Practical Considerations for Testing the Daimler Use Case

The application scenario is early recognition of lane change manoeuvre. Below is a very simplistic explanation of how an algorithm can be tested.

1 Early recognition of lane change manoeuvre

A *lane change manoeuvre* is either a EGO cut in, EGO cut out, object cut in and object cut out. Moreover, a *no lane change manoeuvre* is either a object follow or a lane follow. In the testing regime we have a binary classification problem: lane change or no lane change and the subdivision into the other categories is not a concern regarding testing this particular application scenario.

Daimler has provided us with a number of time series, some including a lane change and some involves only lane follow. Some of the time series for lane change are about 15 seconds long, involving 10 seconds of data before the lane change and 5 seconds after the lane change. The rest of the time series for lane change 5 seconds long, where the last data point is always a lane change. In the further analysis we can assume that all time series are 5 seconds long ending with a lane change, either because they were like this as we got them from Daimler or the Amidst team chopped them. Regarding data for no lane change, all time series are 5 seconds long.

The resolution is fixed, but is dependent on which camera is used (typically in the range of 40 - 60 ms). Missing values are common and about every second measurement is missing (meaning that the average resolution is approximately 100 ms). There are 93 time series of lane follow and 148 time series of lane change. There are no overlap between any of these time series, and they can be seen as independent of each other.

The time series are denoted $\mathbf{s_i}$, where each vector of measurement at timestep $j \in \{1, 2, ...n\}$ is denoted $\boldsymbol{s_{i,j}}$. This vector includes physical quantities such as distances, velocities and accelerations. At every time step j, the static and the dynamic Bayesian networks calculates the values $P(\text{Lane Change}_j \mid \boldsymbol{s_{i,j}})$ and
$P(\text{Lane Change}_j \mid \{\boldsymbol{s_{i,0}}, \boldsymbol{s_{i,1}}, ...\boldsymbol{s_{i,j}}\})$, respectively. In the rest of this document, we will refer to testing the static Bayesian network, because the extension to the dynamic is straightforward. Moreover, testing network fragments can also follow the same testing framework and this is not explicitly stated each time.

It is essential that both models (static and dynamic) predict the probability of an actual lane change at the *present moment* j. As a though experiment it would have been interesting to estimate the probability of lane change as a function of a

time into the future. For the static network this would be the probabilities $f(k) = P(\text{Lane Change}_{j+k} \mid \boldsymbol{s_{i,j}})$, where $k \geq 0$. Notice that $f(k)$ is not a probability density as it does not integrate to one. However, the probability of having a lane change between the 10th and 20th time step ahead is basically $\sum_{10}^{20} f(k)$ (provided that only one lane change can happen in this time frame).

However, this analysis is not part of the model. In fact the ability of detecting a lane change some time before it happens is taken into account by the fact that $P(\text{Lane Change}_j \mid \boldsymbol{s_{i,j}})$ tends to increase before the actual lane change.

The reason for this model choice is that there are strong limitations related to the cpu-time and memory usage. (Can you elaborate a bit here Anders?)

Calculation of area under ROC curve

In the use case scenario, two situations are mentioned; detection of lane change at least 1 second before it happens and detecting the lane change 2 seconds before it happens. To be a bit more exact, we have modified this to detect lane change in interval one (i.e. between 1 and 4 seconds before the actual lane change) and interval two (i. e. between 2 and 5 seconds before the actual lane change).

Let T be a fixed time window (for instance 1 and 4 seconds before the end of the time series) and also let $W_i = W_i(T)$ be a set of all time steps that are inside this time window on time series $\mathbf{s_i}$, then we estimate the class probability of a time series by

$$P(\text{Lane Change}_T \mid \mathbf{s_i}) = \underset{k \in W_i(T)}{\operatorname{argmax}} P(\text{Lane Change}_k \mid \boldsymbol{s_{i,k}}) \tag{1}$$

Any classification rule is basically assigning a positive whenever $P(\text{Lane Change}_T \mid \mathbf{s_i})$ is larger than a threshold C. Otherwise, the classification rule assigns a negative.

The ROC curve is found by plotting the rate of false positives versus the rate of false negatives for any setting of C. The AUC measure is the area under the ROC curve and can be found by taking the definite integral from 0 to 1.

There is also a simpler way, which involves sorting all $P(\text{Lane Change}_T \mid \mathbf{s_i})$ and picking exactly one threshold between each consecutive value in this sorted list. AUC is basically the average of the true positive rates for all these thresholds. This will be explained in more detail in the general methodology chapter.

Clearly, AUC is dependent on which time window that is chosen. In terms of visualising the results, it is of interest to plot AUC as a function of the end time in the time window T. Here, it would make sense to keep the difference between the start and the end time fixed at for instance 3 seconds. Showing plots for both the static and the dynamic model would be very explanatory.

Requirements

It is required that AUC should be above 0.96 for prediction 1 second before lane crossing and 0.90 for the 2 second prediction. It is also a target that the algorithm shall compute the results within 0.1 ms and that the memory space shall not exceed 100 kB RAM and ROM.

Questions:

1. Can you go though all the information and make sure that it is correct.
2. There is also a use case scenario related to the need for a lane change. Thomas has told me after discussions with Andres that testing this is really the same as testing the actual lane change. Can you confirm this? Elaborate?