Modeling Interactions Between Autonomous Vehicles and Pedestrians

Code Readme

January 2018 - May 2018

Mubarak Abdul Kader | Salma Benslimane | Dayi Wang | Dan Xu | Moyu Li
Advisors - Dr. Gabriel Gomes, ME Department at UC Berkeley | Dr. Ching-Yao Chan, Research Engineering, California PATH, UC Berkeley
University of California, Berkeley

Python Code

Relevant Links

1. **Github link to our code -** https://github.com/UC-Berkeley-MEng-Capstone/Capstone_Code/tree/master
2. **Traffic Intelligence for trajectory extraction from videos -** https://bitbucket.org/Nicolas/trafficintelligence/wiki/Home

Files on GitHub

1. Code Readme file that has information about all of the files on GitHub and brief description of the phases of the capstone project
2. Capstone Report with detailed information about the project
3. Capstone Project Presentation
4. Final Code > Random Forest
    a. Random Forest.ipynb has the code for training and testing a Random Forest model
    b. Input1 and Input2 are pickled training data - these input files have different features, kindly refer to the above IPython notebook for more information of how these are created
    c. Get_data.ipynb has the code to preprocess the data
    d. SQL folder has all .sqlite files which have the trajectory of objects in the videos passed through it
    e. SQL > Interactions folder has the database of interactions that were formed from the dataset with individual objects - this was created using dataProcessing.py in Capstone_code which is the main directory above FInal Code
5. Logistic_reg.ipynb has the code to train a logistic regression model on a smaller dataset with lesser features, kindly refer to the notebook for more information

6. dataProcessing.py has code to identify the interactions from datasets having trajectories of individual objects extracted from videos using Traffic Intelligence open source code by Nicholas
7. mub_classify.py has the code to classify objects as pedestrians and vehicles based on the trajectory data extracted from the Traffic Intelligence open source code

Phases of the project

1. Trajectory Extraction
2. Feature Engineering
3. Machine Learning

Trajectory Extraction

Code was written using OpenCV in C++ to extract trajectory of vehicles and pedestrians from videos of real world interactions between pedestrians and human-driven vehicles

Feature Engineering

Code was written using Python to pair objects that are interacting with each other and the trajectory data of these objects were used to compute velocities, relative distances, angles, derivative of angles, time it would take for the vehicle to reach the pedestrian etc. The purpose here is to create features that have predictive power to identify whether or not the vehicle would brake (response variable)

Machine Learning

Code was written using Python to build Logistic Regression and Random Forest classification models. The dataset of 8000 interaction samples were split based on 80:20 train-test split and the model was trained as well as cross-validated on the training dataset. The trained model was then tested on unseen data and the results have been listed on the final report.

Conclusion

Model struggles to predict class 1, possible reasons could include imbalanced training dataset, small number of samples, and absence of sanity checks during feature engineering. Kindly refer to the report and code database for more information