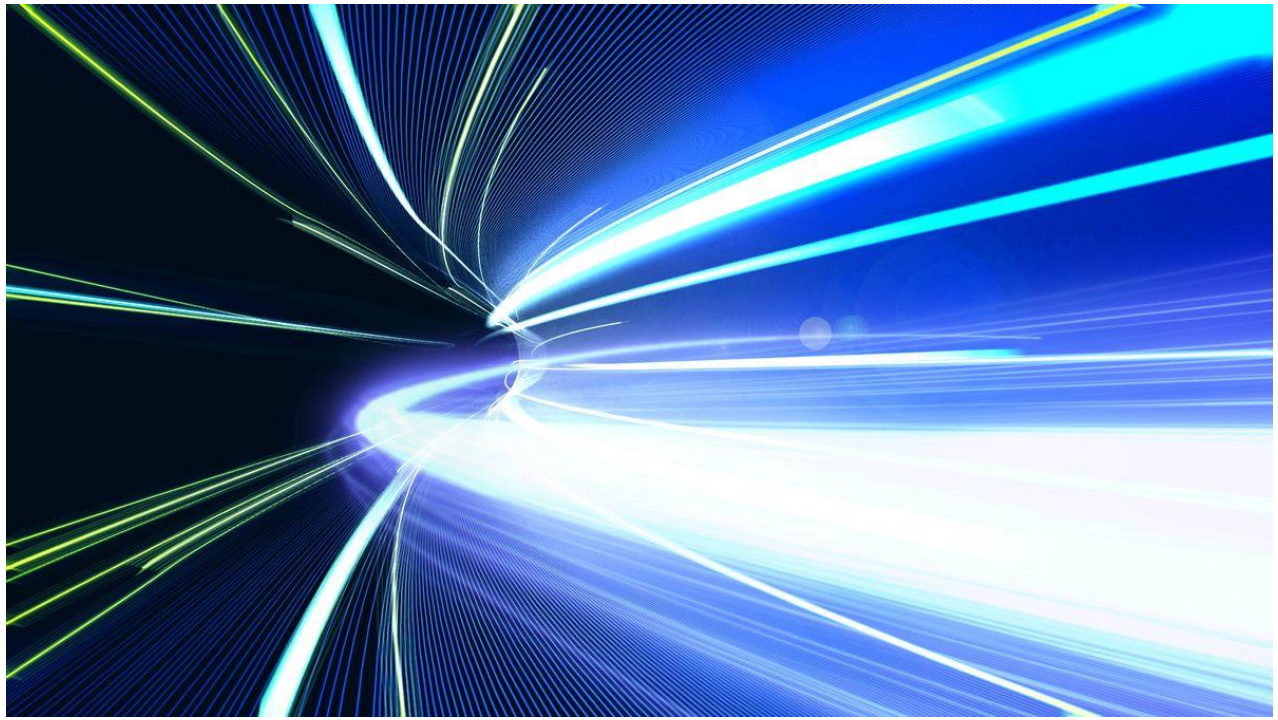
Example

- Traffic data is much available
- Learn how the traffic states evolves from one state to the next
- Long training times
- Limit influence to next-neighbor detectors
- *Traffic information goes upstream and downstream*
- That should be included, if not, predictions are bad
- Limit to next neighbor at each side:

back at cell transmission model which was already known

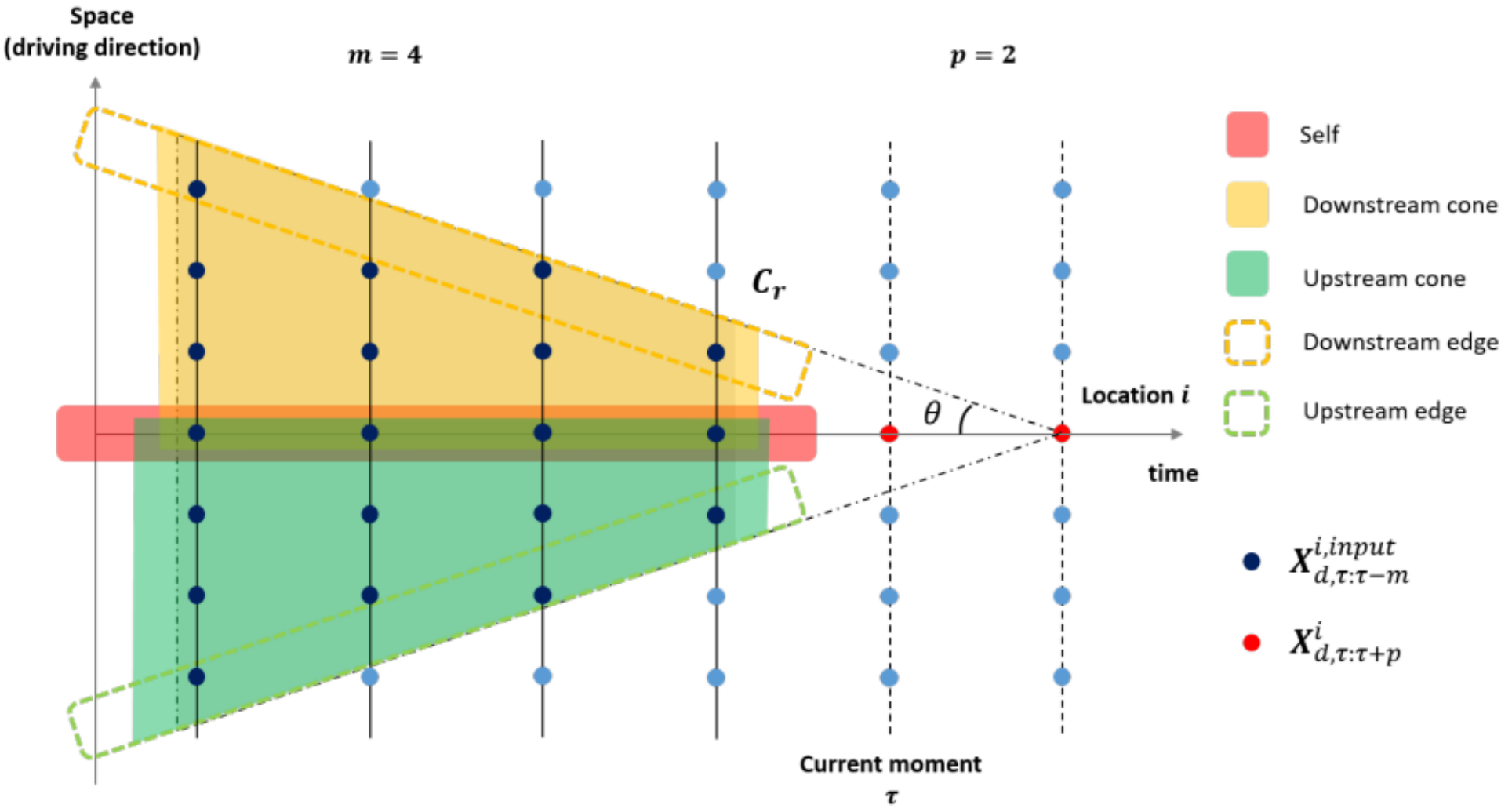
Which information to include

- Speed of information is limited, we know from decades of traffic flow theory
- No need to include all observations: limit in space and time



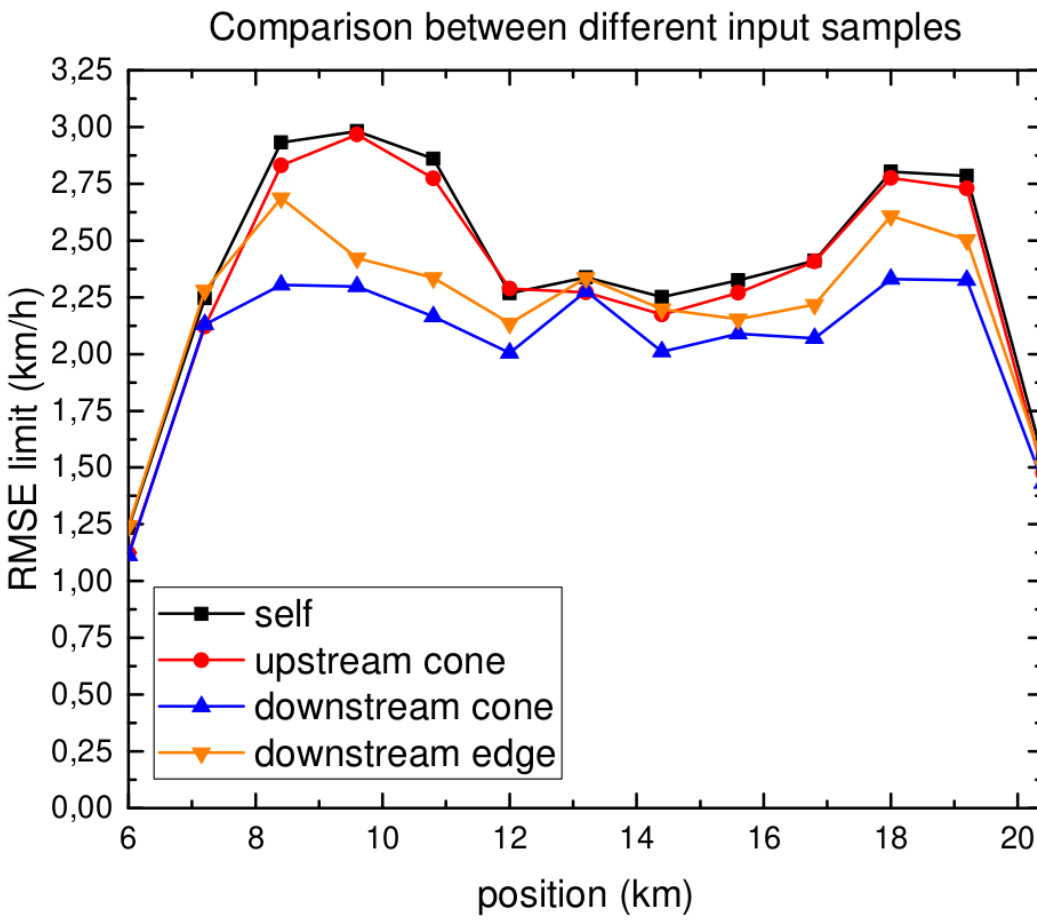
TU Delft

5 different inclusion possibilities



Results

- Self is worst (least information)
- congestion travels upstream, so:
  - Upstream hardly gives information
  - Downstream cone gives more information than just the edge (so different speeds of information)



TU Delft

Limits to predictability

Errors and unknowns

- Every deviation from our prediction is an error
- Deterministic view: improve prediction further and you'll end up without an error
- Traffic engineering mind: collect more data, fit more refined models and improve
- At first: limit in predicting traffic state more than ~30 mins ahead
- Natural limit: length of the trip

Errors due to model or in process

- Consider dice: that is hard to predict well
- The process itself is stochastic, and no deterministic model can make a good prediction
- Collecting more data will not help in accurately predicting dice
- Question: are errors in traffic of the nature of dices or because of badly chosen models?

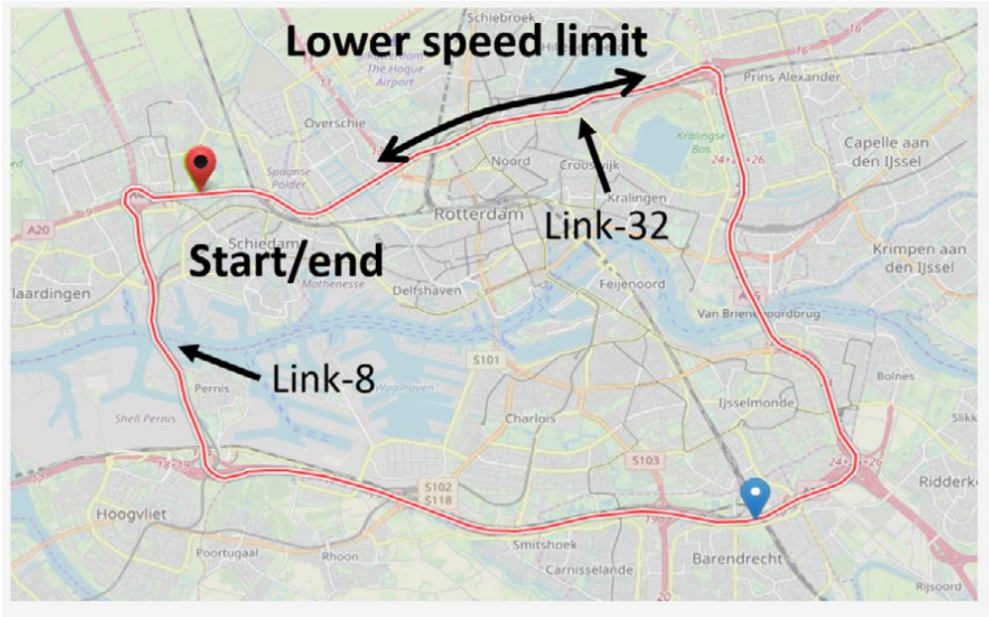
TU Delft

Lower bound of model error

- Regardless of model, check the uncertainty in the data
- Possible due to the amounts of data available
- Use entropy to find this:
  - Find similar traffic states
  - Check how the future diverges
- Compare the minimum with the best models
- Do so for
  - Deterministic models (= predicting one value)
  - Stochastic models (=probabilistic models)

Lower bound of model error

- Network:
Rotterdam
ca 500,000 inhabitants
- ~3-4 lane freeway
ca 10x10 km

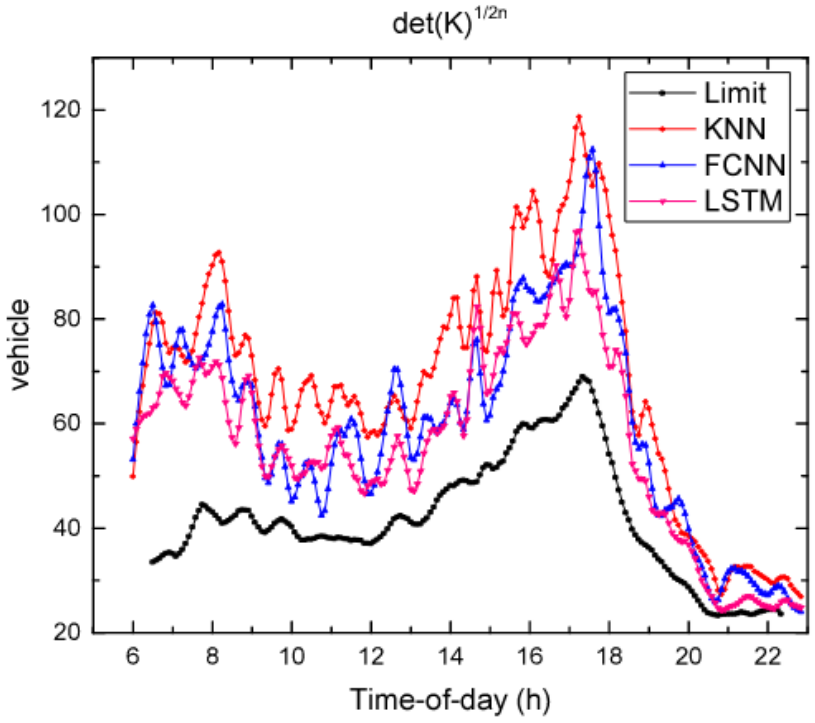


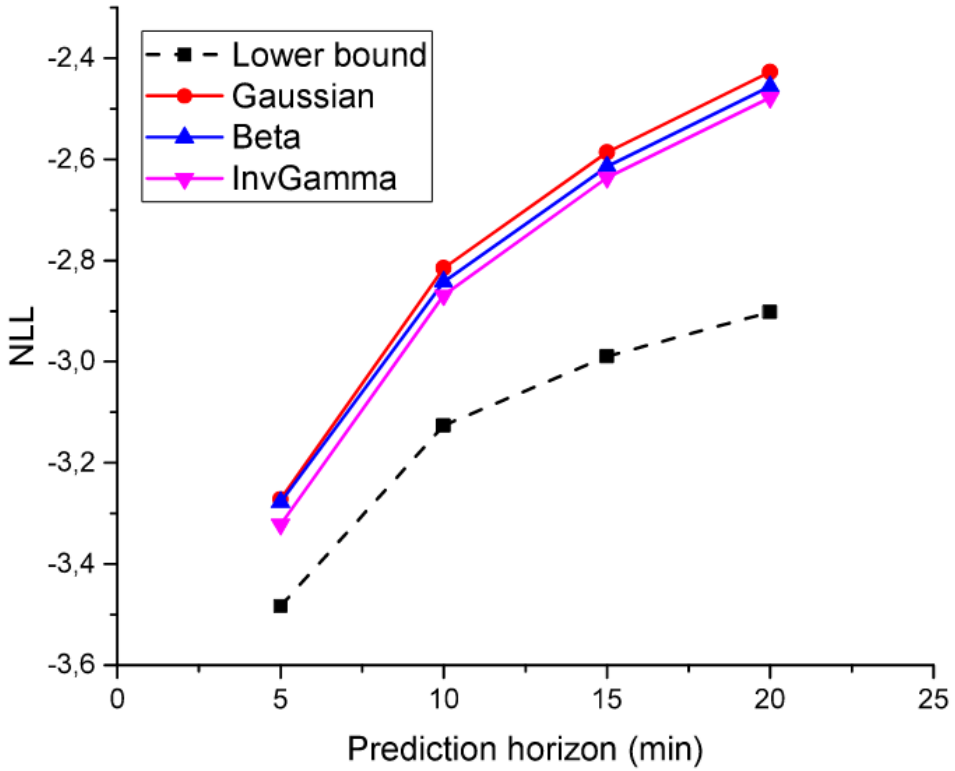
- 2 test cases:
  - Nr of vehicles in the full network (1 dimension)
  - Speeds at all locations (35 dimensions)

TU Delft

Lower bound of single variate

- For stochastic models, the assumed distribution matters…but has little effect
- Close to lower bound with current approaches

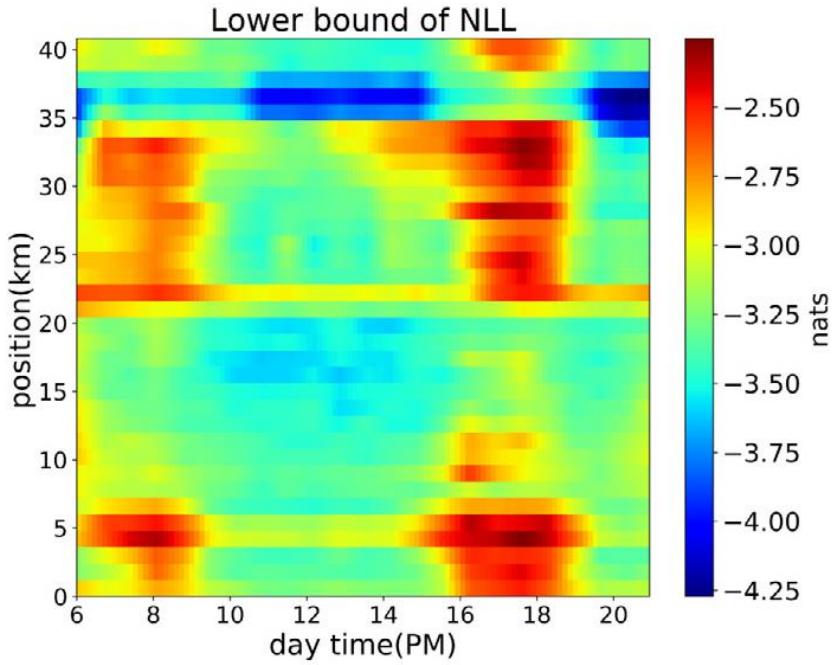


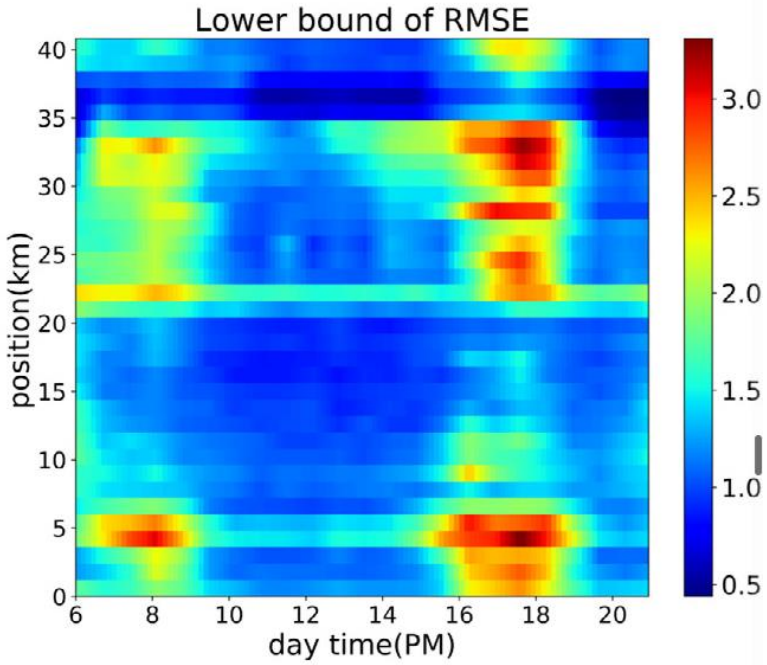


TU Delft

Lower bound multi-variate model

- Speed at all locations
- Lower bound depends on traffic state: sometimes (in peak periods) states are uncertain (mostly near traffic breakdown)





TU Delft

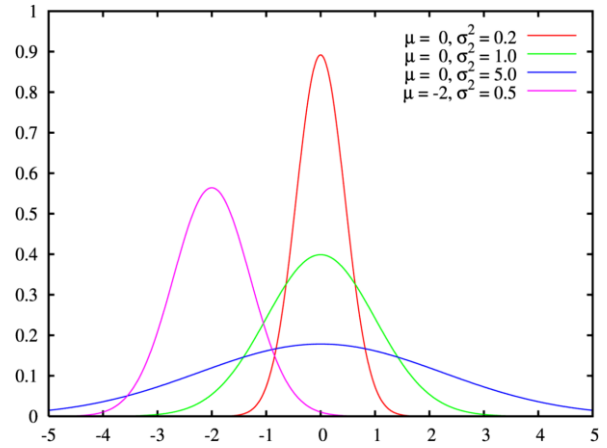
Prediction should be a distribution

- Up to now: next speed(s) have been predicted
- Error as a value or a likelihood in a pdf that the right value has been predicted
- We know now that traffic predictions are inherently stochastic
- Therefore, let's predict the probabilities for a single prediction

Example: current state

1
2
5
7

output →



TU Delft

Two types of uncertainty

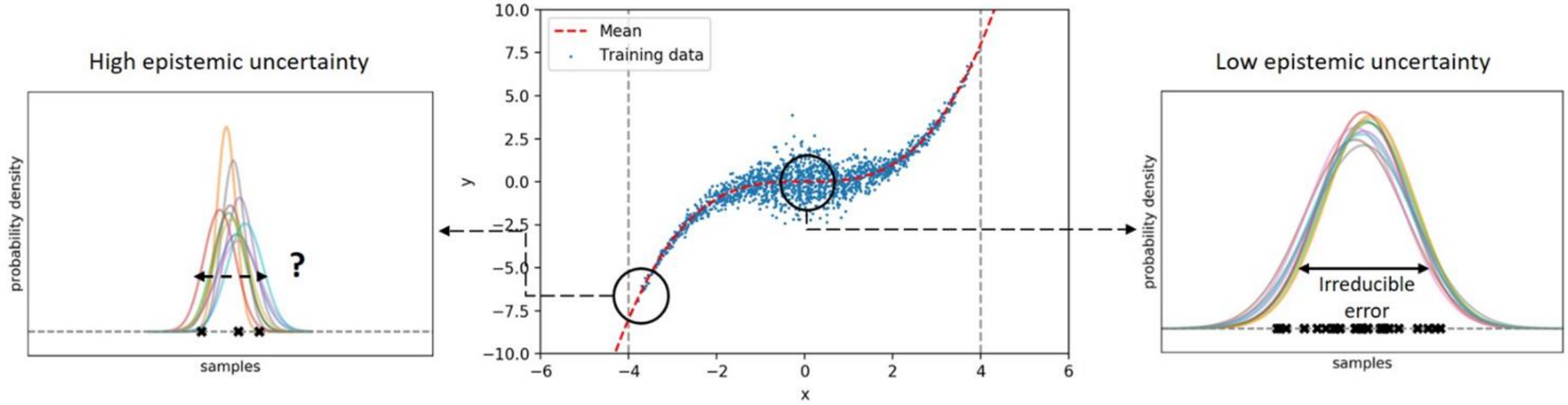
- Aleatoric – process is random (from alea, die)

Even rolling many, many times,
we can never predict dice

- Epistemic – we do not know enough

One (or zero) observations of an unknown case gives not sufficient information

Example

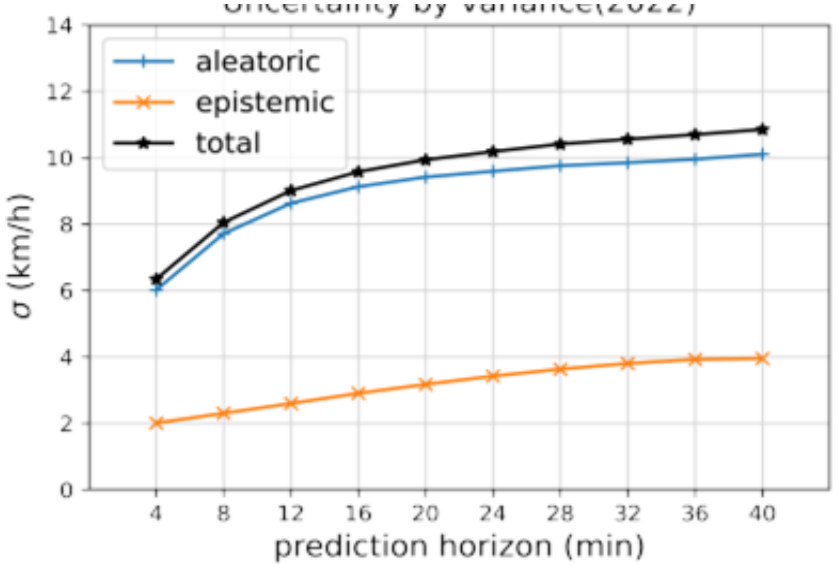


- Many observations around x=0,
  but unknown process, so uncertain (aleatoric)
- Few observations at boundary,
  so uncertain (epistemic)

How does this work for traffic prediction

- Quantify the sources of uncertainty due to each cause
- Test on speed prediction for ringways around Amsterdam
  - 193 links
  - Network
  - 2x 1 month of data
  - 1 minute aggregate
  - Prediction: 4-20 minutes ahead

Resulting uncertainties



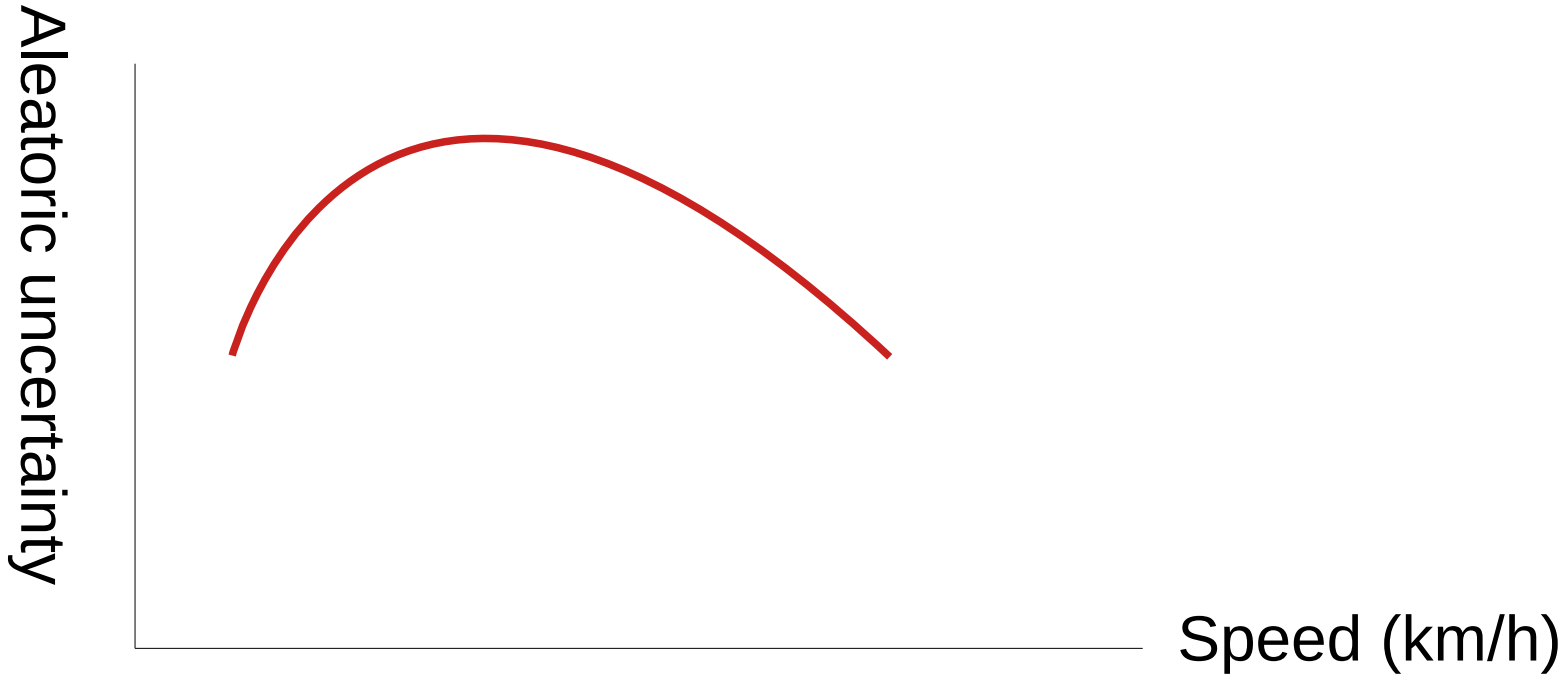
- Uncertainties increase with increasing time horizon (as expected)
- Total uncertainty almost fully caused by aleatoric uncertainty (very interesting new finding)

TU Delft

More data?

- Some very rare cases contribute to uncertainty
- 4 cases in 2022, 7 cases in 2019
- These are very rare events; more data of these, whatever they are, would be useful
- Typically, rare events happen not often (that information actually contains information :)

How do they depend on speed?



- For high speeds: certain (speeds will remain high)
- For low speeds: certain (speeds will remain low)
- Near critical speed: highest uncertainty (might go into breakdown)

TU Delft

How do they depend on location?

- We tracked all links and considered where the highest uncertainty came from
- Highest uncertainty near onramps causing jams (another indication that the start of a traffic jam gives highest uncertainty)

TU Delft

Microscopic data

- Similar methodology to car-following data set of autonomous vehicles
- >90% of the data consists of “the same”
- Outliers are rare, yet hence very valuable
- Easy distinction in hindsight
- Automated driving can focus on the >90%
  - Challenge is to determine the remaining part on beforehand (difficult)
  - Mixture models combining data driven and model based traffic are an option

Lane change detection from GPS data

Using GPS data to find lane changes

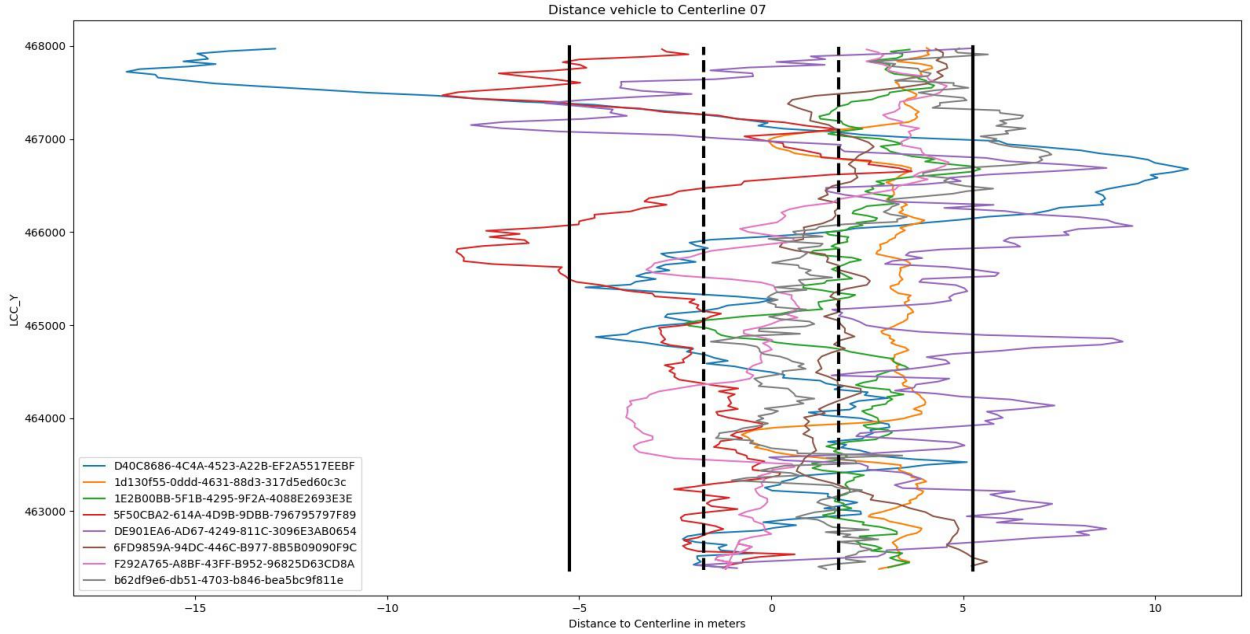
- Finding lanes is hard and requires data fusion including detector data, we know from Arman
- Lane *changes* might be easier, since the errors in absolute position might be correlated

Road stretch

- ~13 km
- 3 lanes
- Relatively few on and off ramps

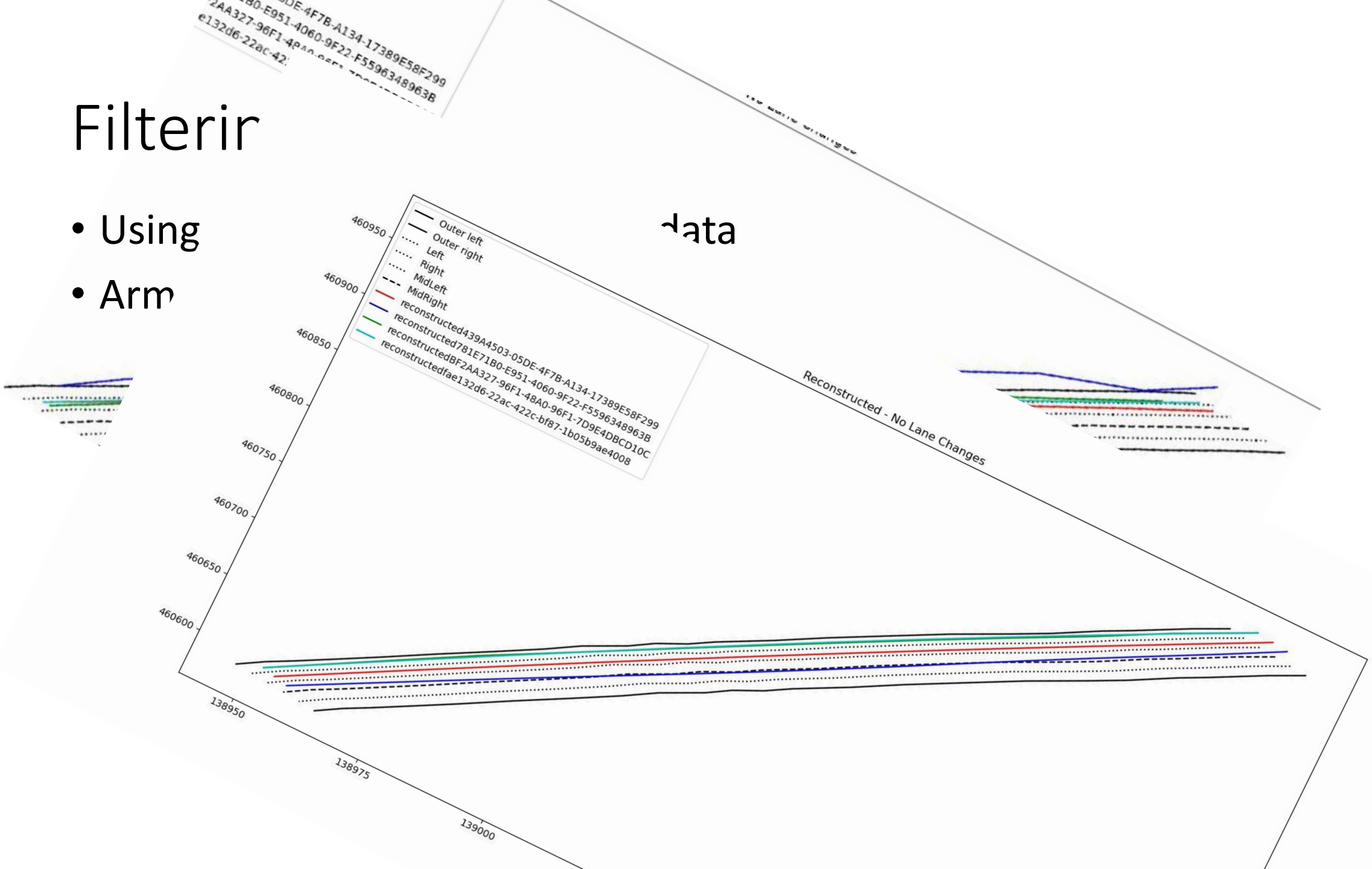
Data sources

- 1Hz GPS data from phones Flitsmeister
- For ground truth: individual loop detector data
- Approximately 1 month of data



Filterir

- Using
- Arm



Lane change detection: algorithmic

- Consider the heading of the vehicle compared to the road axis
- For 3 subsequent time steps (seconds) it should change in either the positive or negative direction
- Besides, the total rotation in these ~~3~~ 4 seconds should be at least 6 degrees
- Results exceed flipping a coin (but not by much)

YES lane change Loop detector method	1024	15365
NO lane change Loop detector method	732	16283
	YES lane change Delta heading method	NO lane change Delta heading method

Data driven

- Data preparation
  - equal number of data points
  - Analysis balanced for number of occurrences
- Random forest on features
- Features: speed, x-distance, y-distance, heading, heading difference to previous data point, heading difference to the centerline
- only instantaneous, hence no subsequent time steps

Data driven

- 4 models:

Labels
Yes / No
Left / No / Right
Left / No + Right
Right / No + Left

Data driven

- 4 models:

Labels	Testing Accuracy	Validation Accuracy
Yes / No	60.61 %	62.02 %
Left / No / Right	48.84 %	50.89 %
Left / No + Right	63.98 %	61.10 %
Right / No + Left	64.50 %	60.26 %

- Data driven exceeds algorithmic methods, even without previous time steps
- Most important features:
  - the heading of the vehicle,
  - the lateral distance between the vehicle and the centerline of the road
- Clear improvement over coin flipping
- Combining headings increases correctness

Finding traffic densities

Introduction

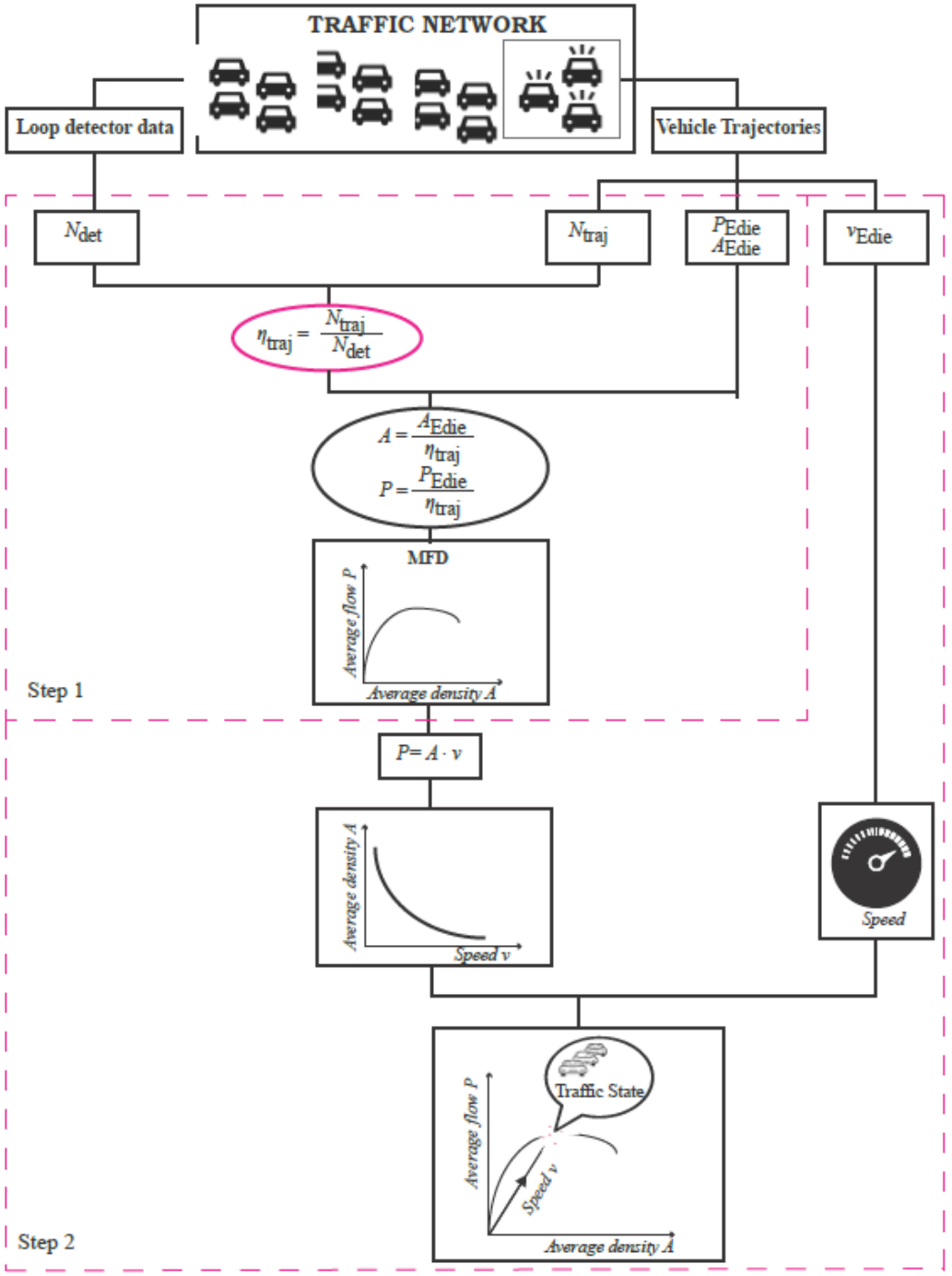
- Macroscopic traffic states are often estimated by using data from loop detectors
- Aim of (Dutch) road authority is to phase out as much road side equipment as possible
- With *floating car data* speeds are easily found
- Number of vehicles (flows/volumes/intensities or density/accumulations) are harder

Method

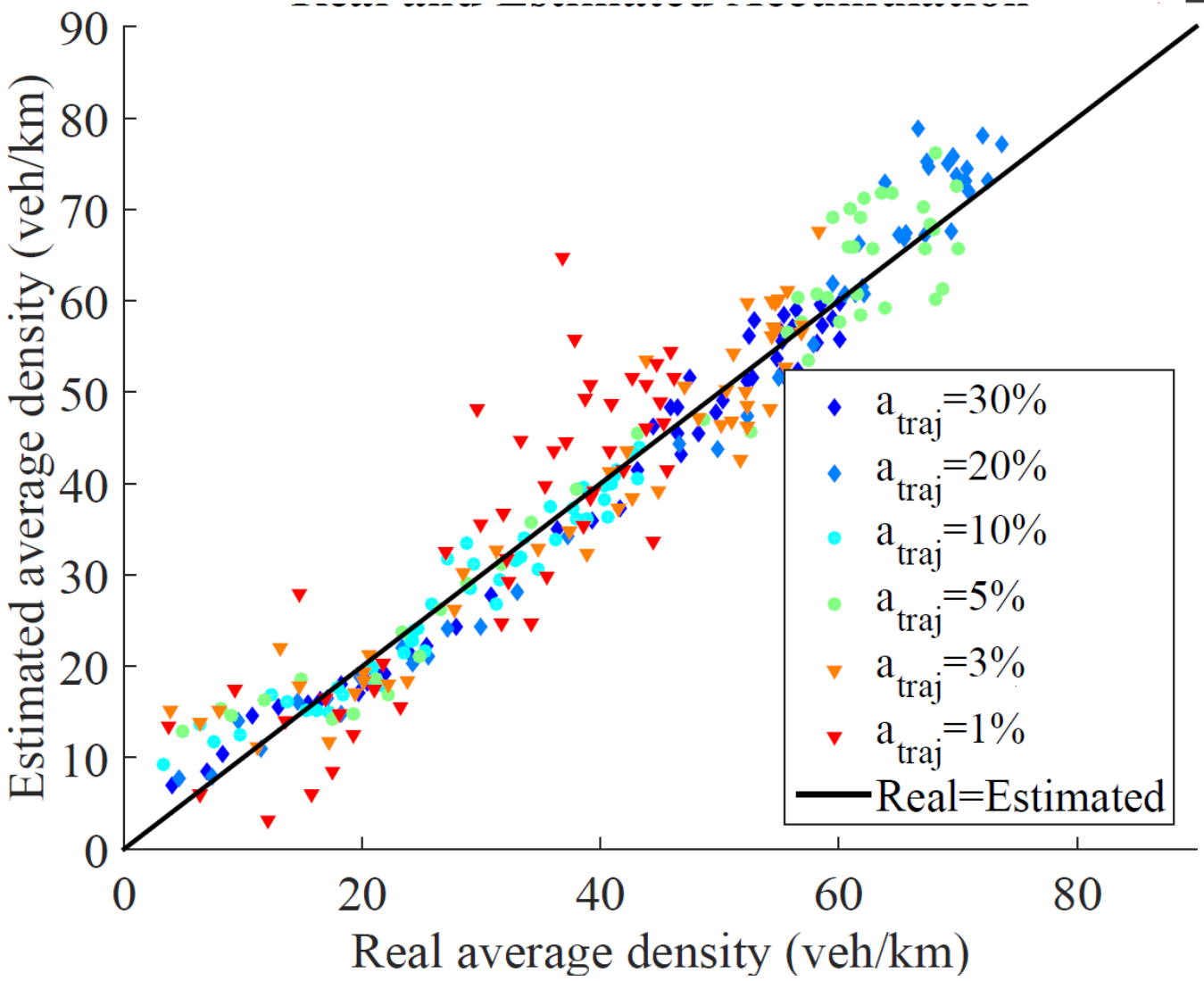
- Determine a penetration rate and multiply by the penetration rate (penetration rate = fraction of drivers sending information)
- Use the speeds to – once calibrated – determine the flow
- (Mermygka and Knoop)

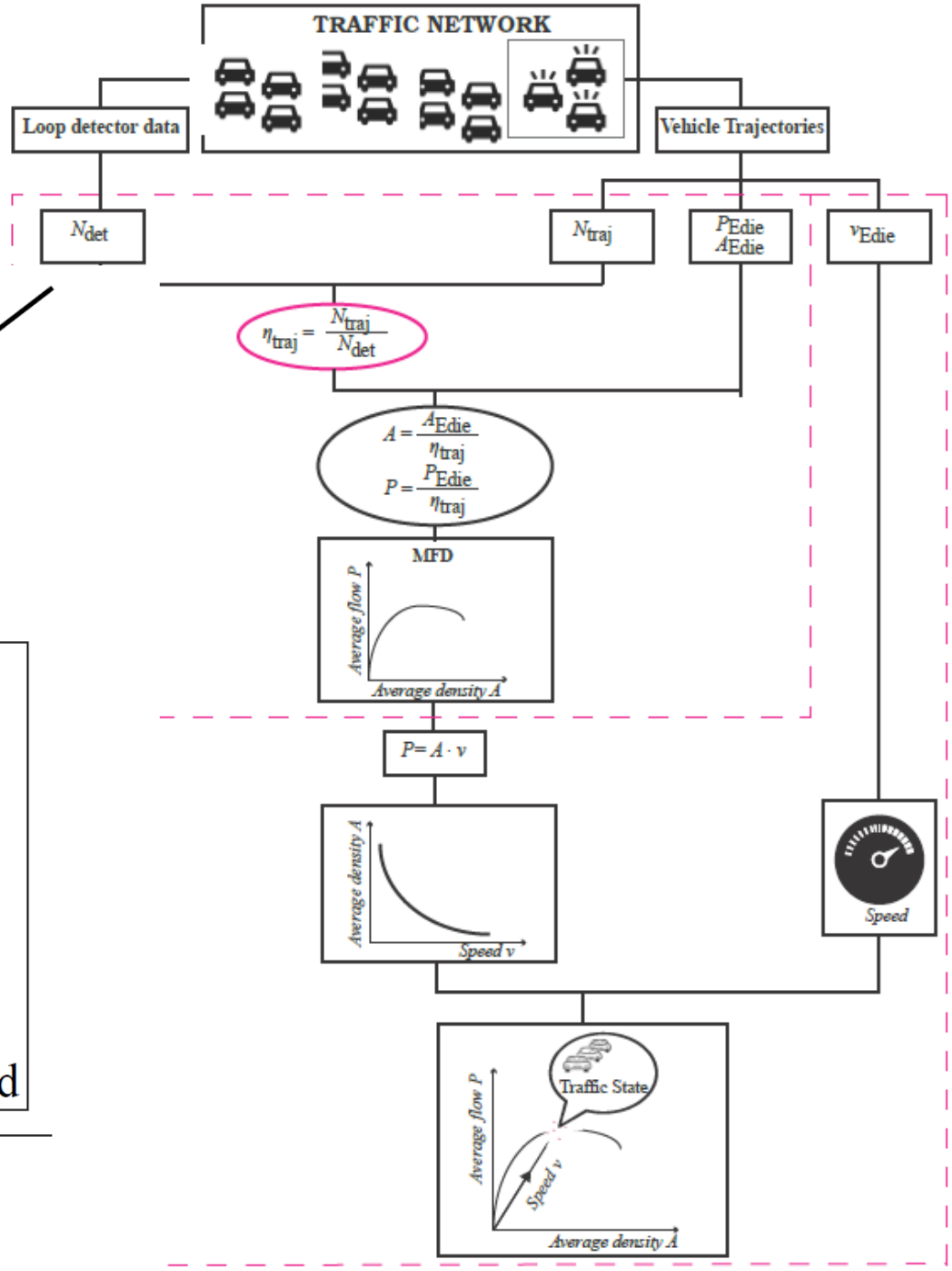
Works quite OK for larger areas, yet for a single road not ideal

Simple methods

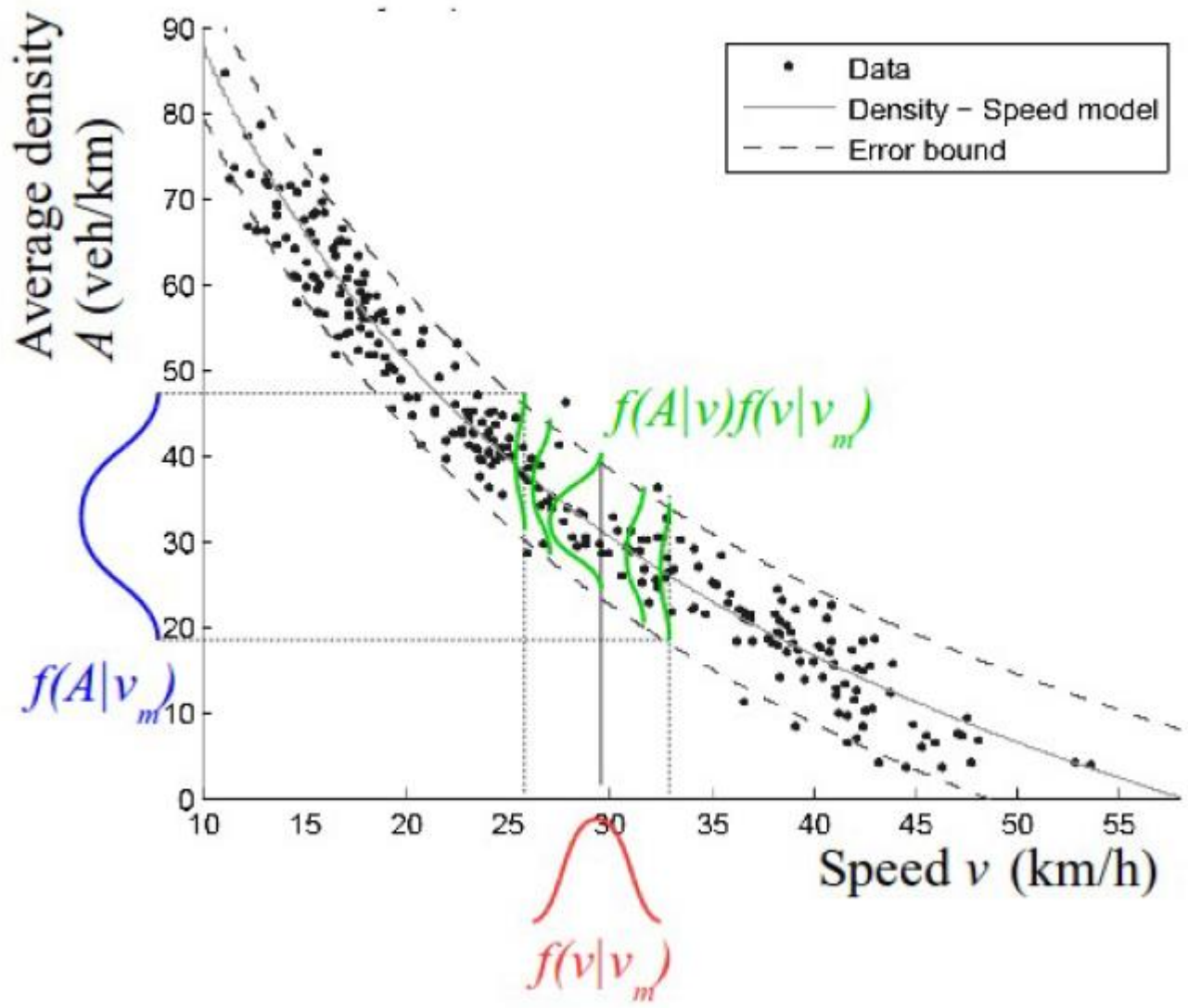


Results


Estimated average density (veh/km)
Real average density (veh/km)
0
10
20
30
40
50
60
70
80
90
0
20
40
60
80
a_{traj}=30%
a_{traj}=20%
a_{traj}=10%
a_{traj}=5%
a_{traj}=3%
a_{traj}=1%
Real=Estimated

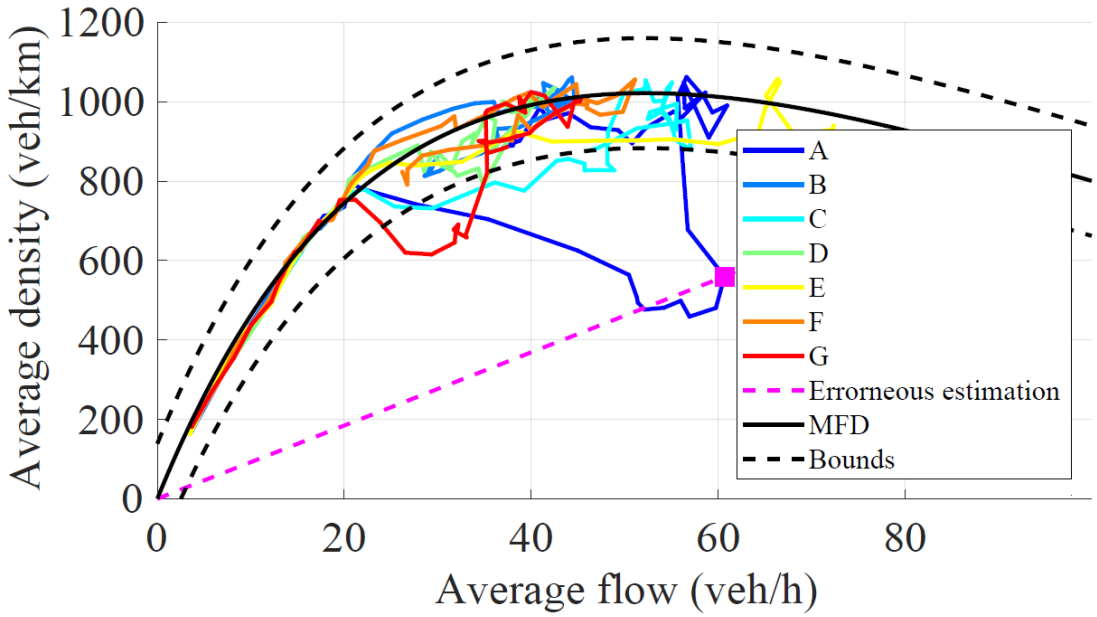

TRAFFIC NETWORK
Loop detector data
Vehicle Trajectories
N_{det}
N_{traj}
P_{Edie}
A_{Edie}
v_{Edie}
$\eta_{traj} = \frac{N_{traj}}{N_{det}}$
$A = \frac{A_{Edie}}{\eta_{traj}}$
$P = \frac{P_{Edie}}{\eta_{traj}}$
MFD
Average flow P
Average density A
$P = A \cdot v$
Average density A
Speed v
Speed
Traffic State
Average flow P
Speed v
Average density A

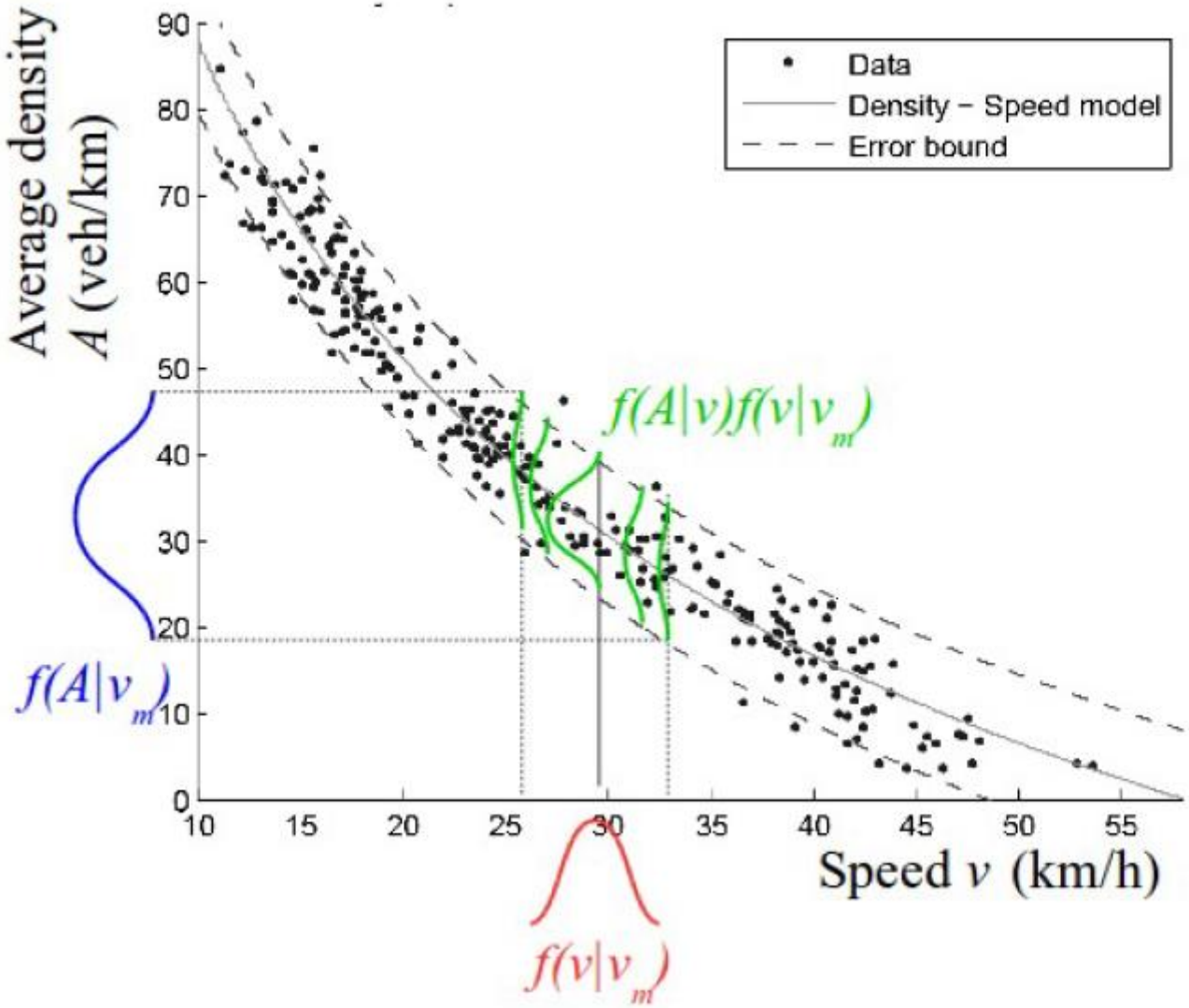
Uncertainty



Deviations & accuracy

- Works mostly relatively well
- Breaks if local disruptions due to e.g. accident





Value of relative flow data

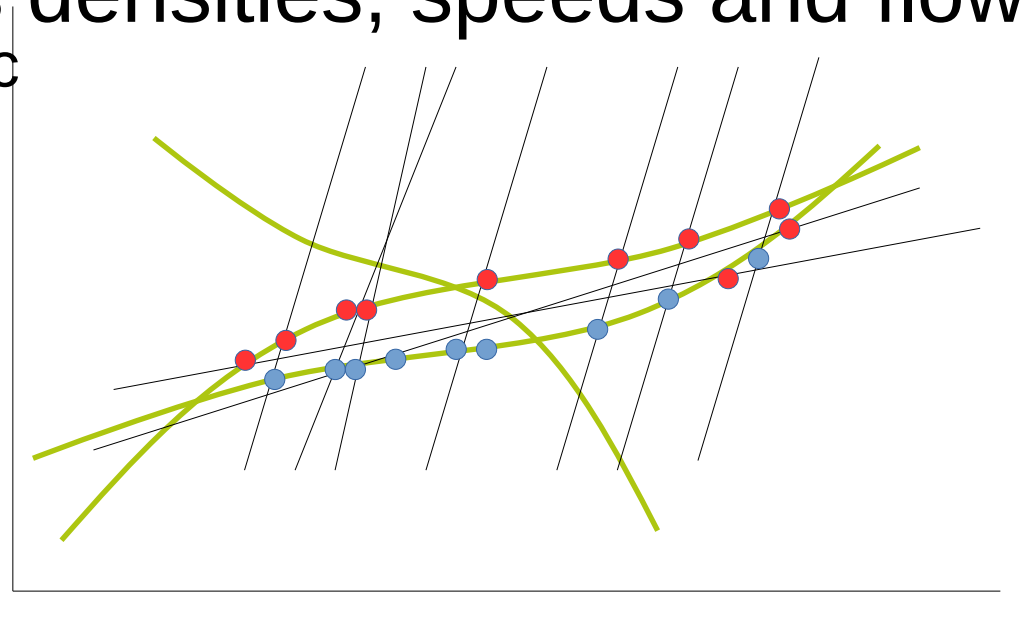
TU Delft

Measuring traffic via vehicles

- Combination of loops and Floating Car Data (FCD) works well
- FCD gives many speeds
- Loops give flows
  (“intensities”, volume ie. number of vehicles)
- Flows are essential for traffic prediction
- Speeds do not (at all) indicate whether traffic is near breakdown or in free flow

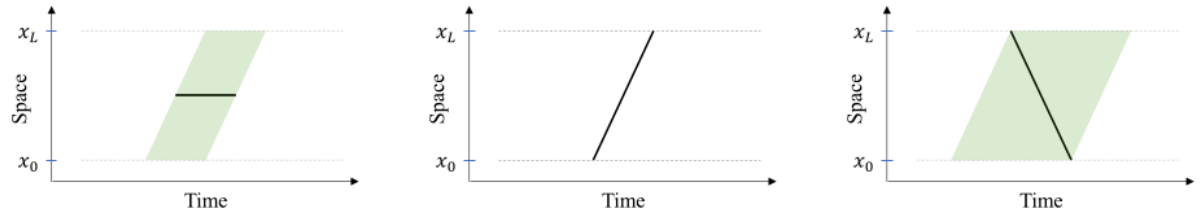
Measuring traffic via vehicles

- Measure the position of some vehicles, **as well as their overtakings**
- Data: in-vehicle sensors
- Combine this for various vehicles, and you can derive the number of vehicles in between.
- This gives densities, speeds and flows



Measuring traffic via vehicles

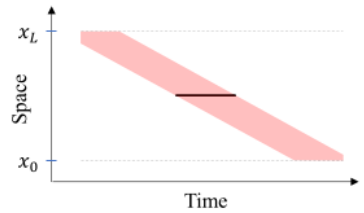
- Potentially combine with loops to have a fixed point zero
- All traffic should be counted
- A moving observer can contribute more than a fixed observer
- Combine with traffic in opposing direction

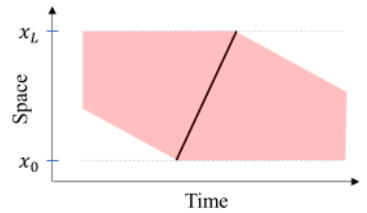


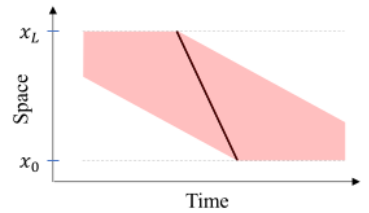
(a) Observer speed = 0 m/s, v^f-wave.

(b) Observer speed = v^f m/s, v^f-wave.

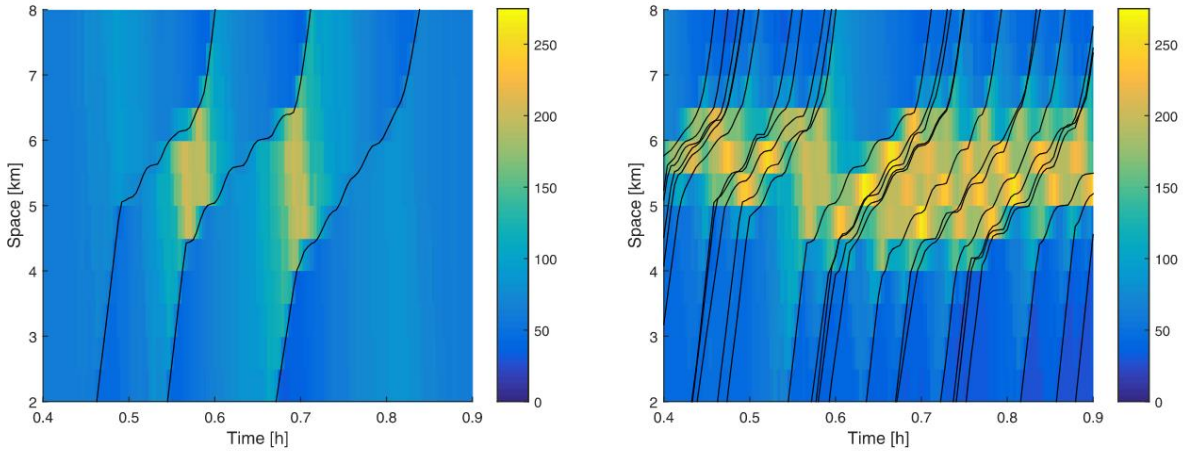
(c) Observer speed = $-v^f$ m/s, v^f-wave.





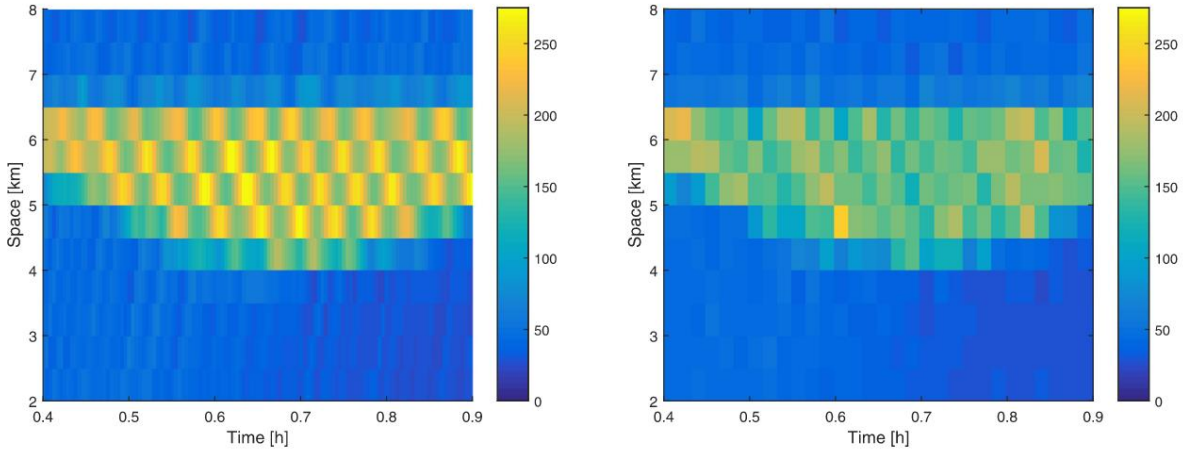


Measuring traffic via vehicles



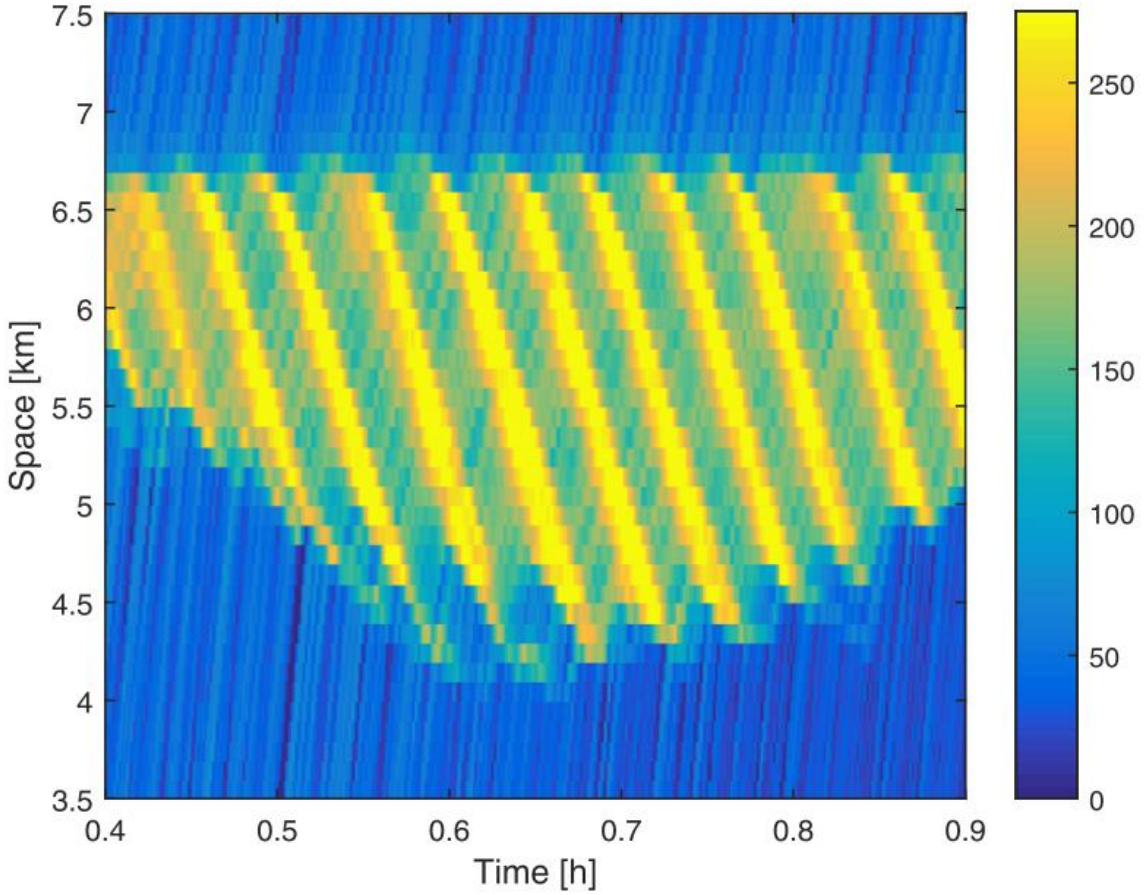
(a) $p = 0.10\%$

(b) $p = 1.0\%$



(c) $p = 10.0\%$

(d) Reference



Conclusions relative flow data

- Sharing “relative flow data” (overtaking times) can yield a very good traffic estimate
- Even with low penetration rates already good traffic state estimation
- Best results if also opposing traffic is also included
- Also works in urban areas (currently no/few loops)

Overall conclusions

- There is plenty of data available (and more will come)
- Many data is data of the same
- We can learn recurrent processes (and we already do quite well)
- For rarer events, physical insights remain necessary

References

- Ehaniz Soldevila, I., Knoop, V.L., Hoogendoorn, S.P., (2021) Transportation Research Records. Car-Following Described by Blending Data Driven and Analytical Models: a Gaussian Process Regression Approach.
- Li, G, Knoop, V.L., and Van Lint, H. (2022) Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach. Transportation Research part C, Vol 138, 103607
- Li, G. PhD thesis, Delft University of Technology
Uncertainty Quantification and Predictability Analysis for Traffic Forecasting at Multiple Scales; Supervisors: Hans van Lint and Victor Knoop
- Lotte Olthof (2022) Lane Change Recognition from Floating Car Data
- Knoop, V.L., Mermygka, M. and Van Lint, J.W.C. (2020) Estimating the urban traffic state with limited traffic data using the MFD, https://arxiv.org/abs/2002.05532