To merge or not to merge?

Data driven approach to behaviour modelling

Chen Zhi Liang
Rulebooks Intern

About me

- 3rd year Undergraduate student at NUS
- Double Degree in Applied Mathematics and Computer Science
- Born in Sichuan, China

Introduction

- What we are interested in

Specifying ‘rules’ for vehicles at uncontrolled intersections, T-junctions

1. How vehicles should behave
2. When should vehicles ‘go’
3. What influences merging vehicles’ decisions - comparing public cars and ego vehicle

Content

1. Problem formulation
2. Data collection and extraction
3. Preliminary analysis
4. Current Implementation
5. Running simulations
6. Future works and Conclusion

Problem formulation

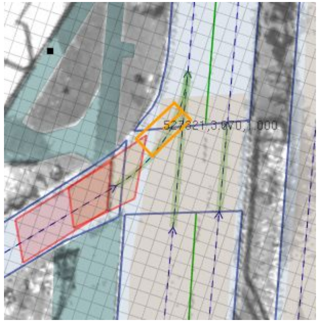
Given a set of human-perceivable features at intersections, find:

1. **Which** features are influential to the merging vehicle
2. The **direction of influence** of these features
3. The **degree of influence** of these features

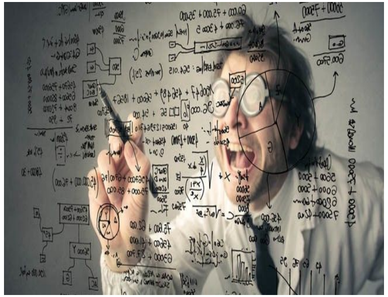
Examples of human-perceivable features: *speed of incoming traffic, distance of incoming traffic, lane of incoming traffic* etc

Data-driven approach

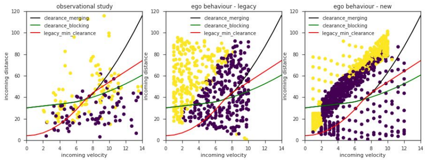
Data collection
from drivelogs



Data analysis



Comparison to current
implementation



Data Collection - let the tedious process begin

- **tracked_object_t**		
	objects[51]	
long	utime	1530771127319525
String	frame_id	global
long	track_id	144648
double	age	1.7997949123382568
short	mostProb_type	0
double	x	407.9387712278563
double	y	1190.7928326477527
double	heading	-1.9994981828479026
double	vx	-0.23456473791127175
double	vy	-0.5132137455043666
double	width	0.5
double	length	0.5
int	ConvexHull_PointNum	5
+ **point3_t[]**	**convex_hull[5]**	
double	hull_height	0.02576998429343913
short	convex_hull_modality	2
boolean	tracking_convergence	true
double	track_status_reside_length	1.6998469829559326
int	function_zone	1
int	is_active	-1
int	type_num	4
- **double[]**	**type_probs[4]**	
double	type_probs[0]	0.9998881704377414
double	type_probs[1]	1.0982956225855019E-4
double	type_probs[2]	1.0E-6
double	type_probs[3]	1.0E-6

Data Collection - let the tedious process begin

t_0

Car 1
ID
X
Y
Velocity
Heading

Car 2
ID
X
Y
Velocity
Heading

Pedestrian
ID
X
Y
Velocity
Heading

Data Collection - let the tedious process begin

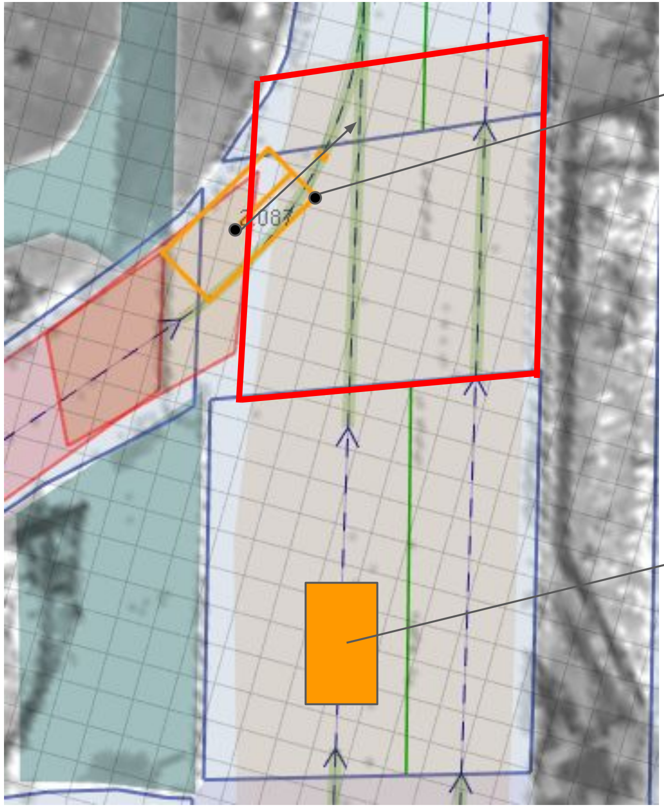
t_1

Car 1
ID
X
Y
Velocity
Heading

Car 2
ID
X
Y
Velocity
Heading

Pedestrian
ID
X
Y
Velocity
Heading

Extracting the data - defining ‘go’ decisions

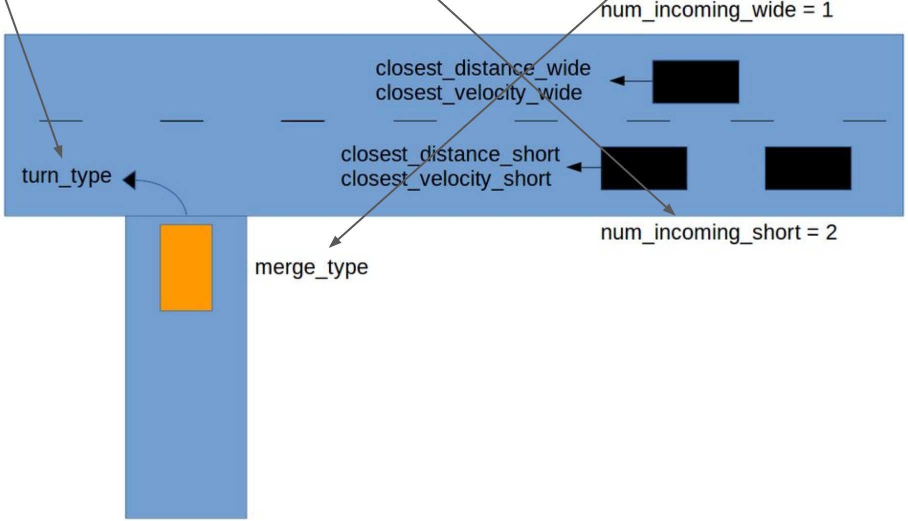


When front footprint of vehicle first crosses into the clearing point

Incoming vehicle lying on valid baseline IDs with trackable velocity and distance

The Data

turn_type	num_incoming_short	num_incoming_wide	closest_distance_short	closest_velocity_short	closest_distance_wide	closest_velocity_wide	merge_type
0	0	1	0.000000	0.000000	17.08004	2.515093	go
0	1	0	2.486818	9.143214	0.00000	0.000000	go
1	0	1	0.000000	0.000000	50.00000	13.945000	go
0	1	0	18.899000	7.382000	0.00000	0.000000	no_go
0	0	1	0.000000	0.000000	16.00000	8.725000	no_go
0	1	0	45.350000	10.635000	0.00000	0.000000	go
0	0	1	0.000000	0.000000	49.66000	13.461400	go
1	0	1	0.000000	0.000000	30.78200	11.761000	no_go
0	1	0	13.614400	5.230000	0.00000	0.000000	no_go
1	1	0	45.688000	13.553000	0.00000	0.000000	go



Data Engineering

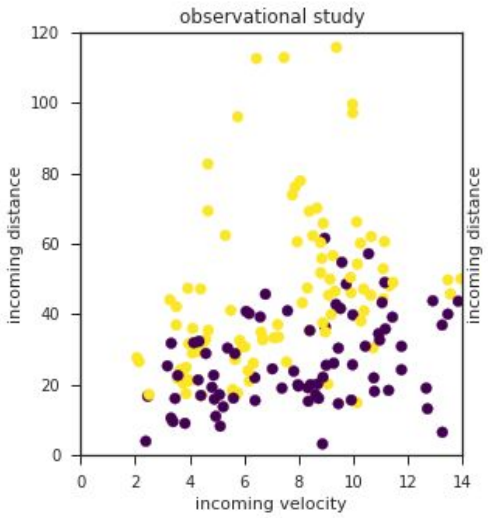
- Data points have varying number of features (incoming traffic on two lanes vs one lane).
    - Solution: ignore data points with two lanes of incoming traffic!
    - Resulting data:

turn_type	lane	closest_dist	closest_vel	num_incoming	merge_type
0	1	46.008214	9.909958	1	1
0	1	62.220547	8.515704	1	1
0	1	65.756337	8.890768	1	1
0	1	47.357017	8.314471	1	1
0	1	112.608609	6.448528	1	1

Preliminary Analysis

Does incoming traffic dynamic even affect ‘go’/’no-go’ decisions?

Seems like it:



Preliminary Analysis - Fitting a baseline Logit model

Cost function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$h_\theta(x^{(i)}) = \frac{1}{1+e^{-\beta_i x^{(i)}}}, \quad y^{(i)} = \text{true labels}$$

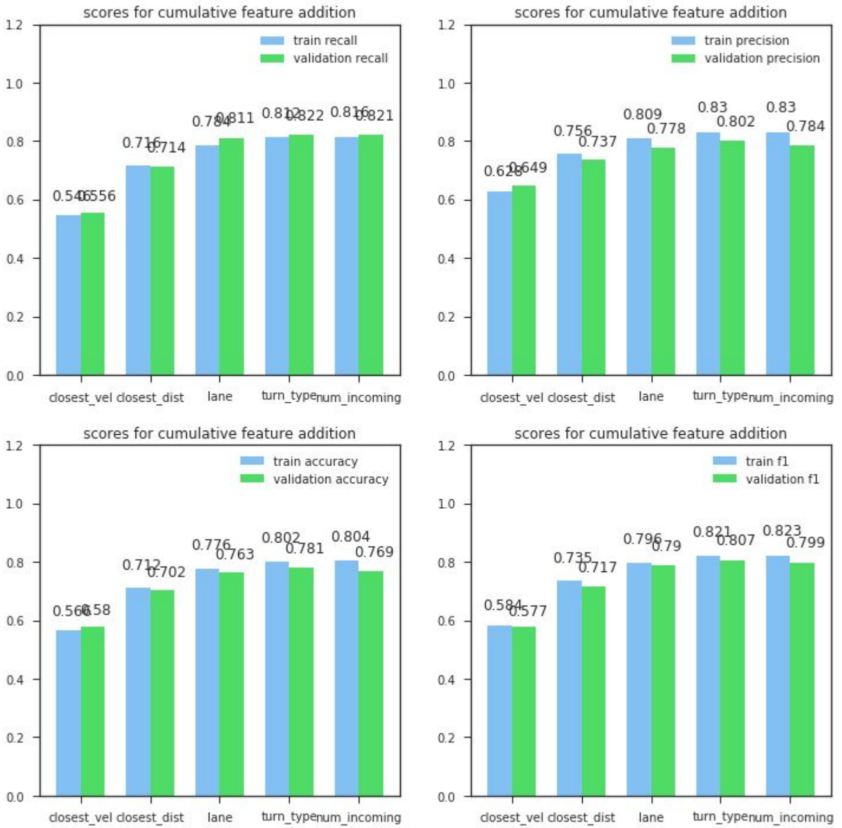
```
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 0.509918689849
            Iterations: 46
            Function evaluations: 49
            Gradient evaluations: 46
                           Logit Regression Results
==============================================================================
Dep. Variable:             merge_type   No. Observations:                  151
Model:                          Logit   Df Residuals:                      146
Method:                           MLE   Df Model:                            4
Date:                Mon, 30 Jul 2018   Pseudo R-squ.:                  0.2518
Time:                        16:27:15   Log-Likelihood:                -76.998
converged:                       True   LL-Null:                       -102.91
                                        LLR p-value:                 1.506e-10
==============================================================================
                  coef    std err          z      P>|z|      [0.025      0.97
------------------------------------------------------------------------------
turn_type      19.7347   7114.071      0.003      0.998   -1.39e+04     1.4e+
lane            1.0662      0.429      2.488      0.013       0.226      1.9
closest_dist    0.0700      0.016      4.438      0.000       0.039      0.1
closest_vel    -0.3155      0.079     -3.993      0.000      -0.470     -0.1
num_incoming   -0.4972      0.597     -0.833      0.405      -1.667      0.6
==============================================================================
```

There are many drawbacks to looking at z-tests and R-squared values solely for model and feature significance. Please refer to http://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/10/lecture-10.pdf for more details.

Preliminary Analysis - Fitting a baseline Logit model

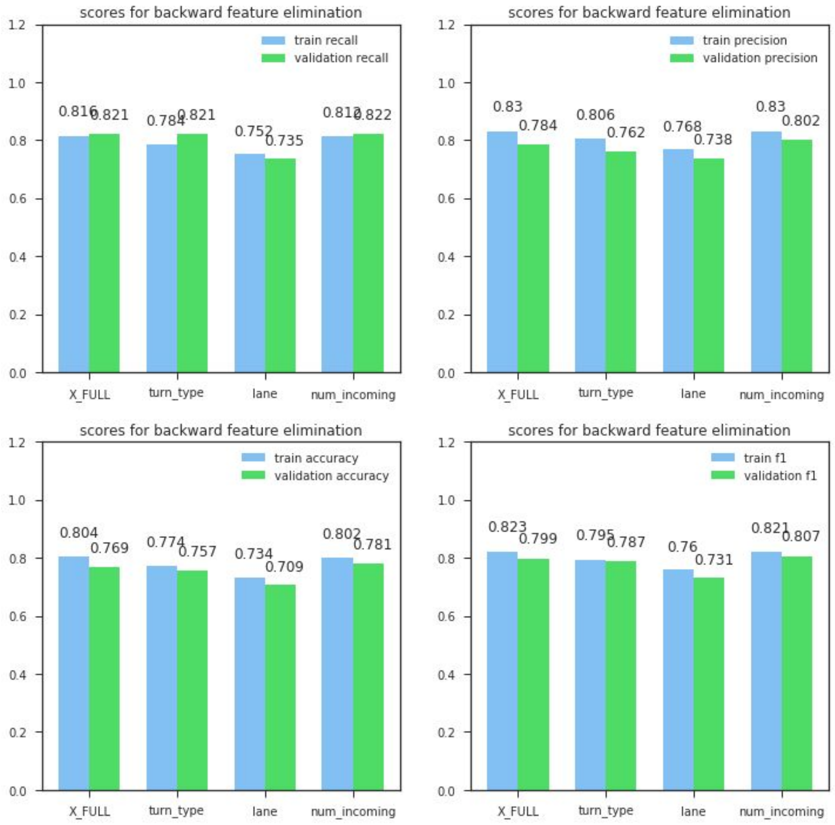
- To confirm the relative importance of features: **use validation dataset**
- Doing it both ways - stepwise feature addition and stepwise feature elimination

Stepwise feature addition



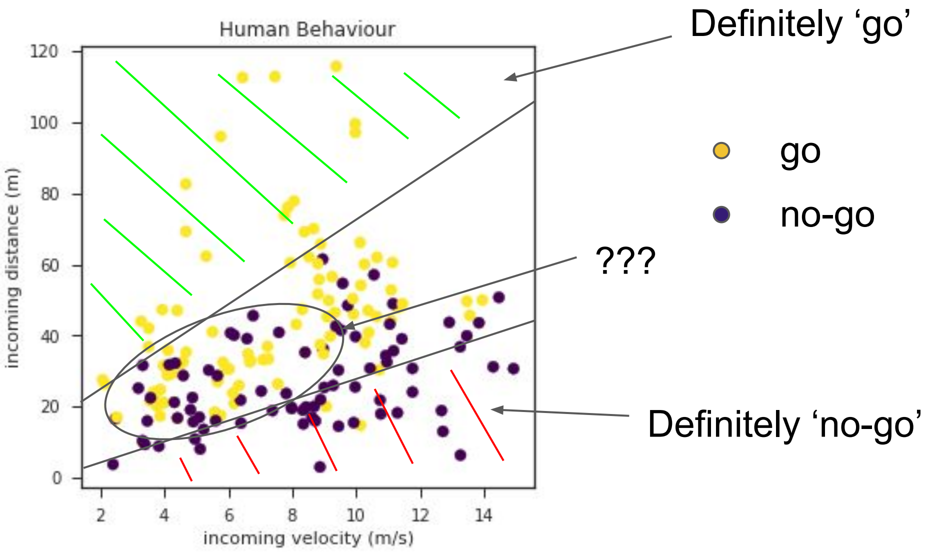
Velocity, ***Distance*** and ***Lane*** increased the performance measures the most.

Stepwise feature elimination

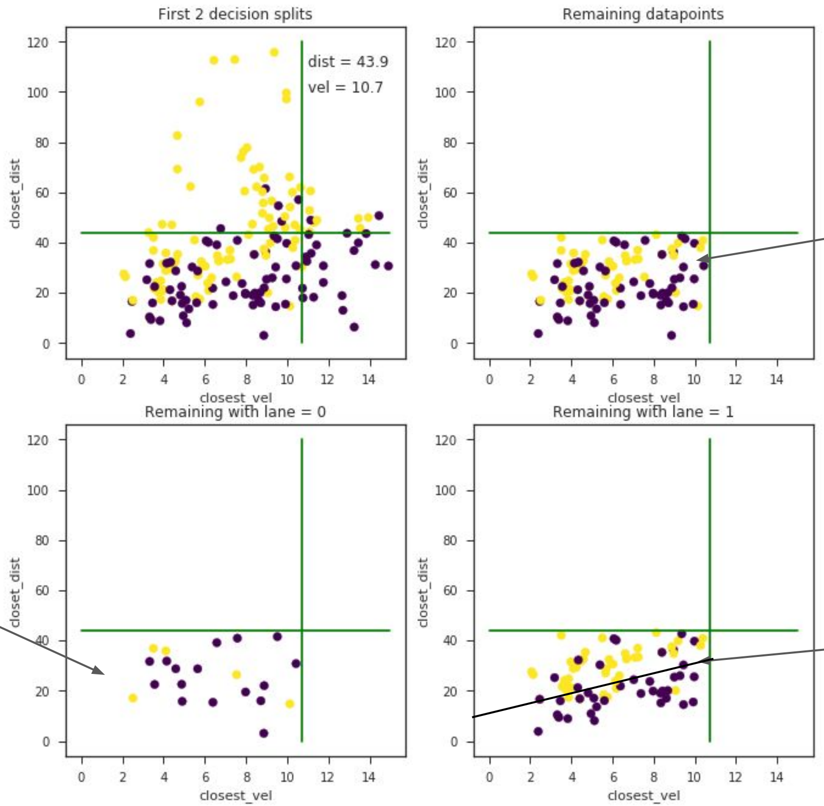


Out of the more subtle features, removing ***Lane*** leads to the greatest dip in performance on validation dataset

Feature importance in feature subspaces



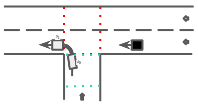
Example of a feature subspace - through visualisation



Decision boundary unclear

go
no-go

Most data points with
lane = 0 have negative labels!

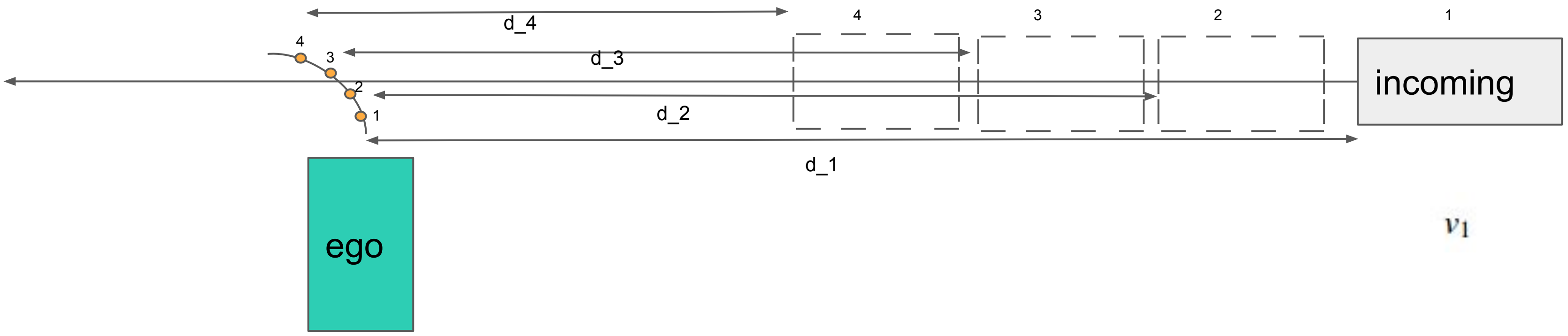


Clearer decision boundary

Current Implementation - In theory

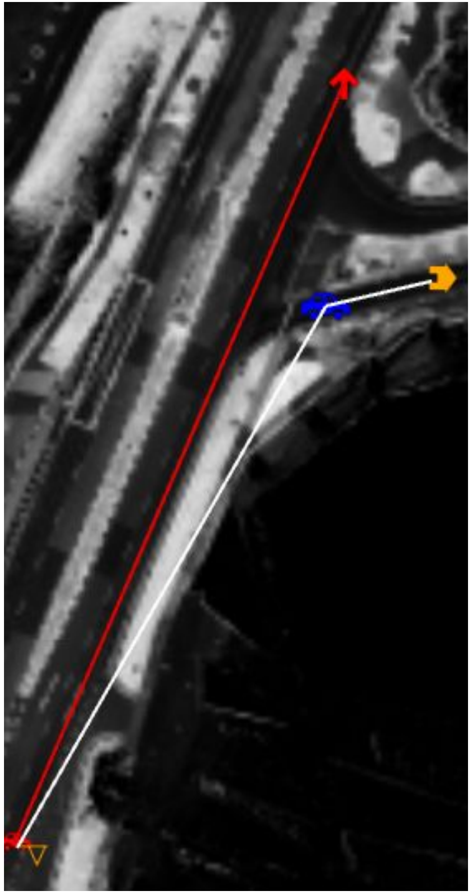
At discrete points on the planned trajectory, the Planner checks if the minimum clearance distance M is satisfied:

$$d_i < M, \text{ where } M = f(v_{\text{incoming at time i}})$$



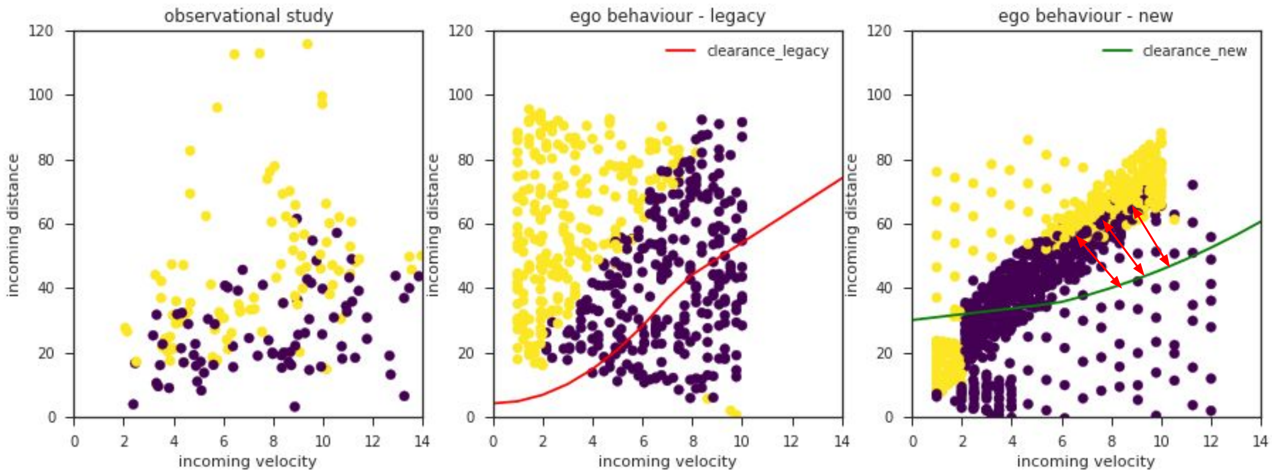
Simulations

- Varies the speed and distance of incoming vehicle (red trajectory)
- Lane = short lane (nearer to t-junction)
- Looks at two planner implementations: *legacy* (~3 months ago) and *new* (current)
- Aims to find the ego's decision boundary



Current Implementation - Visualisation

When maneuver first begins



Bridging man and machine - expected position of cars

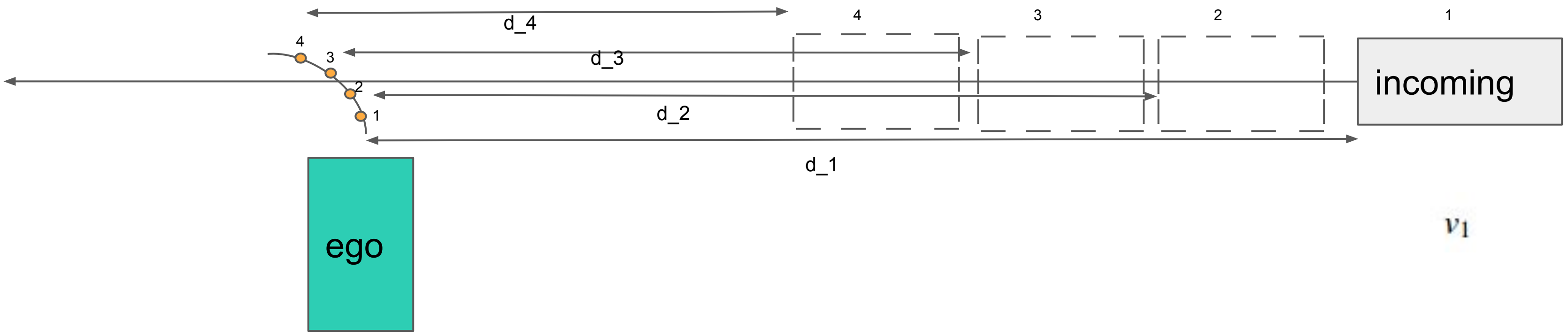
- What causes the disparity between decision boundaries?

    1. Extrapolation of position

Current Implementation - In theory

At discrete points on the planned trajectory, the Planner checks if the minimum clearance distance M is satisfied:

$$d_i < M, \text{ where } M = f(v_{\text{incoming at time i}})$$



Bridging man and machine - expected acceleration of cars

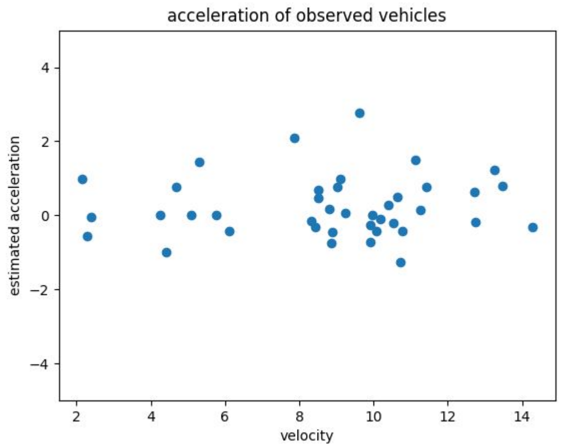
- What causes the disparity between decision boundaries?

2. Expected acceleration

<5 m/s, accel = 1.5 m/s^2
>10 m/s, accel = 0.5 m/s^2
Range 5-10, linear interpolated accel
>speedLimit, accel = 0 m/s^2

Bridging man and machine - acceleration

But do real humans accelerate this way?



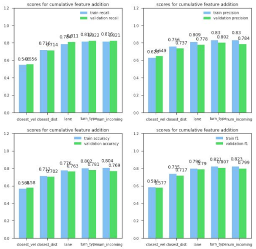
Bridging man and machine - intent prediction **(WIP)**

Changing shoes.

- Aim: Predict whether a vehicle will merge when Ego is part of incoming traffic instead

- We already have a model to try to fit the behaviour of merging vehicles, but performance is not satisfactory (~80% F1)

- Probably need to look at more subtle features near the merging maneuver (e.g how close to stop line? Stationary? Accelerating at stop area?)



Conclusion

1. Human drivers are generally more aggressive than current Ego implementation.

1. Certain subtle features (e.g lane) become important in certain feature subspace. The reason for this is debatable.

1. Extrapolation and expected acceleration plays a large role in current implementation.

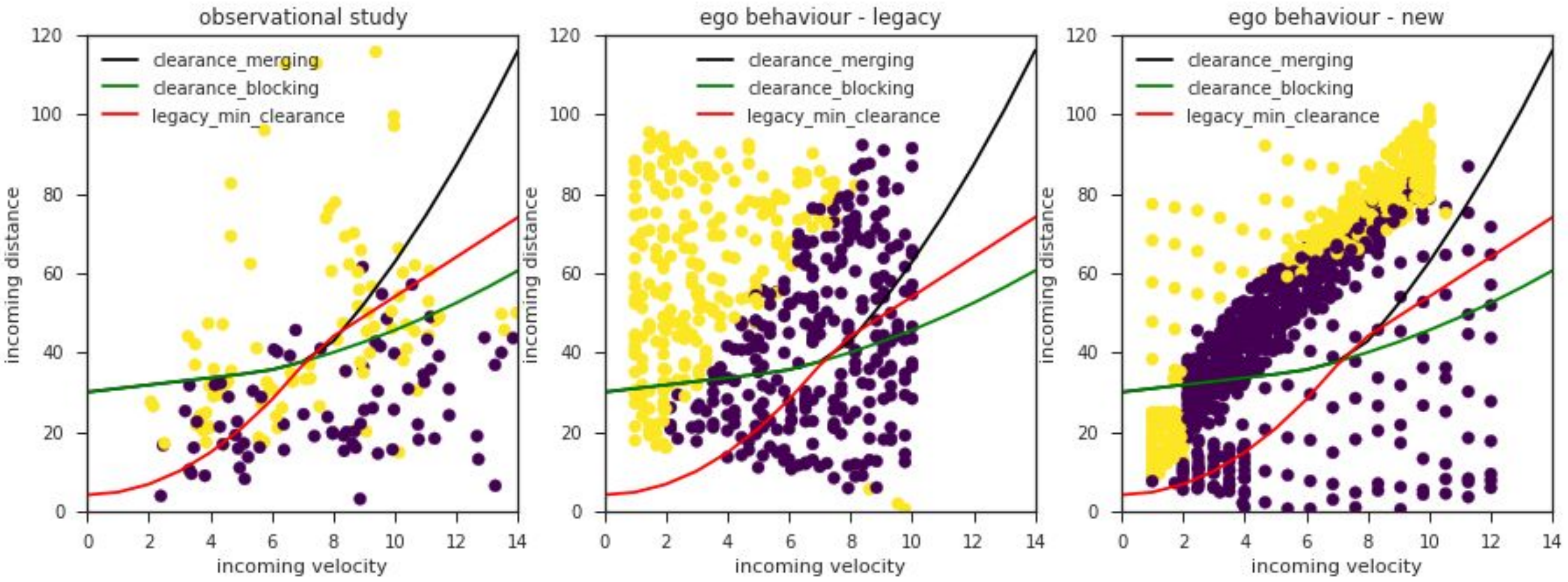
Thank you!

Thanking the following:

1. My mentor, Kostya, for his kind guidance.
2. Team Car, for running the observational studies on public roads.
3. Scott, for providing me with implementation details of the current Ego
4. And everyone else who helped me with issues on Docker, Simulations, Data analysis etc.

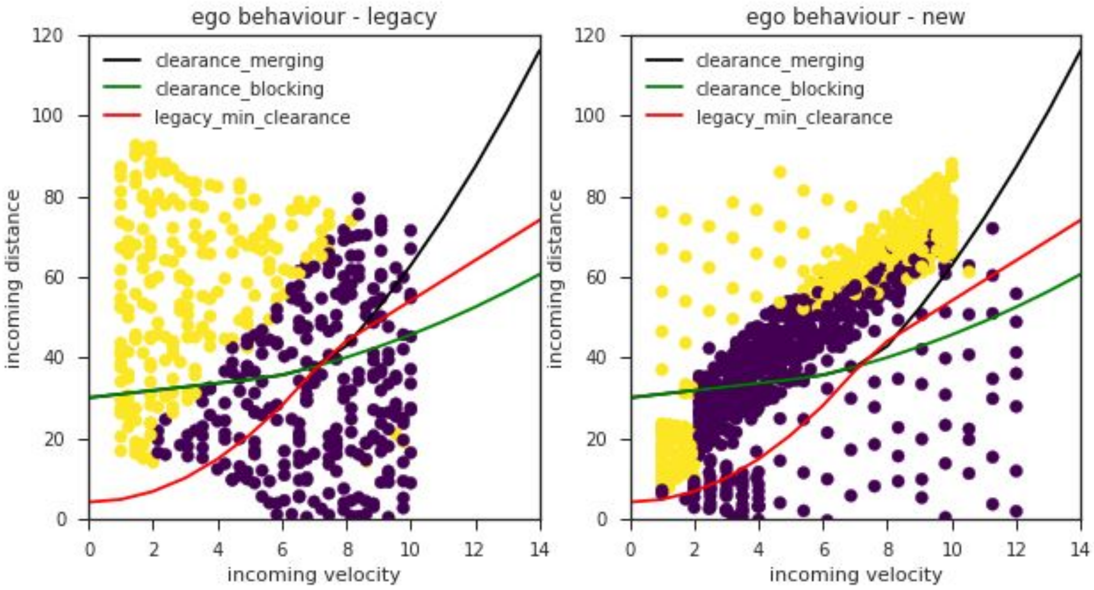
Appendix 1A

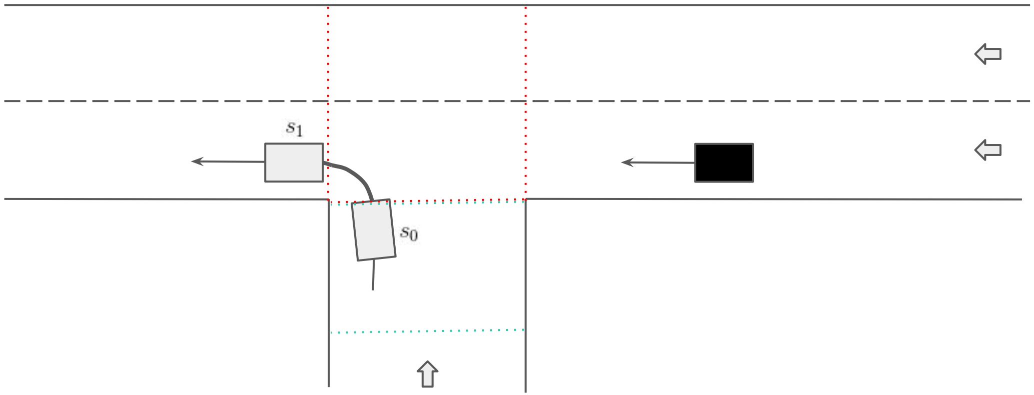
When maneuver first begins



Appendix 1B

When maneuver ends - observe the general downward shift in incoming distance (what we expect)




s_1
s_0