Classification of Drivers Manoeuvre for Road Intersection Crossing with Synthetic and Real Data

Mathieu Barbier[1,2], Christian Laugier[2], Olivier Simonin[3] and Javier Ibañez-Guzmán[1]

Abstract— **When approaching a road intersection, drivers consider several factors and choose amongst different likely manoeuvres. For an autonomous agent, it is fundamental to understand what other drivers are doing before deciding their own manoeuvres. These are seldom be the same as intersections differ and the situations too. Whilst, learning techniques can be used to process features of trajectories and to predict manoeuvres of others cars. The problem with such approaches is the difficult process of recording data for each intersection, not only of the subject vehicle but of the other vehicles.**

To address this problem, a hybrid data set was constructed. It is built in a simulated environment and completed with real data after has driven multiple times across the intersection. To analyze these data, classification technique is used to find the common range of features for each manoeuvre. Random forest classifiers are used in conjunction with our functional discretization to analyze the trajectories of cars approaching an intersection. The classifiers can determine the longitudinal manoeuvre as well as the direction. We show how our approach performs compared to other classifiers and space discretization. In addition, we demonstrate the impact and the usefulness of the mixture between simulated and real data. An improvement of 30% accuracy is obtained with the hybrid data set, and 5% using our functional discretization with respect to baseline approach.

I. Introduction

A. Rationale

Whilst road mortality has seen a reduction due to improvement in vehicle design and legislation, accidents at road intersections remain high. In France alone[1] there were 3384 deaths in 2014 and a cost of 37.3 b € for the society. With occlusion, the main problem invoked by drivers is the misunderstanding of manoeuvres of others drivers and pedestrians.

Approaching an intersection, a car has few manoeuvre choices. First the car can decide to pass in the intersection if the context allows it. In case there is not enough information or a gap in the traffic flow, the driver can decide to yield and merge with the flow of cars. The last alternative is to stop the car because of road regulation (traffic sign) or if the upcoming traffic flow does not allow the car to pass.

It is crucial for an autonomous system to know what is the manoeuvre intended by another driver. Reasoning only at a physical level is not enough to accurately understand the behavior of other drivers [2]. However, at each intersection, trajectories can be different because of local factors such as occlusion, narrow road or a wide turning angle. On a daily commute a driver will pass multiple times the same intersection and be exposed to various scenarios. Thus he/she will adapts his/her manoeuvres to that specific intersection. It also affects his/her comprehension of other drivers manoeuvres, thus improving his/her understanding of the scene.

Learning techniques have been used to address different types of situations (overtaking, merging, intersection). The work done on a single intersection with thousands of trajectories shows that it is possible to identify manoeuvres. However, a recurring difficulty is the availability of datasets and the complexity to gather thousands of passages in an intersection. Furthermore, the model learnt cannot be used for another intersection, thus requiring to rerecording thousands of trajectories. In order to reduce the cost and time of testing and validation, simulation tools are used to generate hundreds of test scenarios with realistic trajectories. The idea is to use simulation techniques to generate synthetic informations for training machine learning algorithm. In our context the nominal manoeuvre of cars can be learnt within a simulation and can be enriched while driving on the real intersection. This process is similar to human drivers on their daily commute. Discovering for the first time an intersection, they apply their knowledge about nominal behaviour to predict what the other drivers are doing. After multiple times he/she adapts his/her manoeuvres and the analyses of the scene helped by past experiences in this specific intersection. In this paper, the performances of data sets composed of different percentages of real and simulated data will be discussed.

In a previous work [3], we showed that the space of an intersection can be divided in order to take into account how drivers behave. This division will help to divide trajectories into segments more relevant for the classification and to create more accurate classifiers.

B. Contribution and paper outline

The purpose of this paper is to classify manoeuvre of cars approaching an intersection and to use this classification to predict other car manoeuvres. We propose to use features generated from segments of a trajectory, to train Random Forest Classifiers (RCF). A different classifier will be used for each areas from [3]. Performances of the approach are

[1]Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. name.surmane@renault.com
[2]Inria Grenoble Rhône-Alpes , Chroma team,655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France name.surmane@inria.fr
[3]INSA Lyon, CITI Lab., Inria Chroma team,655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France name.surmane@inria.fr
This work is supported by a CIFRE fellowship from Renault S.A.S

compared with another classifier (SVM) as well as another manner to discretized the space (Rectangular). Synthetic and real data were used for training. The advantage of using an hybrid data set will be highlighted and discussed.

This paper is organized as follows. Section II presents related work on manoeuvre classification. Principle of random forest classification and its application to our problem is presented in section III. Section IV describes how data has been gathered from real and simulated environment. Section V presents and comments on a comparison between our approach against other classifier and discretization.

II. Related Work

A manoeuvre refers to actions that a driver can do while driving. In the case of a road intersection there are multiple actions that can be combined. The first aspect is the direction that the car will follow at the intersection. It could be turning left/right or going straight regarding where the driver want to go. Secondly, the longitudinal movement that is divided into three categories at an intersection: stop, pass and yield; that motion is constrained by traffic flow and traffic law. At last if the driver complies with what the situation required or if he/she followed an erratic behavior. In this paper the first two aspects will be used as classes for training classifiers.

In most studies, same measurements are used for manoeuvre classification. Velocity, acceleration and distance to the intersection are the most relevant to the analysis of the longitudinal aspect of the manoeuvre [4][5]. Whereas, heading and lateral position in the lane are more relevant for the direction. Other contextual information such as traffic light status or distance to the intersection can also help the classification. They can be recorded by CAN bus or with a perception system. If it is possible to look at the drivers, other information such as gaze, pressure on brake pedal can be recorded and can provide clues about what the drivers intend to do [6]. However the latter option causes a privacy issue and is more dependent to the driver. Thus, information from the ego vehicle that could also be perceived were used to train local classifiers.

Turning light is mandatory when doing a turning manoeuvre approaching an intersection, thus their states would be highly relevant for classification. However, a recent study[7] shows that in 1 out of 4 situations where there would be required, they are not used correctly. Because of this and the complexity to perceived turning light, most of the authors choose ignore them. Thus, it increases the robustness of the classifier against drivers errors.

One of the difficulties for machine learning is the construction of a data set. There is no, to our knowledge, publicly available data set that contains the previously stated measurements (velocity, acceleration, position) with a large number of passages in the same intersection. [8][4] used their own data set with respectively 300 000 and 50 000 passages in intersections to train their classifier. Naturalistic data has for advantage to contain most of manoeuvres that could happened at an intersection with different traffic densities and a large range of drivers. However they require a huge amount of work to be built and resulting classifiers might overfit to the observed intersection. Thus it would not be possible to set up such system for a larger number of intersections. Simulation tools have the advantage of being able to generate a large amount of data in a short period of time. Recorded trajectories are complete (no data missing) and the level of noise in the measurement can be controlled. However, the embedded model for manoeuvres generations might not be capable to grasp all details of the intersections. To address this problem, a simulation tool is used to generate a part of the data set then completed with real measurements while driving.

Different techniques of machine learning have been applied for road intersection crossing. Garcia et al.[5] used multilayer perceptron, a neural like model, to predict behavior primitives at a traffic light. This problems is similar to the yield problem since drivers have to choose their manoeuvre with uncertain informations. However in their scenario, the observed vehicle only reacted to the passage from green light to red light. Thus the behavior is highly dependent of the traffic light state. At a yield sign the behavior is more dependent of other cars and local problems such as occlusions.

Aoude et al.[8] used SVM combined with Bayesian filter to analyze the behavior of drivers at an intersection. They focus on the compliance of drivers at a traffic light. Their approach shows that SVM can be used to learn manoeuvres features. Their Bayesian filter is used to filter the output of the SVM across time avoiding jump in the prediction.

Using Random forest classifier, Gross et al.[4] trained their classifier with trajectories recorded from a fleet of cars. Their data is sparse due to a low sample rate (1 Hz). Therefore they aggregated 4 measurements (dynamic information and information about the preceding car) to create their feature vectors. They managed to classify direction and stopping intent for multiple intersections. However they address the problem of stopping as a binary problems whereas in real life there are many different ways to stop or pass. Also their feature space is composed of 35 features. Thus many trees with a large depth are required to obtain an accuracy score of 0.76 for direction and 0.77 for stopping intent.

III. Proposed framework

A. Random Forest classifier

Popularized by [9], random forest classifier (RFC) are better or at least comparable to other state of the art approaches for classification problems. Several aspects related to their implementation make them attractive for multiple kind of problems. First, they are fast for both learning and classification. Second, they can be easily parallelized for reduced calculation time [10]. They are also more robust than other approaches to noise in the data set [9]. RFC are also able to solve multi-classes problems without any further development compare to Support Vector Machine that need to be adapted.

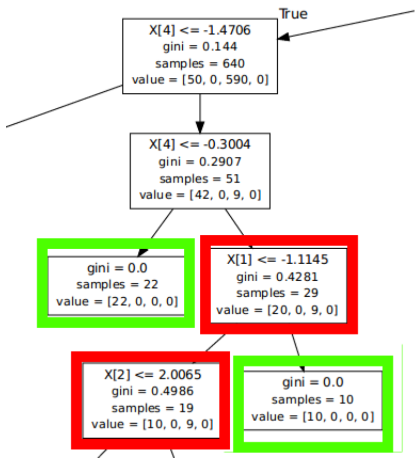


Fig. 1. Example of tree classifier, the figure focus on a branch that generate pure nodes in green and an impure nodes in red. The first row of a node is the selected test, the second the Gini score of the test

An RFC builds multiple independent trees, each of them to be trained with a different bootstrapped data set. A bootstrapped data set is a sub-set of the available training data. Thus a part of the data is used for learning, samples not included are refereed as Out-of-bag samples and will be used to estimate the Out-of-bag-error(OOBE). This method helps to reduce over-fitting and variance. With this approach some trees will be more qualified to classify some labels.

At each node of the a single tree a set of random test on the feature space is generated. Each test is judged regarding a quality measurement (in our case the Gini index) and the better test is selected to split the node. Each tree is grown until each node is pure (only one class in the node) or when a certain depth is reached. Example of a tree is given in figure 1. A tree t can be written as a function $f_t(x, \theta_t)$: $\mathbb{X} \longrightarrow \mathbb{Y}$ with $\mathbb{X}$ the feature space and $\mathbb{Y}$ the class space. θ_t captures stochastic elements of the tree (tests for each decision nodes) and x is a feature vector. Thus the entire forest can be written as $\mathcal{F} = \{f_1, ...f_T\}$ with T the number of tree and the probability of a class k given x is :

$$p(k|x) = \frac{1}{T} \sum_{t=1}^{T} p_t(k|x) \qquad (1)$$

Another advantage is a few tuning parameters required to construct the classifier. Firstly, the number of trees used in the forest T. After a certain amount of trees in the classifier, there is no more improvement on precision or variance. Secondly, the depth is used to controls the number of split in each tree. Normally a branch of a tree stops to grow once the purity measurement is below a certain threshold. However, this process might require too many splittings and this depth parameter stops the growth after a certain number of splits. Theses two parameters work together, stopping the growing of tree allows to fasten computation, and it can be expected than another tree has perform better to split that part of the data set. It is important to test different sets of parameters and to compare them in term of Out-of-bag error to find appropriate values.

B. Functional discretization

In our previous work[3], a functional discretization has been proposed and applied to road intersection crossing (example given in figure 6). It aims to simplify problems that requires a discretization of space by segmenting it differently. Trajectories from multiple cars have been used to train Gaussian processes to represent trajectories patterns. Areas of most likely collisions and approaching areas were found by analyzing these trajectory patterns. Then, this model has been divided in different parts given their overlapping and probability distributions. The approaching branch of the intersection is then divided in multiple areas with a similar context. For the classification of manoeuvre problems, the approaching areas are used since they represent areas where drivers should have a consistent behavior.

C. Framework

An original aspect of our approach is the use of different classifiers for each area found in section III-B instead of one for the entire space. For each area, a RFC is created to classify the direction manoeuvre and one other for the longitudinal manoeuvre. The advantage of local classifiers is that the segment of the trajectories used for training has been executed in a similar context. During the training phase, each trajectory of the data set is segmented using our functional discretization. Then features of each segment are computed and used to train both RFC of the local area. To test a new input, all the measurement recorded in area are used to compute the new feature vector. Then the feature vector is test by the corresponding local classifiers to estimate the manoeuvre of the observed car.

D. Measurement, features and labels

As discussed in section II, information from proprioceptive sensors were used to record trajectories. Available measurements were velocity, position (in UTM referential), yaw of the car and time. UTM position of the real car were converted in the referential of the simulated environment. From this measurement the acceleration of the vehicle was computed. Training classifiers directly with these measurements would be complex due to noise and relatively small deviation due to human errors. Positions of the car were left out of the feature space and used to match a measurement to the correct area. The feature space is composed of 6 features that correspond to the extrema of different measurements in one area.

- V_{max}, V_{min}:Maximum and minimum velocities corresponding to extremas in the velocity profile in ms^{-1}.
- A_{max}, A_{min}:Maximum and minimum acceleration corresponding to the extrema of the car acceleration in ms^{-2}.

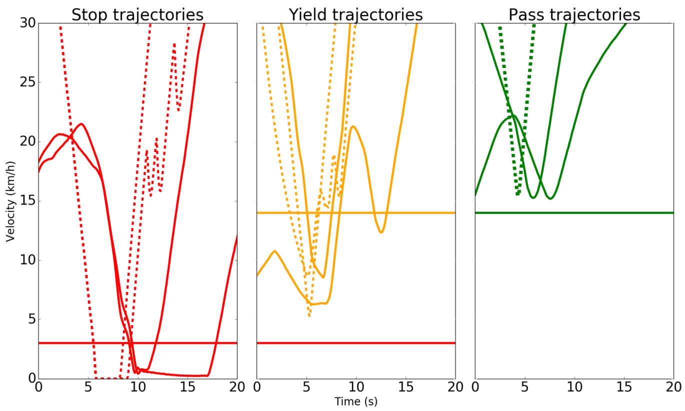


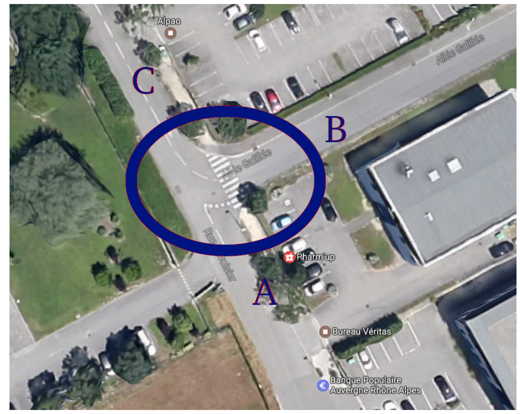
Fig. 2. Example of trajectories with their given labels, dashed curves are synthetic velocity profiles, plain curves are real velocity profiles, horizontal line correspond to velocity threshold of the class.

- H_{max}, H_{min}:Maximum right and left deviation from the mean heading in radian.

Then direction and longitudinal motion of each trajectory were annotated. For the direction the first and the last measurement are used to group trajectories of the same directions together. Then the minimum velocity in the overall trajectory is used to classify the longitudinal motion. With a minimum velocity of 0.8 m/s and below the trajectory is labeled as stop. It could be discussed that this value should be 0 m/s. But during field experiment drivers reduced their speed as if they would stop and kept a low speed instead of coming to stop as recommended by the traffic law. Between 0.8 m/s and 3.8 m/s, the trajectory is labeled as yield for car that reduced their speed to let an other driver pass or uncertain about the behavior of another driver. All trajectories over 3.8m/s were labeled as passing. Drivers reduced their speed according to the traffic law but are confident that their passage in the intersection is possible. Example of velocity profiles and classification is shown in figure 2. In the future these classes should be enriched with perception systems in order to take into account dynamic context and to create more classes.

IV. Data acquisition

The experiment was conducted on a T-intersection in a urban area close to Grenoble. The figure 3 shows a satellite and ground view of the intersection. There is no other intersection in a 50 meter range and cars can approach the intersection with a maximum velocity of 50 km/h. Trajectories of our experimental platform were recorded with an X-sens, that combined inertial measurement information and GPS, to accurately locate our vehicle in space. Velocity, position and heading of the car were recorded at 100 Hz. Three different drivers carried out multiple manoeuvres in an uncontrolled environment. The traffic was relatively low (no more than two vehicles interacting together) and without pedestrians. This environment was reproduced with the simulation platform Scaner™ as shown in figure 4. In total, 2.5 hours of 3 cars driving together in the same intersection was recorded. Trajectories from branch A were extracted



(a)

(b)

Fig. 3. Satellite view (a) and ground view from the branch A (b) of the intersection. It can be found at 45°13’02.2"N 5°48’46.0"E

Fig. 4. Simulation environment in Scaner, cars drove freely at various speeds

and will be used for the rest of the paper. Drivers chose freely to turn right or to continue straight. This branch is the most interesting, because of its yield sign that force drivers to choose between three possibles manoeuvres: pass, yield and stop. Each trajectory was given a label regarding the lowest velocity during the manoeuvre. The table I shows the composition of the data set that was gathered by simulation and field experimentations.

V. Results and discussion

Following results show the performances of the proposed approach. The implementation is made using python and scikit learn[11] for training and testing.

TABLE I
DATA SET COMPOSITION

data set	passage	pass	yield	stop	Straight	turn right
Simulated	100	43	37	20	39	61
real	37	10	21	6	18	19

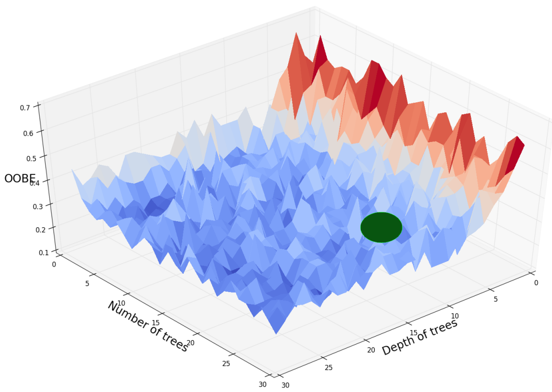


Fig. 5. OOBE error with different parameters for the random forest classifiers, with 20 trees of maximum depth 10 a score of 0.20 is obtained (highlighted in green). Obtained with a data set containing 20% of real data.

A. Random forest topology

The depth and the number of tree are important parameters that control the creation of the RFC. Thus, it is important to find optimized value for each of this parameters. The OOBE is used to control the quality of the learning phase of the RFC. Figure 5 shows the errors obtained with different set of parameters. It can be observed that with trees with more than 10 split (depth parameter) there is no improvement of the OOBE. The number of trees does not affect a lot the OOBE when the forest contains more than 10 trees. A depth of 10 and a number of 20 trees have been selected to construct RFC for the rest of the paper. The number of trees has been increased for stability reason. In this configuration an OOBE of 0.20 is obtained (highlighted in green in Figure 5).

B. Functional discretization against Rectangular discretization

Using our functional discretization proposed in [3], a comparison is made with rectangular areas. The rectangular areas are 10 meters wide (similar to [4]). Areas from both discretization are matched regarding their centroid. Figure 6 shows the results of both discretization.

Accuracy was compute as the result of the classification of real data. It can be observed in table II an improvement of 5.4% for longitudinal manoeuvre and 5.8% for direction detections. Learning Classifier in areas that are more relevant to the motion of the car is thus more accurate. Reducing the size of the rectangle segmentation could improve their performances but would have required more classifiers thus a more complex system.

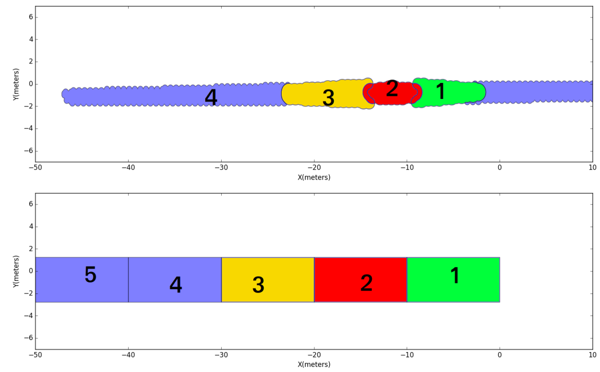


Fig. 6. Discretization of the intersection space using the functional discretization (top) and rectangular segmentation (bottom), the point 0,0 is the center of the intersection

TABLE II
COMPARISON OF THE ACCURACY OF THE CLASSIFICATION USING THE FUNCTIONAL DISCRETIZATION AND A RECTANGLE DISCRETIZATION

Discretization	1	2	3	4	5	Mean
Functional	0.77	0.89	0.81	0.73	na	0.8
Rectangle	0.82	0.85	0.71	0.69	0.67	0.746
Improvement						+5.4%

(a) Longitudinal manoeuvres classification

Discretization	1	2	3	4	5	Mean
Functional	0.84	0.93	0.77	0.71	na	0.81
Rectangle	0.84	0.89	0.67	0.67	0.70	0.752
Improvement						+5.8%

(b) Directions classification

C. Comparison to Baseline approach

Support vector machine (SVM) has been chosen for comparison due to thier popularity in the machine learning community. They are often used as a baseline [4] to show new approach improvements. Due to the multi-class aspect of the longitudinal manoeuvre detection, an “one-against-one” approach is used for the SVM classification.

For this experiment, a hybrid data set, composed of 20% of real data and completed with synthetics ones, were used in conjunction with the functional discretization.

Table III shows the results obtained in each area with K-fold cross-validation with k=5. It can be observed that implementation with an RFC performs slightly better than SVM. This is due to the variety of tree structures and the combination of tree outputs. Another interesting result is the

TABLE III
COMPARISON BETWEEN RFC CLASSIFICATION AND AN SVM

	Discretization	1	2	3	4	Mean
Longitudinal manoeuvres	RFC	0.91	0.88	0.68	0.73	0.82
	SVM	0.91	0.88	0.62	0.69	0.80
	Improvement					+2.0%
Directions	RFC	0.92	0.93	0.57	0.70	0.803
	SVM	0.81	0.70	0.60	0.68	0.712
	Improvement					+9%

TABLE IV

FEATURES IMPORTANCE

	V_{min}	V_{max}	A_{max}	A_{min}	H_{max}	H_{min}
Longitudinal	0.29	0.29	0.16	0.17	0.02	0.03
Direction	0.21	0.19	0.15	0.17	0.13	0.11

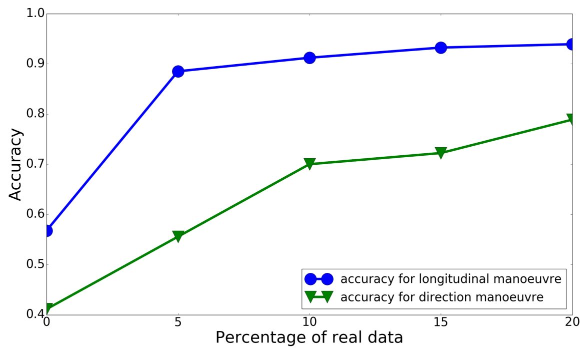


Fig. 7. Accuracy of the classification using different composition of hybrid data set, accuracy is obtain with real data testing

features importances. For RFC, it correspond to the number of time a feature has been used to split a node. Table IV shows results obtained with the same implementation. As expected, features related to velocity and acceleration are more important for longitudinal manoeuvres. Direction classification required a more balanced usage of features.

D. Results with the hybrid data set

For this part different data set composition will be used to train the same RFC with the functional discretization. The accuracy score is obtain using only real data. Figure 7 shows the evolution of the performances using different percentage of real data in the training set. It can be observed that with 20% of real data the performances have doubled for the direction manoeuvre and rise by 30% for the longitudinal. It shows that with a rather small amount of real data, performances of the classifier improved rapidly.

E. Discussion

Results showed that the proposed classification scheme with RFC, the functional discretization and the hybrid data set should perform better than any combination of classification, data set composition and discretization. In order to asses the quality of a classifier, receiver operation curve (ROC) is used to visualize performances of each classifier. The top left point (in figure 8) being an optimal performance with only true positive and no false positive. The closer the curve is to that point the better is the classification. This characteristic in shown trough the area below the curve that can be read in the legend of the figure 8. The steepness of the curve is also important to show how fast the true positive rate increases. The curve is an average of all local classifiers. For the classification of longitudinal manoeuvre, which is

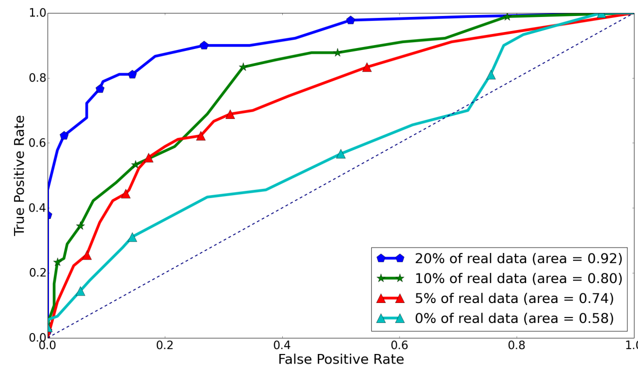


Fig. 9. Receiver operation curves curves obtained with different compositions of dataset for longitudinal classification, testing is made against real data

multi-class, curves are an average of the ROC of every class. Results in figure 8 where obtained with attempting to classify real measurements. It shows that the chosen implementation is always better than any other approaches.

The solution using only simulation data can still perform better than a random guess (dashed blue line),in figure 9 showing that information from the simulation are useful in the learning. The addition of a certain percentage of real data improves the classification. It can be observed that with a purely synthetic data-set, classifiers still managed to perform slightly better than the random guess. It shows that the information provided by the simulation help the classification. Only the ratio of synthetic and real information has been discussed in this paper. It would also be interesting to look at the impact of the volume of data required. For our experimentation the size of data sets were relatively low, especially the real part. If more simulation time is spent, the performances of classification could increase. But, as hinted with these results, the use of even a small amount of real data increases performances. The model learnt with our approach could be re-injected in the simulation tool in order to provide more accurate cars behaviors for validation of other problems.

The use of the functional discretization always improved the result of the classification. This discretization take into account where drivers are most likely to adapt their trajectories to the local context. Thus resulting classifiers are fitted to a more relevant feature range. For example, leaving area 4, the driver is expected to slow down to adapt his speed to the intersection and then entering area 3 slow down if an yield or a stop is required. The third area of rectangular discretization is in-between two areas of the functional discretization and less accurate than its counterpart in the functional discretization. It is caused by the driver being in a transitional state and features show more change in the third rectangle.

The last aspect is the use of RFC rather than an other classification method. Curves using this approach shows better steepness and a wider area. Thus outperforming the

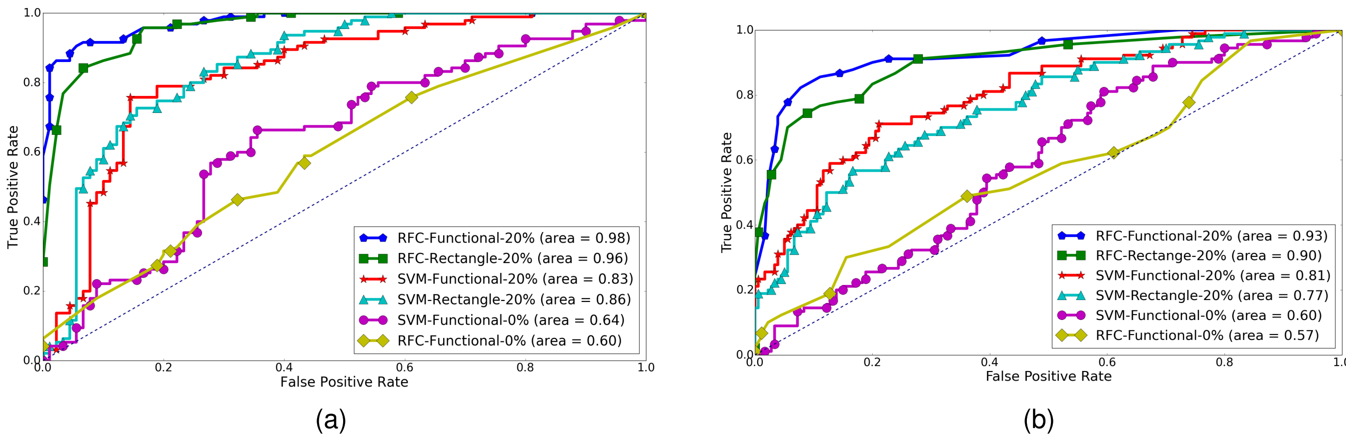


Fig. 8. Receiver operation curves for direction (a) and longitudinal manoeuvre(b). The dashed line corresponds to a random classification. The closer the curve is to the top left corner the better is the classification method.

implementation with an SVM. RFC are known to be robust against over-fitting, thus it is possible that the SVM has over-fit on the simulated part of the data set when the RFC did not. However this computation time is sufficient for real time implementation and there is also enough margin to add more trees if required. A more advanced strategy could be used to train the RFC for on-line learning. For example, starting from a forest learnt with only synthetic data, some trees could be replaced by newly trained trees with real data. This would enhance the management of classifiers in time.

VI. Conclusion

In this paper a model for classification of manoeuvres for a road intersection has been done using Random Forest Classifier and a functional discretization. Results show better performances compared to others baseline approach (SVM and rectangular discretization). To cope with the requirement of naturalistic data set, we showed that with a relatively small data set that contains synthetic data and real data it is possible to accurately classify real manoeuvres. Only informations from the proprioceptive sensors were used, but further development could include information for exteroceptive sensors to add features from the context. It could help to add new labels to more precisely identify the behaviors of drivers. The feature range for each manoeuvre could be used to tune trajectory planner to produce trajectories more dependent on the local situation.

Acknowledgment

The authors would like to thanks Jérôme Lussereau, Jean-Alix David for their support for the experimental part. This work is partially supported by the european project. ENABLE-S3

References

[1] ONISR, “La sécurité routière en france bilan de l’accidentalité de l’année 2013,” tech. rep., Observatoire national interministériel de la sécurité routière, http://www.securite-routiere.gouv.fr/la-securite-routiere/l-observatoire-national-interministeriel-de-la-securite-routiere/accidentalite-routiere, 2011.

[2] S. Lefevre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[3] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, “Functional discretization of space using gaussian processes for road intersection crossing,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 156–162, Nov 2016.

[4] F. Gross, J. Jordan, F. Weninger, F. Klanner, and B. Schuller, “Route and stopping intent prediction at intersections from car fleet data,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, pp. 177–186, June 2016.

[5] M. G. Ortiz, J. Fritsch, F. Kummert, and A. Gepperth, “Behavior prediction at multiple time-scales in inner-city scenarios,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1068–1073, June 2011.

[6] A. Tawari, T. Misu, and K. Fujimura, “Predicting unexpected maneuver while approaching intersection,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2225–2229, Nov 2016.

[7] R. Ponziani, “Turn signal usage rate results: A comprehensive field study of 12,000 observed turning vehicles,” tech. rep., SAE Technical Paper, 2012.

[8] G. Aoude, V. Desaraju, L. Stephens, and J. How, “Driver behavior classification at intersections and validation on large naturalistic data set,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 724–736, June 2012.

[9] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] T. Sharp, *Implementing Decision Trees and Forests on a GPU*, pp. 595–608. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.