Time Series Forecasting of Vehicle Driving Action

Ajay Kumar Chaddha, Kyungtae Han, Shalini Keshavamurthy (Mentor: Cindy Jiang)

https://drive.google.com/open?id=1gR8FiwrDr_A2x5HxPCojODOUp8sbqPLx

Introduction

- There is always a collision risk between driving vehicles and other shared road users.
- If a driver knows the next action of other vehicle even few seconds earlier, he or she can reduce the collision risk.
- In this project, we transformed time-series prediction as a supervised learning problem to forecast the vehicle's lane change action using its location information and information from its surrounding vehicles.

Challenges

- Limited information (no turn signal available, driving behavior depends on routes, time of day, etc.)
- Variety of human driving behaviors
- Uncertainty of vehicles' future actions

Data

Raw data - Next Generation SIMulation(NGSIM)

- Vehicle trajectory sampled every 0.1 s
- I-80 highway (4:00PM-4:15PM)
- Total number of attributes = 25
- Total number of vehicles = 2037
- Total number of datapoints = 11.8 million

Cleaned data - Reconstructed NGSIM data

- Total number of attributes = 10
- Total number of vehicles = 2037
- Total number of datapoints = 11.8 million

From U.S. Department Of Transportation website

	id	frame	laneid	y	vel	acc	headway	class	followerid	leaderid
0	1	147	2	50.08315	2.84104	1.46320	4.3591	2	-1	-1
1	1	148	2	50.38612	3.02972	1.88680	4.3591	2	-1	-1
2	1	149	2	50.69421	3.08086	0.51134	4.3591	2	-1	-1
3	1	150	2	51.00658	3.12377	0.42912	4.3591	2	-1	-1
4	1	151	2	51.32203	3.15445	0.30681	4.3591	2	-1	-1

Sample data from reconstructed dataset

Attribute	Description	Attribute	Description
id, laneid	Vehicle and lane identifier	y	lateral position in meters
frame	timestamp as integers	vel	velocity of vehicle at given frame
headway	distance between vehicle id and vehicle in front of it	acc	acceleration of vehicle at given frame

Features

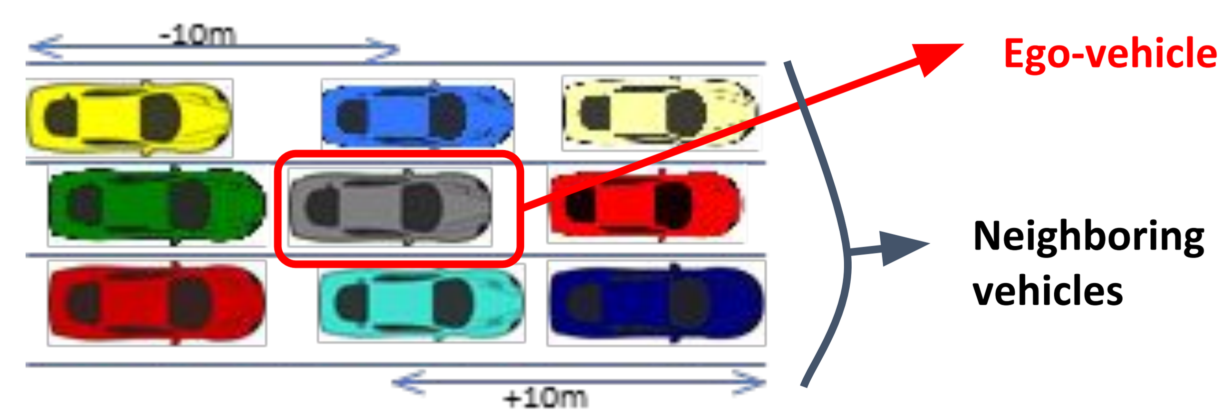
Selected Original Features: Features from each vehicle from each timestep from the reconstructed data

- *Group A* - We chose only 2 features (y, vel)
- *Group B* - We chose only 4 features (y, vel, acc, headway)

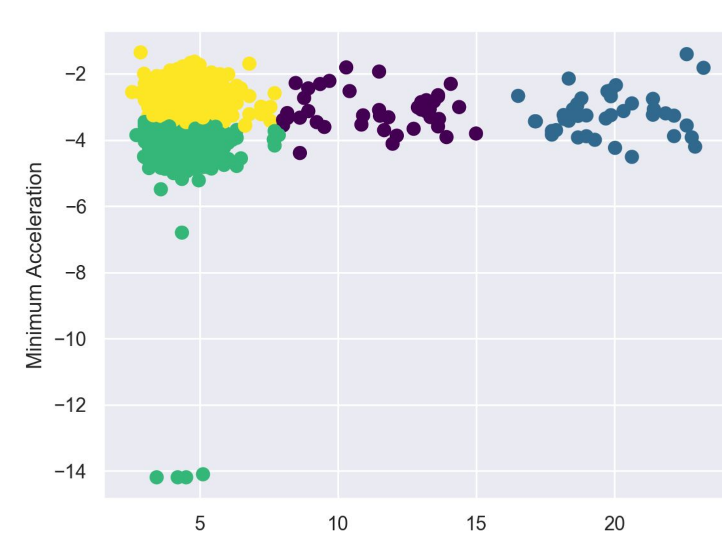
Hyperparameters	Values
Window size (**Win-sz**) with overlap	1, 2, 3, 4, 5 (in seconds)
Window time step	0.1 second
Predicted time step	1 second in the future
Output (class labels)	left lane (-1), in-lane (0), right lane (1)
Input feature size	(Group A or Group B) * 10 * window size

Additional Features:

- *Neighboring vehicles' information* - We chose upto 8 neighboring vehicle information of each vehicle from each timestep from the reconstructed data.



- *Driving type* - Determined by K-means clustering of statistical info. of target vehicle (Average Headway, Minimum Acceleration) from entire trajectory.
  - Hypothesis: Driving type affects lane change action (Aggressive/Careful Driving)
  - Analyzing the scatter plots, K=5 was considered.



Cluster size K = 4

Cluster size K = 5

Results and Discussion

Hyper-parameters	Values
Batch sizes (**B-sz**)	8,16,32,64
Balanced data split	Train (80%), Validation (10%), Test (10%)

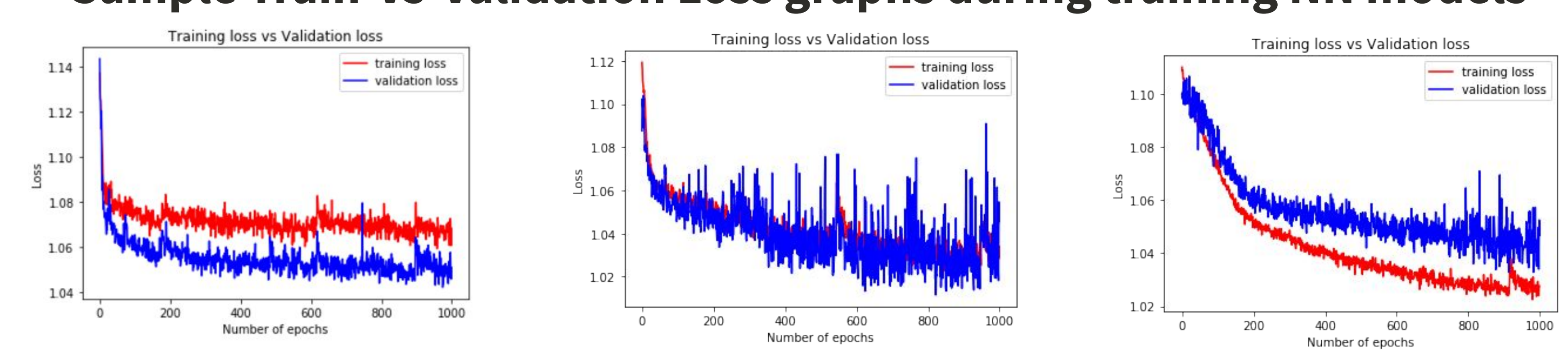
Table 1: Test accuracies for Multi Class Logistic Regression (MCLR)

Win-sz	# Training samples	# Test samples	MCLR - Group A	MCLR - Group B	MCLR - Group B + Driving type	MCLR - Group B + Neighbors
3	2956	329	44.68%	43.36%	41.81%	52.33%

Table 2: Test accuracies for Neural Network (NN)

Win-sz	# Training samples	# Test samples	NN - Group A (B-sz = 8)	NN - Group A (B-sz = 16)	NN - Group B (B-sz = 8)	NN - Group B (B-sz = 16)
1	3545	394	48.47%	48.22%	46.44%	46.70%
2	3204	357	48.45%	49.57%	52.38%	57.98%

Sample Train-vs-Validation Loss graphs during training NN models



- MCLR performs poorly no matter which hyperparameter is modified. Performance improved slightly with additional features.
- To avoid overfitting with too many features, we tried 5-fold cross validation with MCLR on the training data to assess the average accuracies.
- We tried a 2-layer NN to model the prediction. Cross entropy loss decreased as training progressed. We tuned hyperparameters to address underfitting and overfitting but we could not achieve significant increase in accuracy. The best accuracy we get is **58.26%** for **B-sz** = 64, #hidden nodes = 20, **Win-sz** = 2 seconds.

Future Work

- Reduce underfitting and overfitting of neural network using different methods - Increase samples, add dropouts, reduce features, etc.
- Add driving-type and neighboring vehicle data as features to the neural network and conduct more experiments.