

C-ITS data completion to improve unsupervised driving profile detection

Brice Leblanc, Secil Ercan, Cyril De Runz

▶ **To cite this version:**

Brice Leblanc, Secil Ercan, Cyril De Runz. C-ITS data completion to improve unsupervised driving profile detection. 91st IEEE Vehicular Technology Conference (VTC2020-Spring), May 2020, Antwerp, Belgium. 10.1109/VTC2020-Spring48590.2020.9128731 . hal-02885746

HAL Id: hal-02885746
https://hal.science/hal-02885746

Submitted on 25 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

C-ITS data completion to improve unsupervised driving profile detection

Brice Leblanc, Secil Ercan and Cyril de Runz
CReSTIC, Université de Reims Champagne-Ardenne France
{brice.leblanc, secil.ercan}@univ-reims.fr,
BDTLN, LIFAT, University of Tours, France
cyril.derunz@univ-tours.fr

***Abstract*—Connected vehicles is a growing field of research that will produce great amount of data in near future. These data can be mined to generate traffic prediction, detect driver profile, find alternative route, *etc.*. This will help car manufacturers, road operators, telecom operators and other actors in the sector to improve road safety and drivers comfort. But nowadays few data are collected to create these tools.**

In this paper we compare different completion approach on data extracted from real experimentation on road to perform efficient driving profile detection. We analyze the deviations of the driver headings along a defined trajectory on specific Points of Interest (POI) to extract the driving profiles.

I. Introduction

The deployment of connected vehicles becomes a hot challenge since many years.Many interested actors (road operators, telecom operators, car manufacturers) need to explore data generated by this vehicles for purposes such as traffic prediction, driver profile detection, *etc.* We collected a data set from an experimentation of open road testing within the European project InterCor[1]. These data have been provided from different mobiles C-ITS (Cooperative Intelligent Transportation System) stations from more than 10 countries.

The main idea of this paper is to extract driving profile from real driving data. The collected messages contains the position, the speed and the heading of the vehicle. We focus on the heading variable in the message and gather headings over 22 Point Of Interest (POI) based on interesting road section (red light, stop, toll, event position, etc.). After a treatment step, we obtained a data set of 20 observations that covers every POI. Since it was not enough to properly perform clustering we searched for methods to obtain a larger data set. We use 2 different approaches explained in sections V and VI before the clustering has been performed on completed data. Moreover, as vehicular communication data will form Big Data in the future, the clustering methods used in this paper already have scalable version using Spark or MapReduce.

The rest of the paper is organized as follows. Section II gives an overview of related works. Section III introduces the pipeline from the data collection to the clustering. Section IV exposes the trajectory data acquisition process. Section V presents the clustering results before completion and a first attempt to increase the size of the data-set. Section VI presents the data completion approach used in order to obtain a larger set of complete trajectory data. Section VII evaluate the clustering quality with completed data in terms of driving profile extraction. Section VIII gives the conclusion and perspectives.

[1]Project number INEA/CEF/TRAN/M2015/1143833

II. Related Works

A. Driving Profile detection using mobile device data

In the literature, driver profile detection approaches generally use data produced by the GPS of a smartphone and not data issued from vehicles sensors. In [CE15], Castigani et al. developed a platform based on mobile device data (mainly GPS and motion sensor data) that detects risky driving event using fuzzy logic. Gonzalez et al. [RS14] used smartphone data to characterize the effect of aggressiveness on driving signals, using a linear filter, then detect aggressive or calm driving profile using Gaussian Mixture Model. Those two approaches are relying on the reliability of smartphone GPS data. The authors of [JT11] made a comparison of the difference between CAN bus data and smartphone data on calm/aggressive driving behavior. They concluded that using smartphone data is a viable option after a treatment phase using Dynamic Time Warping to suppress the noise in the data. During our collection phase, we also gathered data from smartphone but they do not contain the most important variable for profile detection on smartphone data which are, according to [JT11], x-axis rotation rate, y-axis acceleration and pitch. But it is hard to control the orientation of the smartphone during the whole collection process during real experimentation. Since we have direct access to connected vehicle communication data, we have a more reliable data set that does not needs noise treatment to be exploited.

B. Data clustering

The need for profile detection is growing for road operators, telecom operators, car manufacturers and also insurance company in order to gain a capacity of anticipation of the risk. This conducted us to study and use unsupervised clustering algorithm to detect driving profile.

Several methods can be used to perform clustering. The objective of this king of approach is to regroup data in order to maximize the similarity into each cluster and the pairwise dissimilarities between clusters. We focus in this

paper on well-known approaches that have scalable versions using Spark or MapReduce. We consider:

- K-Medoïds [KR87] algorithm select at random k (defined by user) data points, to consider them as a cluster, regroup the other points in the closest cluster until none are left. The most representative data point (centroid) in it is updated after a new point is added to the cluster. The K-Medoid approach has been efficiently deployed on large scale data [SH17].
- CLARA (**C**lustering **Lar**ge **A**pplication) is an optimization of the k-medoids to handle large data-set. It perform k-medoid on sub-set of the original data-set.
- K-Means algorithm is very similiar to the k-medoid algorithm, the sole difference is that the centroid is not a real data point but a fictive one (the mean of all the member of the cluster). A k-means Spark version is available in the official Spark ML library.
- Fuzzy Analysis Clustering (Fanny) [RK90] can have data points belonging to more than one cluster. Fanny groups data into k clusters according to the fuzzy memberships of each data to the different clusters. Several proposals exists in the literature in order to deploy the approach on MapReduce or Spark such as [Lud15].
- DBSCAN (Density-based Algorithm for discovering clusters in large spatial databases with noise) [EKSX96] is a fast algorithm with a strong resistance to outliers. It is very weak to cluster with varying density. [HC16] propose a Spark version of DBSCAN.
- AGNES (**Ag**glomerative **Nes**ting) and DIANA (**Di**visive **An**alysis) are two hierarchical clustering methods. For Agnes, at start every data point is a cluster then, at each step, the two closest clusters are merged together until all the points are in the same cluster. To obtain the final clusters, the agglomeration tree is cut when the number of cluster meet the number of classes. For Diana it work the the same way but upside-down (one cluster is splitted). Jin et al. [JC15] introduce a scalable hierarchical agglomerative clustering algorithm based on spark.

In order to compare clustering results, we use the purity index [MS10]. It is a measure of the extent to which clusters contain a single class. Its computation principles are: firstly, for each cluster, count the number of data points from the most common class in said cluster; secondly, take the sum over all clusters and divide it by the total number of data points. Formally, its equations is:

$$PI = \frac{1}{n} \sum_{q=1}^{k} max_{i \leq j \leq l}(n_q^j) \tag{1}$$

where n is the total number of samples and k is the number of samples in the cluster q, that belongs, to original class j ($1 \leq$ j $\leq$ l). The larger the value of purity is, the better the clustering performance is. The purer the clusters are, the better they represent one driving (sub)profile.

C. Data completion

Handling missing data is an important issue of data mining. This problem can be handled by ignoring, deletion and imputation (to replace with a value). For replacing missing values, there are several conventional approaches and new proposals. Lana et al. [LS18] handled the missing data of daily road traffic data. They proposed new approaches to impute the missing data and compared with conventional methods which are easy to implement such as basic imputation (assigning a constant value), linear interpolation, mean value, and 1-nearest neighbor (*1NN*) which takes the value of the closest data based on the similarity. However, they have some limits in different cases. Basic imputation does not produce good estimation since a constant not the reality. A larger length of gap which is the interval of successive missing data is a limit for interpolation method. If the variance of data is high, mean value method cannot yield good results. The lack of *1NN* method is that higher number of data may take more time to find the closest data.

Ranking method is another good way to handle missing value. Guo et al. proposed an improved rank method to complete traffic data comparing with mean-value method, mean of neighbors, and principal component analysis [GZ18]. Although their proposal gave good estimations, a conventional rank method has a restriction on non-numeric data.

Bayesian methods are also preferred to deal with the missing data problem [LT16]. It is usually hard to obtain the posterior probability which means the probability of missing data given the observed data and Bayesian network. There are several approaches to estimate the unknown parameters for this method. Bayesian approach is frequently used with Markov Chain Monte Carlo to avoid this problem [NI05].

For other approaches, a review can be found in Pratama et al. [PI16]. In this study for replacing missing values, we used the following approaches which are simple but still produce good results: interpolation, mean value, rank, similar case (nearest neighbor) and Bayesian approach.

III. General description of our approach

Our objective is to detect driving profiles without any a priori, this is why unsupervised learning approaches, i.e. clustering, are privileged. To perform it we use mobility data from vehicle collected during the InterCor TestFest. We used two mean of collect to obtain vehicles mobility data.

First we deployed a mobile application and used the GPS of the mobile device to capture the position, speed and heading of the vehicles. This approach is troublesome for profile detection because, as told previously, it relies a lot on the orientation of the device which we could not control during the test.

Second we asked every participant to record the network communication they received on the antenna used by their On Board Unit (the functioning of C-ITS communication will be described in part IV) in log file which can provide us very precise data about the vehicles movement. This approach also raised some problems that reduce the possibility of treatment: the vehicle could not register the messages they emitted. We

had to reconstruct the trajectories using other vehicles log files. This can only work if the vehicle is in range of communication of another vehicle, range that is varying according to weather condition and topography. In addition some participant did not provide log files due to technical problem.

These problems lead to incomplete data. Therefore we construct two scenarios illustrated in Fig.1:
In the first one a clustering is made on the complete trajectory data, then each incomplete data is associated to its closest cluster according to a distance matrix. The second one consists in a completion of a maximum trajectory before the clustering.

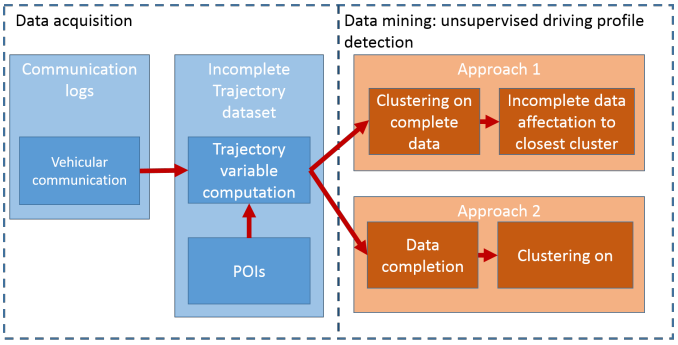


Fig. 1: General pipeline for unsupervised driving profile detection using vehicular communication data

IV. Data Acquisition

The C-ITS data used in this paper are under a protocol stack defined and standardized by the ETSI[2].

The vehicle sends messages to give dynamic information about the vehicle (i.e. position, speed, heading, etc.) to its neighborhood using WiFi IEEE 802.11p (denoted also ETSI ITS-G5). Depending on the speed of the vehicle, the frequency of messages varies from 1 to 10 message per second.

Road Side Units (RSU) can be found along the road to collected all received from vehicles in order to run road operator's computations such as traffic management, event recording. This general architecture is presented on Fig. 2.

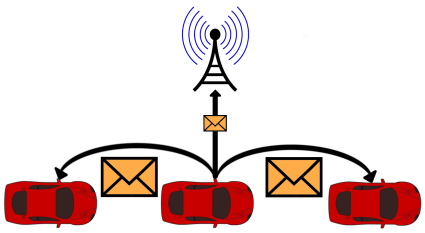


Fig. 2: General communication schemes in C-ITS

Based on the track followed by the vehicles during the test, we selected 22 POI in location meaningful to estimate a driver behavior (red light, stop, toll, event position, etc.). On these POI, we regrouped the data on a 100 meters section and a 10 seconds time fork to have a view on the behavior of the driver on each POI. The Fig. 3 depicts a representation of a POI and a vehicle passing by the POI. The opacity of the vehicle is illustrating the time the vehicle is staying around the POI. If the vehicle stays stopped near the POI for a long time, the previous data is not considered. And if the vehicle is passing without stopping at the POI (e.g. the light is green), the distance will limit the data capture.

We could now perform clustering on these data in attempt to detect driving profile. To do so we focused on the heading information. We estimate that a vehicle with a heading diverging significantly from the mean/median heading can be considered as having a more aggressive behavior. This is why we created 4 variables for each POI base on the heading: mean heading, median heading, maximum heading, minimum heading.



Fig. 3: Data collection using position and time around a POI

Consequentially to the different potential problem in the data described previously, after the treatment we obtain 20 complete observations (a complete observation contains values for the 22 POI) which is not a lot. Also, many vehicles had only one observation, that is too few to perform correct clustering. Only data with at least half the data filled will be considered on the following.

According to our detection process pipeline, we propose two approaches (scenarios). The next section is devoted to the first one: firstly do a clustering on complete trajectory data, secondly put each incomplete data to the closest cluster.

V. Approach 1 : Clustering on full value and find representative for missing values

Consequentially to the different potential problem in the data described previously, after the treatment we obtain too few complete observations (a complete observation contain all 22 POI) to perform a realistic clustering.

According to the first scenario/approach presented in Fig. 1, the different clustering approaches described in section II-B were used on the 20 complete trajectory observations.

Clustering results, in terms of purity index, on 13 classes from 20 complete observations are of 0.95 for K-Means, K-Medoïds, AGNES, DIANA, CLARA and 0.15 for DBSCAN. Please note that the small size of the data set may not allow to correctly run the Fanny algorithm. Since the result of DBSCAN is very low (here and in every experiments performed), we removed it from the global result table I.

[2]European Telecommunications Standards Institute : http://www.etsi.org

The second step is to associate each incomplete data to its closest cluster. To do so we used 2 different methods both based on euclidean distance and distance matrix between an observation and the medoids/centroids.

In the first method, we calculate the global distance for each incomplete observation to each cluster of complete data. The observation is added to the cluster with the global shortest distance to the observation.

In the second method, the distance is calculate for each group of 4 variable of a POI. Then the the cluster with the shortest distance to most of the POI variables is elected and the observation is then assigned it.

After using these approaches we obtain 54 percent of clustering purity for the first approach and 46 percent of clustering purity for the two others (see result table I). Since the result is not satisfying we looked for a new approach and tried to computationally complete the missing values from the data before performing the clustering and then cluster the new data set formed using complete and computationally completed trajectory data.

VI. Approach 2 : Data completion

Since the complete data we have are few, the missing data will be completed by proper procedures. In this study, some approaches to replace missing values will be compared and their performance evaluated using two indicators: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Since the data are missing at random in our problem, we can use general imputation approaches instead of special approximations for not random missing data.

For interpolation of mean values method, firstly the mean values of each variable, then the mean of these mean values are calculated. In addition, the mean values of each data which have missing values are calculated in order to see the behavior of the incomplete data. To complete a missing value, three mean values are considered for the interpolation: the mean value of the related variable, the mean of all mean values of variables, and the mean value of the related observation. The missing value is replaced with the mean value of the related variable from the training data set in mean value approach.

In ranking method used in this paper, for each incomplete observation, each variable of the incomplete data set is merged with the variables of every observation of the training data set, sorted and given a rank. Then, for each incomplete variable of the observation, the most common rank for the other variables of this observation is selected to replace the incomplete data.

In similar case approach the closest data is selected based on the similarity. The similarity measurement is the euclidean distance in general. When the closest data is chosen according to the distance, the related value of the closest data is assigned to the missing value.

Bayesian approach allows to replace missing values based on probabilities. The probability density function (PDF) of observed values and the PDF of the related variable are calculated. By the probability of each observed value of the incomplete data, an average probability is calculated for each missing value. Then the missing value is estimated by using this probability in the PDF of the related variable. In this study, the observed data are assumed to have normal distribution for simplification. Let Y_{obs} and Y_{mis} denote the observed part and the missing part of Y. Eq. 2 gives a general expression of Bayesian approach for missing data imputation.

$$P(Y_{mis} \mid Y_{obs}) = \frac{P(Y_{obs} \mid Y_{mis})P(Y_{mis})}{P(Y_{obs})} \tag{2}$$

For the implementation, 10% of the complete data will be considered as test data set while the remaining is used as training data set. The replacing methods will be applied on these data sets for different missing rates, from 5% up to 60%. We will randomly remove some values based on the missing rates from the test data set and use the training data set to replace the missing values. For each method and each missing ratio, the algorithm is executed 100 times to avoid deviation caused by randomness. Finally, the performance indicators for each method are compared.

The methods explained will be compared by performance indicators based on errors between observed and predicted data. RMSE is defined as the square root of the sum of the mean square value of the difference between observed data and predicted data. MAE is the mean absolute value of the difference between observed data and predicted data.

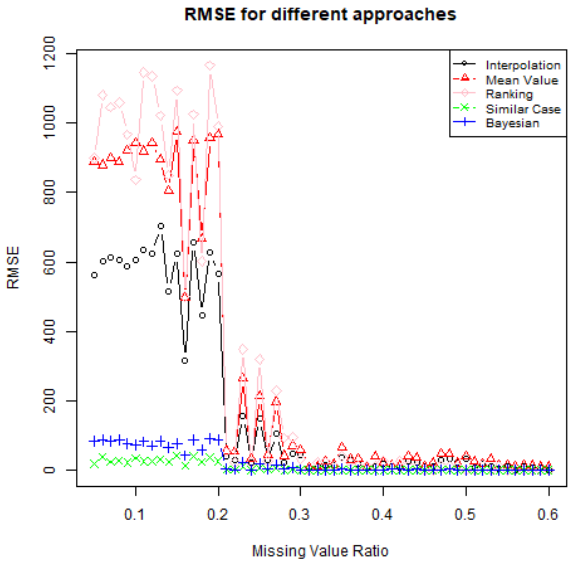


Fig. 4: RMSE for different approaches

Fig. 4 and Fig. 5 represent the RMSE and MAE values, respectively, for different imputation approaches. Both of the figures show that similar case and Bayesian approach give better results although similar case is better than Bayesian approach for smaller missing rates.

To replace missing values for the incomplete data set, we select Similar case and Bayesian approach due to the low errors. Since these approaches give good results for 60%

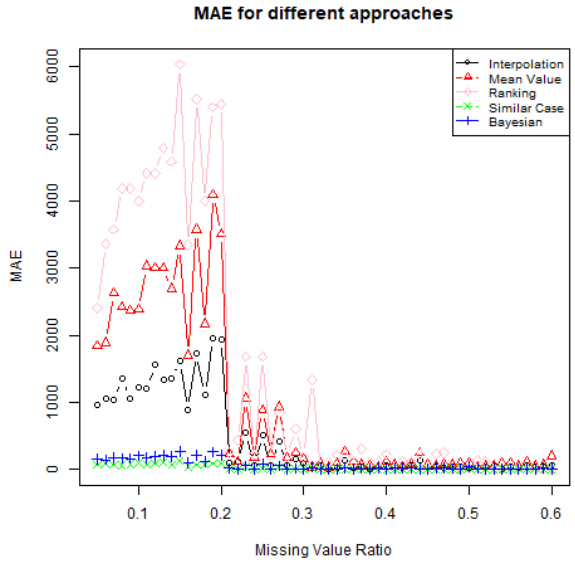


Fig. 5: MAE for different approaches

missing ratio, we will use the incomplete data set which has at least 40% data.

VII. Clustering using complete and completed trajectory data

We can now use the data sets issued of completion step to perform the same set of clustering methods as those used in section V. Results are also presented in table I.

Clustering methods are performed 19 classes from 59 completed observations on the two different data sets. In both cases, the fuzzy clustering (Fanny) is given the better results (around 73%, 74%) which are quite important for unsupervised approaches. Each extracted cluster is mainly representing the same class, therefore the same real driving profile. Thus, using C-ITS communication data, without taking into consideration privacy data such as car information (static mac address, name of the car, etc.), we can detect trajectory that have been produced by the same driver then the information is characterizing its driving profile.

	without completion		with completion	
Algorithm	method 1	method 2	Bayesian	Most Similar Case
K-Means	0.23	0.08	0.723	0.718
K-Medoïds	0.54	0.46	0.678	0.713
AGNES	0.23	0.31	0.645	0.662
DIANA	0.23	0.31	0.662	0.662
CLARA	0.54	0.46	0.679	0.713
Fanny			0.730	0.747

TABLE I: Table of purity

VIII. Conclusion

In this paper, our objective was to detect driving profiles from C-ITS communication data. In order to do so, we proposed a new data processing pipeline which goes from trajectory data construction, to their completion and clustering. The trajectory data are built using vehicular communication logs and Point of Interests. Better results are obtained when incomplete data were computationally completed. According to our experiments, the combination of using the most similar case for the data completion, and a fuzzy clustering gives the best results. The most important result is to see that using our methodology, we can extract profiles of drivers that are significant without taking into consideration privacy data. Even though our approaches were not deployed on real big data infrastructure (due to a volume of data being too small), the methodology we proposed and the methods we used are fully big data infrastructure compliant.

In our future works, we will study if it is also possible to characterize a driver directly from log analysis using stream data clustering.

References

[CE15] Frank Castignani, Derrmann and Engel. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *IEEE Intelligent Transportation Systems Magazine*, 7(1):91–102, 2015.

[EKSX96] Ester, Kriegel, Sander, and Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[GZ18] Wang Guo, Wang and Zhang. An improved low rank matrix completion method for traffic data. In *2018 11th ICICTA*, pages 255–260, Sep. 2018.

[HC16] Liao Han, Agrawal and Choudhary. A novel scalable dbscan algorithm with spark. In *2016 IEEE IPDPSW*, pages 1393–1402. IEEE, 2016.

[JC15] Chen Hendrix Agrawal Jin, Liu and Choudhary. A scalable hierarchical clustering algorithm using spark. In *2015 IEEE First International Conference on Big Data Computing Service and Applications*, pages 418–426. IEEE, 2015.

[JT11] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Transportation Systems (ITSC)*, pages 1609–1615. IEEE, 2011.

[KR87] Kaufman and Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1 Norm and Related Methods*, pages 405–416. North-Holland; Amsterdam, 1987.

[LS18] Vélez Laña, Olabarrieta and Del Ser. On the imputation of missing data for road traffic forecasting: New insights and novel techniques. *Transportation Research Part C: Emerging Technologies*, 90:18–33, 2018.

[LT16] Bo He J Elm Luo, B Lawson and C Tilley. Bayesian multiple imputation for missing multivariate longitudinal data from a parkinson's disease clinical trial. *Statistical Methods in Medical Research*, 25(2):821–837, 2016. PMID: 23242384.

[Lud15] Ludwig. Mapreduce-based fuzzy c-means clustering algorithm: implementation and scalability. *International journal of machine learning and cybernetics*, 6(6):923–934, 2015.

[MS10] Raghavan Manning and Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[NI05] Ni and D. Leonard II. Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data. *Transportation Research Record*, 1935(1):57–67, 2005.

[PI16] Ardiyanto Pratama, Permanasari and Indrayani. A review of missing values handling methods on time-series data. In *2016 ICITSI*, pages 1–6, Oct 2016.

[RK90] Rousseeuw and Kaufman. Finding groups in data. *Hoboken: Wiley Online Library*, 1990.

[RS14] Vinagre Díaz Rodríguez González, Wilby and Sánchez Ávila. Modeling and detecting aggressiveness from driving signals. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1419–1428, Aug 2014.

[SH17] Lee Song and Han. Pamae: Parallel k-medoids clustering with high accuracy and efficiency. In *Proceedings of the 23rd ACM SIGKDD DMKD*, pages 1087–1096. ACM, 2017.