E205 - Lab 2

OneD Filters

Due: 1:00pm Monday September 21, 2020

1. Download the Data

In lab2, we will use data from two sources. First, we will use yaw angle data collected by a BNO055 IMU. Fig. 1 illustrates the approximate path taken by a "bot" equipped with the IMU. Download the data files 2020-02-08_09;22;47.csv, 2020-02-08_08;34;45.csv, and 2020-02-08_08;52;01.csv. The data includes the [x y ϑ] logged at 1/13 second time steps. Be aware that the IMU yaw angle is 0 degrees East, 270 degrees North.

Figure 1: Approximate path followed by a bot equipped with a BNO055 IMU to log yaw data.

Second, we will make use of NuScenes's open source autonomous vehicle data that can be found at nuscenes.org. We have extracted and adapted data from this data set to make it usable for this lab. Download the data file E205_Lab2_NuScenesData.xlsx. The data includes the [x y ϑ] for different vehicles logged at 0.5 second time steps. Note that ϑ represents the vehicles' yaw angle.

Figure 2: Screenshot of the image processed video that accompanies the NuScene data set.

This lab is meant to be done in python (version 3+), although it is possible to do it in Matlab. Executing the lab in python will allow you to progress towards completing a final project that works in real time. If python is being used, the "pandas" library is helpful for importing Excel data.

2. OneD Kalman Filter For Filtering Yaw

In this part of the lab, you will filter raw vehicle yaw data to estimate a vehicle's yaw angle using a 1D KF.

a) Model the raw yaw data measurements as normally distributed and calculate the variance associated with this measurement. Use data from the file 2020-02-08_09;22;47.csv where the IMU and GPS were held stationary.

b) Write code in python or Matlab that implements the correction step of a 1D Kalman Filter, and applies it to the raw data of the IMU. A prediction step is not necessary for this lab. This code should output the estimated yaw angle and the associated covariance. Apply this code to the data in the files 2020-02-08_08;34;45.csv and 2020-02-08_08;52;01.csv. *Note*: if coding in python, one can use the stubs code provided in the file run_1D_KF_student.py where students can add their code to implement the filter.

c) On a single figure, plot the raw data, estimated data, and estimated data +/- 2σ bounds. Use a second figure that zooms in for only 10 seconds of data to illustrate if the raw and filtered data lie within the 2σ bounds.

2. Bayes Filter For Estimating Vehicle is stopped

In this part of the lab, we will make progress in answering the question: Parked or Not Parked? This is an import capability for autonomous vehicles. Here we will only be asking the question: Stopped or Not Stopped, although extra work could be done to modify the algorithm for the more difficult question.

a) Examine the Excel sheet that has [x y ϑ] data for ego vehicle and 6 other vehicles. At the same time, watch the video of the attached scene and try to spot each vehicle in the video. The vehicles are described in a table in the Excel sheet, (i.e. far right columns).

b) Using vehicle 4, which is parked the entire time, create a histogram of vehicle speed. You can use the [x y] data from these vehicles to calculate vehicle speed s at each time step. Using this histogram, create a PDF that represents the conditional probability $p(s_i|x_i = stopped)$, where s_i and x_i and are vehicle i's speed and stop state respectively.

c) Using vehicles 2, 3 and 5, which are moving, create a histogram of vehicle speed. Create a PDF that represents the conditional probability $p(s_i|x_i = not\ stopped)$.

d) Implement a Bayes filter that outputs $p(x_i = stopped|s_i)$, the probability that vehicle i is stopped, at each time step. For the prediction step, you can use the following probabilities:
   a. $p(x_{i,t} = stopped|x_{i,t-1} = stopped) = 0.6$
   b. $p(x_{i,t} = not\ stopped|x_{i,t-1} = stopped) = 0.4$
   c. $p(x_{i,t} = stopped|x_{i,t-1} = not\ stopped) = 0.25$
   d. $p(x_{i,t} = not\ stopped|x_{i,t-1} = not\ stopped) = 0.75$

e) Plot the output of the Bayes filter as a function of time for all vehicles.

f) Describe vehicle #6 using at least a color descriptor and a vehicle type (small car, mini-SUV, SUV, flatbed truck, semitruck, van, minivan).

3. Applying Bayes Filter to Police Violence Data

Using the police violence data from lab 1, and some additional US census data, we will compare the American population data to some particular queries of police data using 1 step of a Bayes filter. This could be done in many ways, but one suggested implementation is given below.

a) First, determine the probability of being white, black, Hispanic, or Asian in America. Some simple web searches should be helpful.

b) Create a prediction step of a filter that predicts the race of a police killing victim based on the labeling of victim being unarmed, armed, or unclear. For the data set in hand, we will equate the labels “allegedly armed” with “armed”. To start, use the data from lab 1, to create the following conditional probabilities:
   a. p(race=white | person_was_killed_by_police=true, victim_was_armed=true)
   b. p(race=white | person_was_killed_by_police=true, victim_was_armed=false)
   c. p(race=white | person_was_killed_by_police=true, victim_was_armed=unclear)
   d. p(race=black | person_was_killed_by_police=true, victim_was_armed=true)
   e. p(race=black | person_was_killed_by_police=true, victim_was_armed=false)
   f. p(race=black | person_was_killed_by_police=true, victim_was_armed=unclear)
   g. p(race=hispanic | person_was_killed_by_police=true, victim_was_armed=true)
   h. p(race=hispanic | person_was_killed_by_police=true, victim_was_armed=false)
   i. p(race=hispanic | person_was_killed_by_police=true, victim_was_armed=unclear)
   j. p(race=asian | person_was_killed_by_police=true, victim_was_armed=true)
   k. p(race=asian | person_was_killed_by_police=true, victim_was_armed=false)
   l. p(race=asian | person_was_killed_by_police=true, victim_was_armed=unclear)

c) Calculate the predicted probabilities for different races, for the two specific cases:
   a. p(victim_was_armed=true) = 0.8, p(victim_was_armed=unclear) = 0.2
   b. p(victim_was_armed=false) = 0.8, p(victim_was_armed=unclear) = 0.2

   You may want to take advantage of the probability rule:

   p(race=i | pwkby=true) =
   $\sum_j$ p(race=i | pwkby=true, victim_was_armed=j) p(victim_was_armed=j)

d) Create a correction step that predicts the race of a police killing given the victim was < 20 years old. To start, use the data from lab 1, to create the following conditional probabilities:
   a. p(age<20 | person_was_killed_by_police=true, race=white)
   b. p(age<20 | person_was_killed_by_police=true, race=black)
   c. p(age<20 | person_was_killed_by_police=true, race=hispanic)
   d. p(age<20 | person_was_killed_by_police=true, race=asian)

   Then, to fully implement the filter, recall Bayes rule which can be applied here as:

   p(race=i | pwkby=true, age<20) = *n* p(age<20 | pwkby=true, race = i) p(race = i)

4. Deliverables

A lab report should be submitted by 1:00pm Monday Sept 21st on Sakai under Lab 2 of *Assignments*. You can use the IEEE conference paper template (similar to E80), but the report should be no longer than 3 pages. The report should include histogram and other plots listed above. A short few-sentence justification can be used for any modeling decisions. For both filters, concisely describe the filter algorithm, which includes important equations or update steps.