CS5340 Project Proposal

Li Xiaochen
A0231442U
e0703474@u.nus.edu

Zhi Jun
A0243717A
e0833564@u.nus.edu

Shi Yuang
A0229927N
e0686126@u.nus.edu

Wang Binluo
A0224877R
e0575511@u.nus.edu

Guo Ziyi
A0248353B
e0925505@u.nus.edu

Abstract

To increase the comforts and safety of the driver and the passengers, and avoid accidents, human activity detection and analysis attracts increasing number of researchers in self-driving domain. In this project, we aim to build a pedestrian behavior probabilistic model to predict whether a pedestrian crosses the road or not. We propose two algorithms, Hidden Markov Model (HMM) and Kalman Filter, for trajectory measurement and predictions. We plan to conduct experiments on two large scale datasets: TrajNet++ Dataset and L-CAS 3DOF Dataset.

1 Motivation

In the global background, human activity detection and analysis is used for wide-range applications, including surveillance systems, environment understanding, robotics application, content-based image retrieval, video annotation, assisted living, intelligent vehicles, and advanced user interfaces [1].

Pedestrian detection is an indispensable part of self-driving cars and vision-based driver-assisted systems. To avoid collisions and guarantee the passengers' and drivers' experience, when the probability of a pedestrian crossing the road is small enough, the car will not stop and give way to the pedestrian. To this end, our system aims to ensure the least amount of stopping of self-driving cars on the condition of safety, reduce the times of passengers waiting and improve passengers' experience.

2 Methodology

2.1 Problem Statement

Our objective is to forecast the future trajectory of the pedestrian present in a scene, and then detect the pedestrian crossing the street. Figure 1 shows the scenario of our task. The model takes as input the trajectory of the person in a scene denoted by $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$. The position and velocity of the pedestrian at time-step t is denoted by $\mathbf{x}^t = (x^t, y^t)$ and $\mathbf{v}^t = (v_x^t, v_y^t)$. Our task is to forecast the corresponding future trajectory $\mathbf{Y}$, and then classify whether the pedestrian intends to cross the street. Specifically, we receive the positions of the pedestrian at time-steps $t = 1, \ldots, T_{obs}$ and want to forecast the future positions from time-steps $t = T_{obs} + 1$ to T_{pred}. We denote our predictions using $\hat{\mathbf{Y}}$. According to the predictions, we then detect the pedestrian as a pedestrian crossing road or not with probability p_c.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

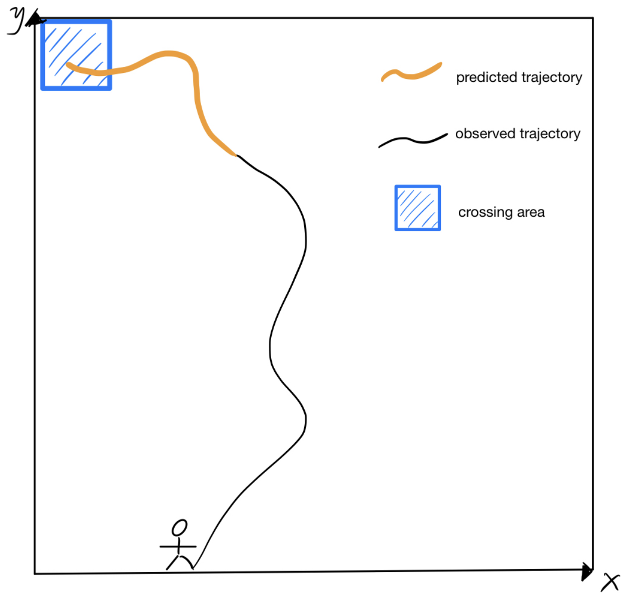

Figure 1: Example of the proposed problem. The goal is to predict the trajectory of the pedestrian given the current observed trajectory and detect whether the pedestrian intends to cross the road. The pedestrian will be regarded as a pedestrian crossing road when he/she reaches the crossing area.

2.2 Trajectory Prediction

We consider two methods for trajectory prediction: (1) Hidden Markov Models (HMM) [2], and (2) Kalman Filter (KF) [3].

Hidden Markov Models: HMM is a Markov model with hidden state that can be used to track processes with hidden state. In this case, the process is the observed pedestrian's trajectory, and the hidden state is the sequence of pedestrian's actions used to follow a trajectory. The main use of the model is to predict the next action which the pedestrian will do, and then compute the position of the pedestrian based on the predicted action. Specifically, HMM in our task can be regarded as a tuple $< S, A, O, T, Z, \pi >$. S is a set of states, A is a set of actions, and T is the transition function $T : S \times A \times S \rightarrow \Re$, where $T(si, a, sj) = p(si|sj, a)$, which is the transition probability of transitioning to state s_i given that the system is in state s_j and action a is executed. π is the initial state distribution. In addition, O is a set of observations and Z is the observation function $Z : O \times S \times A \rightarrow \Re$, where $Z(o, s, a) = p(o|s, a)$, which is the emission probability of receiving observation o given that the system ends up in state s after executing action a. It is notable that considering the complexity of human activity in the wild, we assume that the pedestrian will perform one action at every time-step, where we confine the action within $\{Up, Left, Right, Stop\}$.

Moreover, the researchers in [4] built on top of a double layer hidden states sequences HMM and proposed a novel algorithm for vehicle trajectory prediction. Inspired by this work, we consider dividing the prediction phase into two parts, that is, training part and prediction part. During the prior stage, we will collect historical trajectory information and then construct the HMM, while in the second prediction stage, just driven vehicle trajectories are served as input and determine the most likely double layers hidden sequences with the adoption of Viterbi algorithm.

Kalman Filter: There are two major limitations of HMM in our task. First, we need to discretize the problem into finite sets of states and actions for HMM, which results in discretized trajectories. Such trajectories cannot well fit the trajectories in the real world. Second, we ignore the noise from data sampling and pre-processing, which may leads to bias in learning and prediction phase. Therefor, we consider KF as a comparable method for trajectory prediction. KF is regarded as analogous to HMM, with the difference that the hidden state variables have values in a continuous space as opposed to a discrete state space as for the hidden Markov model. Also, KF can filter noise effectively, thus predicting smoother trajectory compared to the prediction of HMM.

2.3 Crossing Pedestrian Detection

We use the kalman filter to predict the possible successive locations of pedestrians within a 5 seconds window, and add the prediction error to describe the distribution of future position. Since the kalman filter is a linear equation with a Gaussian distribution, the errors on x and y can be accumulated. Thus, for each predicted point along the trajectory in the future, by using the accumulated errors about x, y, we can draw a set of ellipses with different radius. Thus we can see that the area of the ellipse becomes larger and larger as the prediction time grows. We use the overlapping area of the ellipse and the "intersection" area, denoted by a square, to determine the probability that a pedestrian will cross the road in 5 seconds. If the predicted ellipse falls exactly into our specified square, then we can assume that the probability of the pedestrian crossing the road at that moment is 100%. If the ellipse partially intersects the square, then we indicate the probability of the pedestrian crossing the road at that moment based on the proportion of the area where the ellipse intersects the intersection.

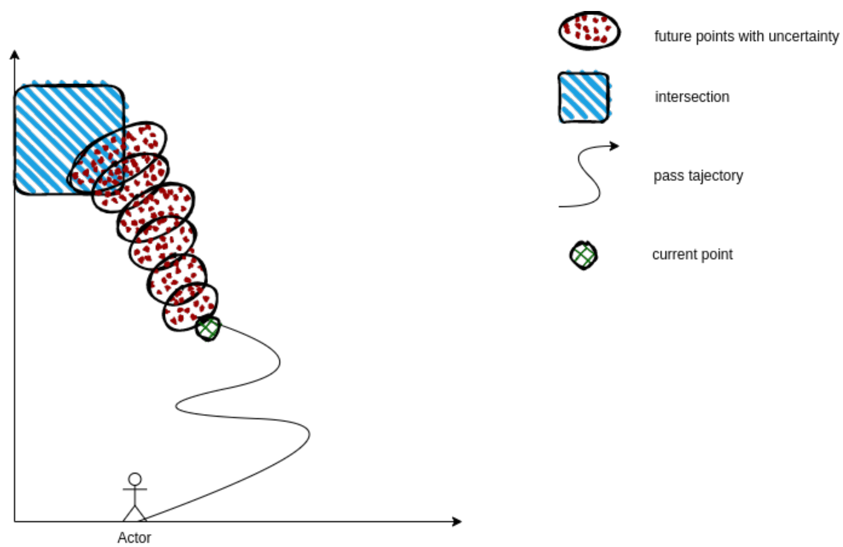

Figure 2: Mechanism for crosswalk probability

3 Experiment Setting

3.1 Dataset

We plan to use two existing datasets of pedestrian trajectory for evaluation, they are (1) TrajNet++ Dataset, and (2) L-CAS 3DOF Pedestrian Trajectory Dataset.

TrajNet++ Dataset: TrajNet++ is a large scale interaction-centric trajectory-based benchmark. The TrajNet++ dataset provides not only proper sampling of trajectories but also a unified extensive evaluation system to test the gathered methods for a fair comparison. The dataset files contain two different data representations[1]:

L-CAS 3DOF Pedestrian Trajectory Dataset: The L-CAS dataset was collected by a Velodyne VLP-16 3D LiDAR.[2]. The researcher use the sensor reference frame to record the data, and made all human detections and tracks transformed to the world frame. The dataset includes 935 pedestrian trajectories, which were extracted from the first 19 minutes of recorded data.

[1]https://github.com/vita-epfl/trajnetplusplusdataset/

[2]https://lcas.lincoln.ac.uk/wp/3dof-pedestrian-trajectory-dataset/

3.2 Evaluation Metric

We draw the receiver operating characteristic (ROC) curve by plotting true positive rate (TPR) the classifiers positive rate (FPR) as functions of the threshold t for the decision boundary, for classifier $p(y = 1|x, w) \geq t$, to analyze the best decision boundary. The larger the area under the curve (AUC) is, the better the performance of the classifier is. If there are lots of negative samples, precision is preferred over FPR.

In addition, we make use of N-fold cross-validation if the size of the training set is very small. In order to ensure the random, uniform, and unbiased sampling of the validation set, the training set is divided into N blocks, and each block is used as the validation set in turn. Eventually, we calculate the average accuracy of the validation set for each model under N times of training and select the model with the highest average accuracy. Any resulting model's function in the N times of training results is then trained through the validation set to obtain the final model.

4 Work Plan

Task/Week	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Preparation	X						
EDA		X					
Model construction		X	X	X	X		
Evaluation					X	X	
Finish report						X	X
Submission							X

Figure 3: Work Plan Flow

References

[1] Joko Hariyono and Kang-Hyun Jo. Detection of pedestrian crossing road. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4585–4588. IEEE, 2015.

[2] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

[3] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[4] Ning Ye, Yingya Zhang, Ruchuan Wang, and Reza Malekian. Vehicle trajectory prediction based on hidden markov model. 2016.