DRIVING BEHAVIOUR CLUSTERING FOR REALISTIC TRAFFIC MICRO-SIMULATORS

Alessandro Petraro, Federico Caselli, Michela Milano
Department of Computer Science and Engineering
University of Bologna
viale Risorgimento 2, Bologna, Italy
alessandro.petraro2@studio.unibo.it, f.caselli@unibo.it,
michela.milano@unibo.it

Marco Lippi
Department of Sciences and Methods for Engineering
University of Modena and Reggio Emilia
via Amendola 2, Reggio Emilia, Italy
marco.lippi@unimore.it

KEYWORDS

Agent-based modelling; Traffic Micro Simulators; Clustering Algorithms

ABSTRACT

Traffic simulators are effective tools to support decisions in urban planning systems, to identify criticalities, to observe emerging behaviours in road networks and to configure road infrastructures, such as road side units and traffic lights. Clearly the more realistic the simulator the more precise the insight provided to decision makers. This paper provides a first step toward the design and calibration of traffic micro-simulator to produce realistic behaviour. The long term idea is to collect and analyse real traffic traces collecting vehicular information, to cluster them in groups representing similar driving behaviours and then to extract from these clusters relevant parameters to tune the micro-simulator. In this paper we have run controlled experiments where traffic traces have been synthetized to obtain different driving styles, so that the effectiveness of the clustering algorithm could be checked on known labels. We describe the overall methodology and the results already achieved on the controlled experiment, showing the clusters obtained and reporting guidelines for future experiments.

INTRODUCTION

Vehicular mobility is a complex man-made socio-technical system involving the road and communication infrastructures, road side units and drivers. Vehicular mobility affects many aspects of our everyday life, shaping the environment around us, underpinning economic growth and affecting our health and quality of life. If we restrict ourselves to urban mobility, it accounts for more than 40% of CO_2 emissions and more than 70% of other pollutants from transport[1]. Understanding mobility patterns, identifying criticalities in such a complex system, assessing the performance of the road network in specific areas is essential for decision makers that have to manage the system and plan interventions to improve the infrastructure.

Important tools that support such decision making process are traffic simulators, namely mathematical or agent-based models mimicking the traffic dynamics on a road network. They are classified in macro- and micro-simulators (Helbing et al., 2002). Macro-simulators are often based on traffic flows and describe the collective behaviour of vehicle dynamics in terms of vehicle density and average speed in time. Micro-simulators, instead, model the single vehicle and account for the driver behaviour that influences the speed and position of the vehicle in time. Thus, in micro-simulators, traffic patterns emerge by the interaction of each vehicle dynamics. It is extremely difficult to model single drivers and obtain a realistic emerging behaviour.

To provide useful support to decision makers, traffic simulators should be realistic and model real traffic patterns. It is clear that if the drivers are modelled in an unrealistic way, many emerging and realistic patterns are lost, while unrealistic patterns arise.

In this paper we propose a method for configuring realistic micro-simulators and we introduce the first results achieved in this direction. The idea is to use real traffic traces, to cluster them in groups sharing similar driving behaviour and then use the cluster features to configure the agents in the traffic simulator. To assess the feasibility of this methodology, we have started with a controlled experiment and generated synthetic traffic traces to understand which parameters are discriminant for the clustering algorithm and how to ex-

[1] https://ec.europa.eu/transport/themes/urban/urban_mobility_en

Proceedings 31st European Conference on Modelling and Simulation ©ECMS Zita Zoltay Paprika, Péter Horák, Kata Váradi, Péter Tamás Zwierczyk, Ágnes Vidovics-Dancs, János Péter Rádics (Editors)
ISBN: 978-0-9932440-4-9/ ISBN: 978-0-9932440-5-6 (CD)

tract and manipulate them. We have compared traces that are ordered and not ordered, long and short and we have come out with some guidelines for conducting experiments on real traces.

The paper is organized as follows. In the next section, we first propose the concept underlying our method, then we introduce SUMO (Krajzewicz et al., 2012), the micro-simulator used, and we show how we applied the clustering algorithm. Finally, we present an experimental evaluation on a set of synthetic traces providing results and guidelines for future tests.

CONCEPT

We have designed a process aimed at configuring a micro-simulator to obtain realistic behaviour. The system pipeline is depicted in Figure 1. The starting point concerns the collection of real traffic traces from vehicles. The main parameters we have to collect are speed, position and acceleration at any point in time. These time series should be processed in real-time, stored in a data base and then manipulated to extract a meaningful training set for the clustering algorithm.

One important aspect of this processing phase is that, often, time series have different lengths, or contain many values that refer to stops (either at traffic lights or in congestions) that do not provide any meaningful insight on the driving styles. Therefore we have basically three possibilities: (1) either cut the time series to obtain feature data of the same length, (2) order the time series and cut them after the sorting or (3) use aggregate values (average speed/acceleration, standard deviation, maximum speed/acceleration). In the experimental result section, we will consider these aspects and test each alternative to understand its effectiveness in extracting driving styles.

The clustering algorithm then creates clusters on the feature space and obtains groups of traces that hopefully share some driving style characteristics. The cluster dimension (namely the number of vehicle traces in the cluster), and other aggregate parameters (max speed, max acceleration, standard deviations of speed and acceleration) are extracted from clusters and fed into the simulators to generate vehicles with the same characteristics. In the future, self-driving cars could also take advantage of these clustered driving styles, so as to learn typical human behaviors.

This paper is a first but significant step toward the process described above. Here we focus on a controlled experiment, where traffic traces are synthetic and are generated to have specific features. In this way we artificially create traffic traces exposing a controlled number of driving styles, to double check if the clustering algorithm finds homogeneous clusters with respect to the driving style.

Clearly, the experimental setup should cover scenarios where driving styles are very different one another and scenarios where driving styles are somehow more similar. We will show in the experimental result section the generated scenarios and the corresponding results.

THE SUMO SIMULATOR

SUMO (Simulation of Urban MObility) (Krajzewicz et al., 2012) is a microscopic, time-discrete traffic flow simulation platform used in this paper to collect the traffic information for the controlled experiments.

In SUMO each vehicle is uniquely simulated: it has an unique identifier, a departure time and a route (defined as a list of streets) that it will follow through the road network. Furthermore, each vehicle can be characterized by a set of features (called type) describing how it will behave in the simulation, including physical information (ie its maximum speed and acceleration) and other more specific parameters that regulate advanced aspects of the simulation, such as the driver's willingness to respect the speed limits.

Since our goal is to assess the feasibility of our concept with SUMO, to create the different driver behaviours we focused on values that can be easily collected from the observation of real traffic. In particular, to characterize the different types we used the maximum acceleration and deceleration and the willingness of the driver to follow the speed limit. This last parameter can be controlled in SUMO via a Gaussian distribution where $speedFactor$ is the mean and $speedDeviation$ is the standard deviation. When a vehicle enters the simulation SUMO computes its speed factor, using the Gaussian defined by its type. This means that the real maximum speed of a vehicle on a lane is $vehicleSpeedFactor \times laneSpeedLimit$.

SUMO allows many kinds of data outputs for each simulation, and also offers an API that can be leveraged to control the simulation online through another program. When used in this mode, SUMO allows the client to access many aspects of the simulation undergoing at every time step and also to change the values. This makes collecting custom information very efficient, without the need to parse large output files.

We used the software platform to generate custom traffic demands and interacted directly with the simulator online. We used these features to create the synthetic scenarios that were used to validate our approach, and to extract the traffic traces to perform the analysis of the simulations.

CLUSTERING

Given a collection of traffic traces, our goal is to look for drivers that share similar characteristics. From a machine learning point of view, this is an unsupervised learning task, since we do not know in advance the categories into which the examples should be partitioned. Differently from a supervised learning setting, when one is given a collection of object-target pairs with the aim of learning a function that associates the objects to their targets, in an unsupervised setting we are only given unlabeled observations, with the goal of automatically detecting relevant, common patterns among the examples (Hastie et al., 2001). This setting is particularly appropriate for our scenario, since it is unlikely to pretend to know in advance a precise set of driver categories, but it is much more reasonable to search for

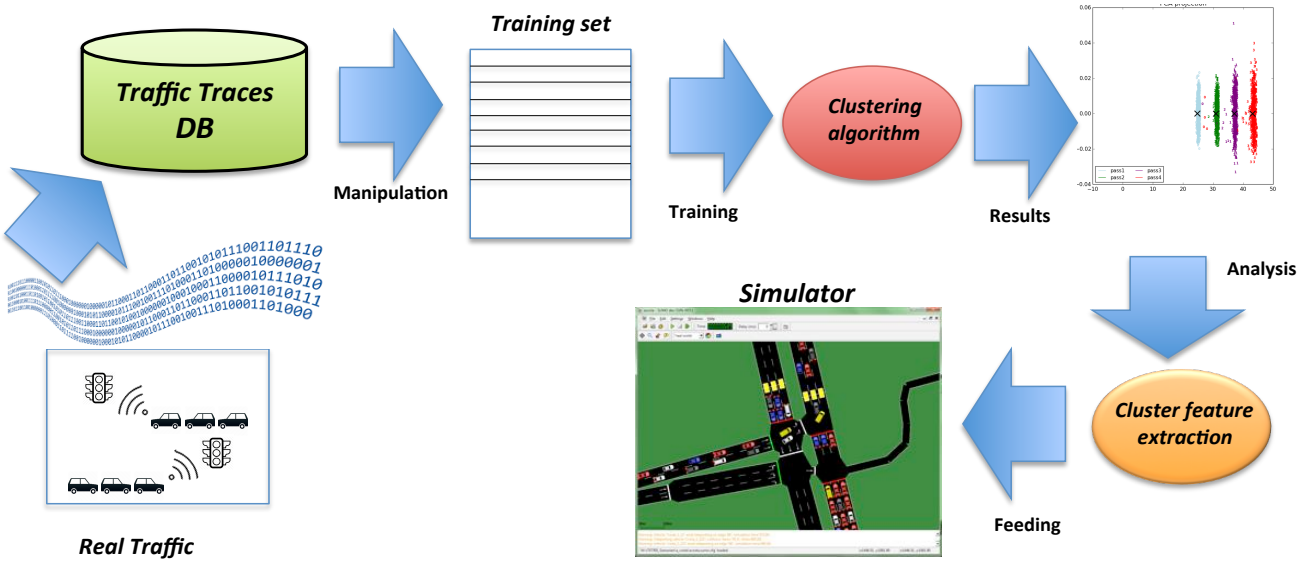


Fig. 1: Process flow for configuring realistic micro-simulators

emerging behaviors directly from data observations.

To this aim, we employ one of the most used and studied clustering algorithms, namely k-means (Hartigan and Wong, 1979). We hereby remark that other more sophisticated algorithms could indeed be applied to the same problem: yet, our goal in this paper is to provide a proof-of-concept of the whole system, thus we selected such a simple yet effective algorithm, leaving for future work the analysis of alternatives. Given n data points and an integer k, k-means partitions the data into a set of k clusters. This is done by finding the k cluster centers with an iterative procedure: starting with k initial centers (e.g., randomly chosen), the remaining points are associated each to the closest center. Then, the centers of mass of the so-obtained clusters are computed, which produces a new set of k cluster centers. Such steps are repeated until convergence. The algorithm is guaranteed to converge to a local optmimum, but it greatly depends on the initialization of the cluster centers. In this paper we employ two standard techniques that typically improve clustering results: (i) a smart initialization of the centers, named k-means++ (Arthur and Vassilvitskii, 2007), that tries to have an initial set of spatially distant points; (ii) multiple re-starts of the algorithm, finally choosing the best solution according to a certain criterion. More details on the metrics used to assess the goodness of clustering will be given in the experimental section. Here we just anticipate that, since we operate in a controlled (simulated) environment, clearly we also know the true labels of the vehicles (since we generate them). This allows us to use clustering evaluation metrics that exploit knowledge of the labels.

With respect to our specific problem, the main issue that has to be solved when feeding data to the clustering algorithm is how to represent each traffic trace. In fact, clustering algorithms (including k-means) typically assume all data samples to have the same dimensionality, that is they are represented by the same number of features. In the case of traffic traces, instead, this is clearly not true, since the length of the trace of each vehicle depends on how long the vehicle has been observed (in the controlled experiment, how much time it spends within the simulation). Moreover, the trace may possibly contain information about both the speed and the acceleration of the vehicle. Several solutions can be designed to address these issues. In this work we considered four different possibilities: (i) compute aggregated statistics (e.g., mean, variance, etc.) of each trace, to be used as feature vectors; (ii) choose the m largest values of the trace (in decreasing order); (iii) choose the m smallest values of the trace (in increasing order); (iv) choose the first m timestamps of the trace.

EXPERIMENTS

We now present the experimental evaluation that we conducted in a controlled setting implemented within SUMO, version 0.27.1. We considered three different scenarios of increasing difficulty for the clustering algorithm, to study the performance of our approach with respect to the heterogeneity in the generated traffic traces. Each simulation has run for a period of 4 hours, with about 14,000 vehicles. Then, we extracted the 25% of vehicles with the longest traces in the simulation, so that several lengths for the feature vectors could be tested in the experiments. Therefore, each scenario contains about 3,600 vehicles. As for the road network, we used a portion of the Bologna metropolitan area, depicted in Figure 2. In all the simulations, we kept SUMO parameter $speedDev = 0.1$ (representing the speed standard deviation).

For each scenario, we have applied the clustering algorithm on (i) aggregated data (ii) time series containing only speed values in time, (iii) time series containing acceleration values in time, and (iv) combined time

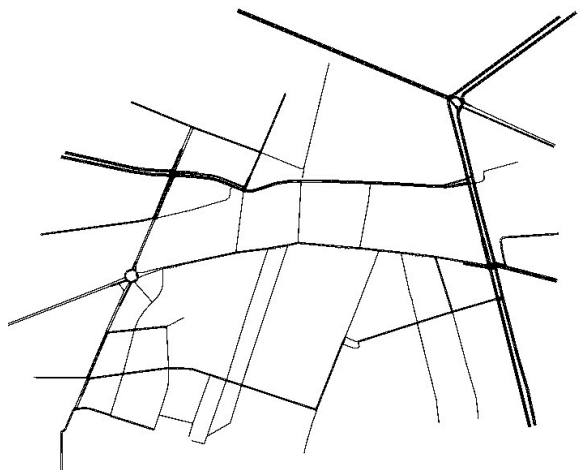


Fig. 2: Simulation scenario: a portion of the road network in the metropolitan area of Bologna.

series of speed and acceleration. The feature vector of aggregated data consists of three values: maximum speed, maximum acceleration and maximum deceleration. Since these will be exactly the parameters tuned in SUMO to generate different class vehicles in each scenario, we expect clustering with aggregated data to be very effective, and to act as a sort of upper bound for all the other approaches. Since perfectly knowing which are the discriminating parameters of the vehicle categories is clearly not realistic, it is interesting to compare the results of aggregated statistics with those obtained by directly employing the time-series (or a portion thereof).

When we did not compute aggregated statistics of the time series, we had to cut feature vectors to a fixed length of m timesteps. We considered values for $m = 5$ up to $m = 60$ with step 5 (from preliminary experiments, we observed no improvements with larger values of m). The trimmed traffic traces were considered both unsorted and in descending sorted order: experiments show that ascending ordering is useless, since low values of the time series are very similar across all the vehicle classes, typically corresponding to vehicle stops. For descending sorting, we ordered the traces for decreasing speed, and then considered the corresponding acceleration (thus, the acceleration values are not in descending order). In all the experiments, we set $k = 4$ for k-means. A validation of the clustering performance as a function of this parameter is left for future work. To evaluate clustering performance, we employ four metrics: Homogeneity (h), Completeness (c), v-measure (v) and Adjusted Rand Index (ARI). Homogeneity and completeness are strictly intertwined metrics: the first measures the degree of homogeneous labels within each cluster, whereas the second measures at what extent members of a certain class are assigned to the same cluster. Formally, they are defined as:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{1}$$

Class	a	d	sf
1	1.6	4.0	0.8
2	2.2	4.5	1.0
3	2.8	5.0	1.2
4	3.4	5.5	1.4

TABLE I: Parameters employed for the generation of vehicles in the easy scenario: a is acceleration, d is deceleration, sf is the speed factor.

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{2}$$

where $H(C)$ is the entropy of the (true) classes, $H(K)$ is the entropy of the clusters, and $H(C|K)$, $H(K|C)$ are the two conditional entropies. The v-measure is the harmonic mean between h and c. The Adjusted Rand Index (ARI) is instead defined starting from Rand Index (RI):

$$RI = \frac{A + B}{Z} \tag{3}$$

where A is the number of pairs of examples that belong to the same class and to the same cluster, B is the number of pairs of examples that belong to different classes and also to different clusters, Z is a normalization factor (the sum of all possible pairs of examples). ARI is defined as an adjustment of RI taking into account chance normalization:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{4}$$

where $E[RI]$ is the expected RI of a random cluster assignment, and $\max(RI)$ is the maximum possible value of RI.

A. Easy scenario

In the first scenario, we generated four vehicle categories having different physical properties (acceleration and deceleration) and also different aggressiveness (willingness to ignore the speed limit). This is clearly the easiest scenario for the clustering algorithm, as vehicle categories should be well separated and distinguishable. The chosen parameters are shown in Table I. The simulation covers 3,595 vehicles, 36.2% of class 1, 25.1% of class 2, 21.2% of class 3, 17.5% of class 4. Table IIa reports the clustering metrics employed with the different settings. As expected, different driving styles are easily recognizable as they greatly vary for what concerns the speed and acceleration of the vehicles. Unsorted traffic traces lead to reasonable clustering results, with a v-measure in the best case equal to 0.754 (using both speed and acceleration). Descending, sorted time series produce instead very good results: the shortest time series (e.g., $m = 5$) work best, as large speed and acceleration values are the most informative data. Results with aggregated data confirm an almost perfect clustering, as expected. In Figure 3a we show a representation of the best clustering results obtained

Trace	Sorting	m	h	c	v	ARI
S	None	10	0.745	0.772	0.758	0.788
A	None	15	0.635	0.671	0.652	0.693
SA	None	20	0.741	0.768	0.754	0.786
S	Dec.	5	0.977	0.979	0.978	0.989
A	Dec.	5	0.979	0.980	0.979	0.989
SA	Dec.	5	0.872	0.882	0.877	0.915
Agg	–	–	0.985	0.985	0.985	0.993

(a)

Trace	Sorting	m	h	c	v	ARI
S	None	15	0.656	0.685	0.670	0.697
A	None	15	0.572	0.570	0.571	0.602
SA	None	25	0.716	0.739	0.727	0.755
S	Dec.	5	0.987	0.988	0.988	0.994
A	Dec.	50	0.318	0.326	0.322	0.233
SA	Dec.	5	0.884	0.889	0.887	0.914
Agg	–	–	0.993	0.993	0.993	0.997

(b)

Trace	Sorting	m	h	c	v	ARI
S	None	10	0.723	0.737	0.730	0.741
A	None	15	0.712	0.723	0.718	0.748
SA	None	20	0.727	0.740	0.733	0.747
S	Dec.	30	0.485	0.843	0.616	0.476
A	Dec.	25	0.590	0.661	0.623	0.509
SA	Dec.	15	0.474	0.740	0.578	0.459
Agg	–	–	0.989	0.988	0.989	0.994

(c)

TABLE II: Best clustering results obtained in each scenario: (a) easy, (b) intermediate, (c) hard. h is cluster homogeneity, c is completeness, v is the v-measure, and ARI is Adjusted Rand Index. For traces, S stays for speed, A for acceleration and SA for a combination of the two. Agg indicates aggregated features.

for this scenario ($m = 5$, descending ordering, negligible differences if using acceleration or speed). Examples have been projected to a two-dimensional feature space via Principal Component Analysis (PCA) (Jolliffe, 2002), where colors represent the cluster assignment by k-means, and digits indicate the true vehicle label.

B. Intermediate scenario

The second scenario we consider has an intermediate level of complexity, since only the aggressiveness of the vehicles changes. Again, we generated four traffic categories corresponding to four different values for SUMO parameter $speedFactor$: $\{0.8, 1.0, 1.2, 1.4\}$, which are the same employed in the easy scenario (but this time without differentiating also the physical properties). The simulation covers 3,643 vehicles, 32.6% of class 1, 26.1% of class 2, 21.5% of class 3, 19.8% of class 4. Table IIb reports the clustering metrics employed with the different settings. Similarly to the previous scenario, the clustering of unordered traffic still achieves metrics

k	m	h	c	v	ARI
4	20	0.727	0.740	0.733	0.747
6	25	0.793	0.683	0.734	0.742
8	25	0.806	0.585	0.678	0.592
10	25	0.821	0.521	0.638	0.489

TABLE III: Performance measurements with different numbers of clusters k in the hard scenario. In this setting we employ unsorted traces with both speed and acceleration. We report h as cluster homogeneity, c as completeness, v as the v-measure, and ARI as Adjusted Rand Index.

Class	a	d	sf
1	2.2	4.5	0.8
2	2.8	4.5	0.8
3	2.2	4.5	1.2
4	2.8	4.5	1.2

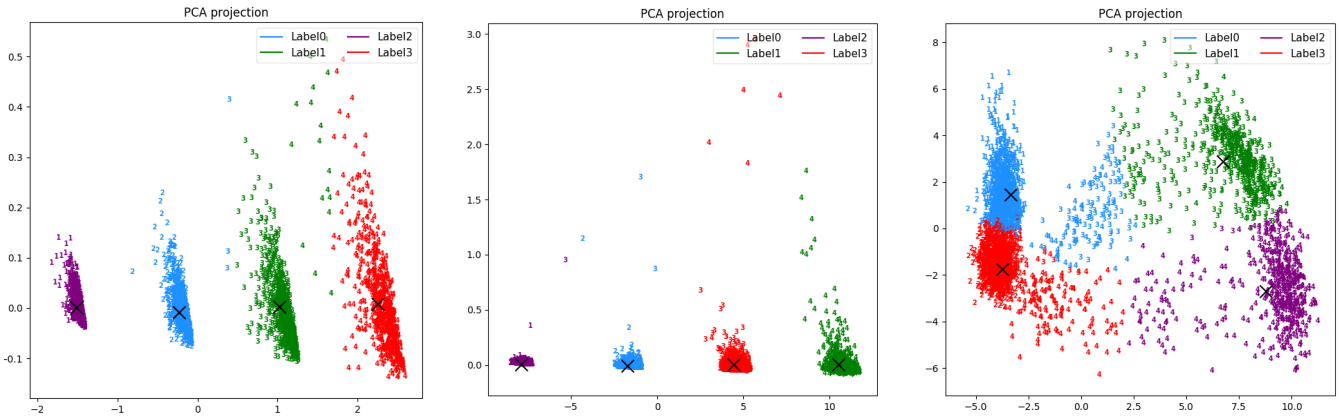
TABLE IV: Parameters employed for the generation of vehicles in the hard scenario: a is acceleration, d is deceleration, sf is the speed factor.

above 0.7, with slightly larger values of m with respect to the easy scenario. As in the easy scenario, switching to a descending ordering of the traces has lead to the best overall performance, and again $m = 5$ provided the best results. Differently from the previous setting, the acceleration in this case is ineffective in identifying the driving styles (since it is the speed factor that actually differentiates vehicles), whereas using speed or pairs speed/acceleration results in effective clustering. Interestingly, unsorted time series suffer somehow less of this problem, probably indicating that the *trend* in the variation of the acceleration contains information that can be exploited to characterize the driver's behaviour. Results with aggregate data again indicate an almost perfect clustering. The best results obtained for this scenario ($m = 5$, descending ordering, speed traces) are shown in Figure 3b.

C. Hard scenario

The third scenario is the hardest for the clustering algorithm. Here, some vehicles have the same physical properties but different aggressiveness, whilst other have the same aggressiveness but different physical properties. Therefore, driving styles will be similar even among different categories, which makes the clustering more challenging. The chosen parameters for SUMO are shown in Table IV. The simulation covers 3,604 vehicles, 29.4% of class 1, 31.0% of class 2, 19.8% of class 3, 19.8% of class 4.

As shown in Table IV, this scenario is much more challenging than the previous ones, and results are very different. In this case, in fact, the best overall results were obtained when the data is left unsorted. The best v-measure achieved is 0.733 for $m = 20$ and time-series that combine speed and acceleration. These results can be explained with the observation that, in a complex



(a) (b) (c)

Fig. 3: Results on the three scenarios: easy (a), intermediate (b) and hard (c). Colors indicate the cluster assignment by k-means, whereas digits represent the true vehicle label. Black crosses indicate cluster centroids.

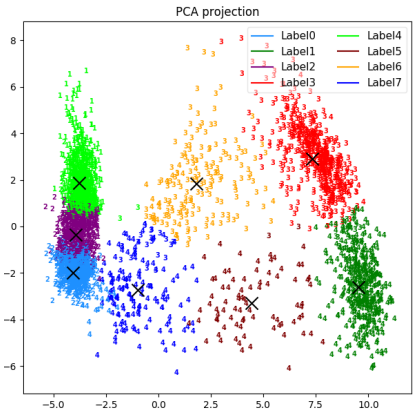


Fig. 4: Results on the hard scenarios obtained with 8 clusters. Colors indicate the cluster assignment by k-means, whereas digits represent the true vehicle label. Black crosses indicate cluster centroids.

scenario, the largest speed and the corresponding acceleration values are not sufficient to identify the driving style. It is instead much more informative to observe the *trend* in the time-series, which allows to consider, for example, how long it takes, for a driver, to accelerate and decelerate up to a certain speed. This information is clearly encoded in the unsorted time-series, but not in the ordered case. Table IIc reports the clustering metrics employed with the different settings, and Figure 3c shows the clustering after PCA projection.

For this scenario, in Table III we also report results for different numbers of clusters, obtained with unsorted traces with both speed and acceleration information. As the number of clusters grows, homogeneity increases, which means that the algorithms finds more clusters, but with stronger intra-cluster similarities: as a trade-off, not surprisingly, completeness decreases, as vehicles of the same class are sometimes split across different clusters. For example, Figure 4 shows that the bottom blue, brown and green clusters cover almost completely the fourth class of vehicles.

D. Discussion

The controlled experiments conducted with the SUMO simulator confirm that the proposed clustering approach could be effectively used to extract common driving behaviors from traffic traces. Clearly, aggregated statistics work extremely well, but this can happen only when there is a strong correlation between maximum values of speed and acceleration and driving style. Although such features are certainly crucial to detect common relevant patterns, they are not necessarily the only significant information in all real scenarios. Using decreasing ordering for time-series, and exploiting short feature vectors, typically leads to very good performance when the driving styles to be recognized are well distinguishable. On the other hand, it is very interesting to notice that unsorted traffic traces lead to very similar performance, in terms of clustering metrics, in all the three scenarios, thus showing to be robust across different, heterogeneous settings.

RELATED WORK

The problem of identifying common driving styles has been the subject of several studies, although without considering its integration within the context of micro-simulation tuning and optimization. Data mining techniques, including clustering, are employed in (Constantinescu et al., 2010) to identify common driving behaviors, by constructing aggregated statistics from a collection of GPS data. Supervised learning techniques for the classification of data coming from inertial sensors are presented in (Van Ly et al., 2013). In (Wang and Lukic, 2011) a review of the most widely employed features for driver characterization is presented, with the aim to build support systems for hybrid electric vehicle control strategy. Typical and aggressive driving style behaviors are classified in (Johnson and Trivedi, 2011), by exploiting smartphones as sensor platforms, and by employing a simple nearest neighbor classifier.

CONCLUSIONS

In this paper we presented a methodology for driving style characterization, that could be exploited for a more realistic design and calibration of micro-traffic simulators. The experimental results that we conducted in a controlled environment suggest that driving styles are best identified from the extreme behaviours of a driver, namely from the top-m largest values in the speed and acceleration time-series, but only when the vehicle categories are characterized by marked differences. Within this setting, we observed that speed values are typically more informative for the characterization of a driver. On the other hand, in more complex scenarios, it results to be more useful to consider a portion of the unsorted traffic trace, which allows to observe also the trend in the speed and acceleration time-series. Among future research directions, we are currently exploring the use of unsupervised deep networks, namely Stacked Denoising Autoencoders (Vincent et al., 2010), to perform automatic feature extraction from traffic time-series. More clustering algorithms will also be tested, and their performance will be evaluated as a function of the parameter representing the expected number of clusters, as we already started investigating in the reported experiments. Dynamic time warping (Berndt and Clifford, 1994) could also be an alternative for the comparison of time series with different lengths. It would be interesting also to assess the performance of the approach when vehicle categories are extremely imbalanced (i.e., few vehicles for some of the categories). Finally, our ultimate goal would be to use our system with real traffic traces.

REFERENCES

Arthur, D. and S. Vassilvitskii (2007). "k-means++: The advantages of careful seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 1027–1035.

Berndt, D. J. and J. Clifford (1994). "Using dynamic time warping to find patterns in time series." In: *KDD workshop*. Vol. 10. 16. Seattle, WA, pp. 359–370.

Constantinescu, Z., C. Marinoiu, and M. Vladoiu (2010). "Driving style analysis using data mining techniques". In: *International Journal of Computers Communications & Control* 5.5, pp. 654–663.

Hartigan, J. A. and M. A. Wong (1979). "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1, pp. 100–108.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc.

Helbing, D. et al. (2002). "Micro-and macro-simulation of freeway traffic". In: *Mathematical and computer modelling* 35.5-6, pp. 517–547.

Johnson, D. A. and M. M. Trivedi (2011). "Driving style recognition using a smartphone as a sensor platform". In: *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, pp. 1609–1615.

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

Krajzewicz, D. et al. (2012). "Recent Development and Applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.3&4, pp. 128–138.

Van Ly, M., S. Martin, and M. M. Trivedi (2013). "Driver classification and driving style recognition using inertial sensors". In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, pp. 1040–1045.

Vincent, P. et al. (2010). "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". In: *Journal of Machine Learning Research* 11.Dec, pp. 3371–3408.

Wang, R. and S. M. Lukic (2011). "Review of driving conditions prediction and driving style recognition based control algorithms for hybrid electric vehicles". In: *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE*. IEEE, pp. 1–7.

ALESSANDRO PETRARO is a software engineer at Cubbit, a digital startup accelerated by the investment fund Barcamper Ventures (PrimoMiglio SGR). His main interests are in the fields of artificial intelligence, machine learning and distributed networks. His e-mail address is: alessandro.petraro2@studio.unibo.it and his complete profile can be found at https://www.linkedin.com/in/alessandro- petraro-b9aa308b

FEDERICO CASELLI is research associate at Department of Computer Science and Engineering at the University of Bologna. His main research interest are in the fields of transportation systems, simulations, distributed networks and decision support systems.

MARCO LIPPI is assistant professor in Computer Engineering at the University of Modena and Reggio Emilia. His main research interests are in the fields of artificial intelligence and machine learning, with applications to bioinformatics, argumentation mining, transportation systems and computer vision. His e-mail address is: marco.lippi@unimore.it and his webpage can be found at http://www.agentgroup.unimore.it/Lippi.

MICHELA MILANO is full professor of Artificial Intelligence at the Department of Computer Science and Engineering at the University of Bologna. Her main research interest cover decision support systems and their integration with machine learning algorithms. She is Editor in Chief of the International Journal Constraints, member of the EurAI board, Executive Counsilor of AAAI, author of more than 130 papers on peer reviewed journals and conferences and coordinator of EU projects and industrial collaborations.