Unsupervised Classification of Routes and Plates from the *Trap2017* Dataset

Massimo Bernaschi
Institute for Applied Computing (IAC-CNR),
Via dei Taurini 19, Rome, Italy
m.bernaschi@iac.cnr.it

Alessandro Celestini
Institute for Applied Computing (IAC-CNR),
Via dei Taurini 19, Rome, Italy
a.celestini@iac.cnr.it

Stefano Guarino
Institute for Applied Computing (IAC-CNR),
Via dei Taurini 19, Rome, Italy
s.guarino@iac.cnr.it

Flavio Lombardi
Institute for Applied Computing (IAC-CNR),
Via dei Taurini 19, Rome, Italy
f.lombardi@iac.cnr.it

Enrico Mastrostefano
Institute for Applied Computing (IAC-CNR),
Via dei Taurini 19, Rome, Italy
e.mastrostefano@iac.cnr.it

Abstract

This paper describes the efforts, pitfalls, and successes of applying unsupervised classification techniques to analyze the Trap2017 dataset. Guided by the informative perspective on the nature of the dataset obtained through a set of specifically-written perl/bash scripts, we devised an automated clustering tool implemented in python upon openly-available scientific libraries. By applying our tool on the original raw data it is possibile to infer a set of trending behaviors for vehicles travelling over a route, yielding an instrument to classify both routes and plates. Our results show that addressing the main goal of the Trap2017 initiative (*"to identify itineraries that could imply a criminal intent"*) is feasible even in the presence of an unlabelled and noisy dataset, provided that the unique characteristics of the problem are carefully considered. Albeit several optimizations for the tool are still under investigation, we believe that it may already pave the way to further research on the extraction of high-level travelling behaviors from gates transit records.

1 Introduction

The advances and widespread availability of automatic Number Plate Reading Systems (NPRS) allow the collection of large amounts of traffic data [1]. For law enforcement agencies, this raises the problem of finding convenient techniques and tools to analyze such data, in order to find meaningful traffic patterns and identifying anomalous and criminal behaviors [2]. As a matter of fact, analyzing large amounts of traffic data is challenging due to the huge size of the dataset and the complexity of traffic dynamics. As such, developing an effective and scalable automatic traffic analysis system that can detect, track and gain useful insights about the behavior of road-users is vital to law enforcement agencies.

To support researchers willing to contribute to this difficult yet critical task, the Italian National Police (INP) made available a sample of data collected through automatic NPRS. The dataset con-

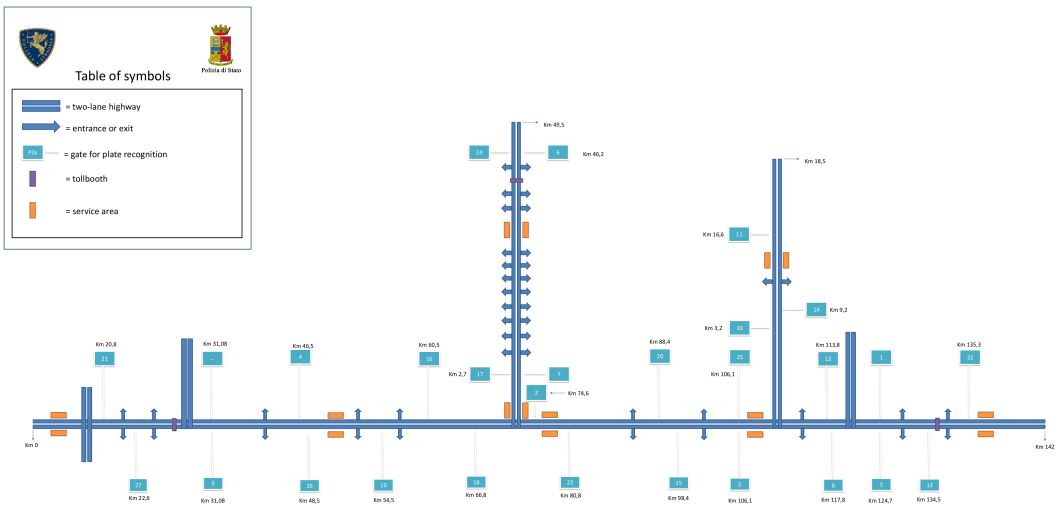


Figure 1: The highway section under study

sists of 365 Comma Separated Values (CSV) files[1], each containing as many lines as the number of events registered by the 27 gates distributed along the highway section under study. Each line, in turn, contains the following fields: `plate; gate; lane; timestamp; nationality (of the plate)`. Plates have been anonymized, mapping them to a set of consecutive positive integers. A sample of the data is shown below:

```
492788;20;1.0;2016-06-27 08:41:14;CH
144843;5;1.0;2016-06-27 12:38:13;I
144843;16;1.0;2016-06-27 09:01:30;I
7147369;4;2.0;2016-06-27 11:46:05;CH
```

Figure 1 shows the map provided along with the dataset. The map shows the highway section under study, where distances in Km between adjacent gates have been respected, but everything else has also been anonymized.

Hereafter, we describe the activities we carried out in order to analyze the sample of traffic data provided by the INP. Specifically, we present: (i) a set of exploratory bash/perl scripts, useful to compute descriptive statistics of the dataset and to highlight a few pitfalls it hides; (ii) a thorough python tool which, based on popular openly-available scientific libraries (numpy[2], scipy[3], mathplotlib[4], sklearn[5]), allows classifying both travel routes and plates. Remarkably, we used an unsupervised approach not relying on any *a priori* knowledge about the nature of the dataset or possible/interesting behaviors of road-users. Our findings show that relevant information can be extracted from similar transit records datasets despite the limitations introduced by the inevitable flaws of existing automatic NPRS. However, the analysis also suggests that directly applying standard statistical data mining techniques can hardly provide the desired insight into the intents of road-users. The unique characteristics of the problem require clustering algorithms to be composed and tailored based on a "measurable" definition of behavior of a vehicle. Simply guided by common sense, we therefore propose such a definition and highlight a few promising design choices (*e.g.*, related to possible data filtering/cleaning strategies), which pave the way for future research in the area. Among the benefits of our approach, our classification mechanism provides significant information about the obtained clusters that can be rightfully expected to support law enforcement agencies in understanding/labelling the behavior of a road-user. In addition, our approach is inherently robust towards quality issues that are unavoidable with real datasets.

[1]The dataset consists of a file per day for the whole year 2016, but the file corresponding to October 7^{th} is missing. As a consequence, the total number of files is 365 despite 2016 being a leap year.

[2]http://www.numpy.org/

[3]https://www.scipy.org/

[4]https://matplotlib.org/

[5]http://scikit-learn.org/stable/

1.0.1 Roadmap.

The rest of the paper is organized as follows: Section 2 reports a high-level statistical analysis of the dataset; Section 3 details the design of our classification/clustering tool; in Section 4 we summarize our findings; Section 5 discusses state-of-the-art approaches to traffic monitoring and analysis; finally, Section 6, draws conclusions and suggests possible directions for future work.

2 Statistical Analysis

We start with an overview of the information extracted from the Trap2017 dataset by means of our bash/perl scripts [6]. Let us highlight that the procedures to extract information are pretty simple (few lines of scripting languages) but fully automatic. We do not discard *a priori* any data, including those that would appear clearly suspicious, for the simple reason that we do not look at the data but we directly use them as input to the scripts. Further, discarding data could prevent malicious behaviour from being discovered at a later analysis stage. This choice is motivated by the idea that our approach has to be scalable to much larger datasets where direct inspection is out-of-question.

A first statistics concerns the number of transits *per* plate. In fact, about 52% of the plates appear just once in the dataset. The total number of plates in the dataset is 14351059, whereas those for which we have records of at least two transits (*i.e.*, at least one travel) are "only" 6958429. This immediately suggests that considering all available transits is not necessarily a good idea, since it means keeping in a lot of data that are not really informative of the behavior of any plate. Along this line, in view of extracting individual statistics for each plate we selected a relevant subset of the plates (26000), i.e. those that have more associated events. For each of them, we performed several analyses including, but not limited to: (i) collecting all events for that plate; (ii) identifying trips for each plate, where a trip is a sequence of transits separated by one hour at most; (iii) finding the speed for each pair of transits for each plate.

By looking at the extracted information (*e.g.*, velocities or number of transits) it is apparent that there are either several unexpected findings or many errors in the data (or both...). For instance, there are 103897 events for the plate `257`. This is a huge number corresponding to almost 300 passages *per* day. This is one of the anomalies that can be fed to domain experts.. There are, obviously, possible alternative explanations of the phenomenon, such as: (i) the plate has been cloned and there is more than one vehicle using the same plate; (ii) the system that recognizes the plates may be wrong; (iii) the anonymization system may be wrong. Unfortunately, understanding what is the most reasonable explanation is not trivial. We cannot exclude that the *same* problems are present also for plates that have much smaller numbers of events. Despite the simplicity of the dataset it appears difficult to tell apart good and bad data.

Other significant issues emerge analyzing the correlation between two or among more than two events. We defined a trip as a sequence of two or more events in which two transits are separated by less than 3600 seconds. The motivation is that we aimed at identifying events that provided a reasonable proof that the considered vehicle had left the highway, so as to be able to describe the behavior of a vehicle in terms of its individual journeys, instead of considering its whole transits sequence altogether. Unfortunately, some of the journeys defined according to this convention are very strange (in a way that does *not* depend on our discretionary definition of trip). Here we mention just two of the anomalies we found by simply browsing the outputs of our scripts:

- Unrealistic velocities, such as velocities well beyond 1000 Km/h.
- Incoherent sequences of gates passages, such as several cases in which there is a transit through two gates that are not consecutive with no passage through the intermediate gate(s), even in sectors with no intermediate exits (*e.g.*, transits through gates 2 and 6 with no passage through gate 7).

These kinds of anomalies also admit several possible explanations: (i) as before, these events may not belong, actually, to the same plate; (ii) the map that describes the highway sector may contain errors; (iii) our definition of journey could be deeply wrong (and any combination of them). In

[6] Details about the scripts and their output format can be found online `http://twin.iac.rm.cnr.it/manuale.tbz`

general, without detailed information about how the dataset was generated, making any assumption on what to expect seems hazardous.

In fact, our approach allows providing feedback to domain experts, even in a very early phase of the analysis. Nevertheless, additional findings (especially related to anomaly detection) to be fed to domain experts are discussed below.

3 Design of a Plates Behavior Classifier

As motivated in the Introduction, the challenge with the Trap2017 dataset is to develop (semi-)automated classification systems able to identify road-users behaving in an unusual and potentially criminal way. In this Section we present a plates behavior classifier based on unsupervised clustering and accompanied by a cluster characterization to be used for understanding and labeling the clusters. We start with an overview of the underlying logic of the proposed tool, followed by a step-by-step description of its functioning.

3.1 Overview

Despite the INP suggested a few examples of recognizable suspect behaviors (cloned plates causing space-time inconsistent transits; habitual criminals visiting many service areas to select a victim), defining a complete model of malicious transit patterns is prohibitive. Unsupervised clustering is the only possible approach to prevent that unspecified and unpredictable/novel criminal behaviors are incorrectly classified, or not even recognized as malicious. However, rendering the final classification informative to police officers/technicians is vital in order to allow them to correctly interpret and label the obtained clusters.

Aiming at a precise, comprehensible and comparable/measurable definition of behavior of a plate, we introduce the concept of "route" to denote *any* possible pair of gates, and we move the focus of the analysis from gates transit records to *routes travel times*. We set aside rigid sequences of time-space coordinates in favour of a more flexible representation that enables inter-plate comparisons (*e.g.*, to tell apart plates that visited a specific service area) while still allowing for the detection of inconsistencies. We do not limit the definition to pairs composed of two gates which are adjacent on the highway because, as emerged in Section 2, the dataset contains too many cases of consecutive close-in-time transits of a plate at two non-adjacent gates, which cannot be left out of the analysis. Since gates readings are not 100% reliable, even understanding when a plate exited or entered the highway is problematic. We therefore follow the sounder approach of using no filter and feeding our classifier with all available data. Eventually, the classifier will turn out to be able to do the job for us, demonstrating an excellent accuracy in detecting features of a route such as whether it is delimited by two adjacent gates.

Finally, as we deal with more complex tasks comparing what happened at different times may not be enough, and we may as well need to compare what happened at different (but somewhat matching) routes. For instance, identifying which plates stop "too often" at service areas requires elaborating on the travel times of different plates along all routes that contain a service area. We therefore need our plate behavior representation to embed a suitable classification of travel routes. The idea is to first rely on global travel times statistics to automatically classify all possible routes, and then cluster plates according to how their individual statistics fit the global statistics for each class of routes. Exactly as for plates, we discard any *a priori* classification of routes for at least two reasons: it may not work properly with an intrinsically flawed dataset, and it would affect the flexibility and scalability of the proposed solution.

Let us acknowledge that for the moment we set aside both the *lane* and *nationality* fields of each record, other than the time and date in which an event took place. Although we provided a few hints to motivate our choice, we do recognize that these data may have a relevance, and we expect using labels such as "preferred lane", "nationality", or "time and date" to be a potential boost to the performance of our clustering algorithms. Yet, investigating this option is left to future work.

3.2 The Tool

From the user viewpoint, our plates classification tool consists in a python script fed by a text file. Each line of the input file describes a different plate using the following syntax: $G_1\ T_1\ G_2\ T_2\ G_3\ \ldots\ G_n$, *i.e.*, a sequence of gates $G_i, \ldots, G_n$ separated by the corresponding travel times $T_i, \ldots, T_{n-1}$. For $1 \leq i < n$, T_i is the travel time between gate G_i and gate G_{i+1}, expressed in milliseconds, and this syntax tells us that the first record involving that plate reports a transit at gate G_1, the second one reports a transit at gate G_2 exactly T_1 milliseconds after, and so on. Details about the usage of the tool and of all most relevant options can be found online[7].

3.2.1 Step 0: Data Structures and Preprocessing.

As mentioned before, our plates clustering algorithm relies on a suitable classification of routes, in turn based on travel times statistics. The step 0 of our tool is therefore the definition of a data structure which allows easy access to all travel times measured over each possible route, *i.e.*, pair of gates. This data structure is created every time the tool is executed. Prior to compute the routes clustering, the tool allows applying a few transformations to the data, such as:

- Apply the same function to all travel times. For instance, applying a logarithmic transformation to all data may be used to make the distance between a few seconds and a few minutes comparable to the distance between a few minutes and a few hours.
- Normalizing all travel times relative to a specific route, dividing them by either their median or their mean. This is mostly useful to force easily comparable values for different travels, which may be critical for the correctness of the customized metrics we introduce later.
- Removing all times larger than a specific threshold. This threshold may be useful to prevent that pathological cases (such as broken down vehicle or cars that surely took an exit) poison data.

3.2.2 Step 1: Routes Clustering.

The underlying idea for classifying the routes is that the distribution of travel times for a route can be rightfully expected to exhibit a few modes, corresponding to as many possible trends for drivers travelling along that route. For instance, looking at the distribution of travel times for a route we expect to find a peak at the expected travel time when the route is free, another peak at the expected travel time when the route is congested, and another peak at the expected travel time of drivers stopping at a service area (if there is at least one along that route). Comparing the coordinates of these peaks, we can estimate the similarity between two routes. Practically, this is done by applying Gaussian Mixture Modeling (GMM) [3], extending the aforementioned data structure by associating to each route its inferred Gaussian Mixture (GM). The distance between any two GMs is then established relying on a combination of the Earth Mover Distance (EMD) [4] and the Kullback-Leibler (KL) divergence [5]: KL is used to estimate the distance between any possible pair of Gaussians taken from the two mixtures (KL has a closed form for Gaussians), and these distances are fed to EMD together with the weights of the GMs. In short, our custom distance measures the cost of transforming the two GMs in one another, with the cost of transforming two Gaussians being their KL divergence. Once we have a distance matrix for all routes, we apply Hierarchical Agglomerative Clustering (HAC) [6] to find the desired routes classification. Let us remark that each class[8] can be described by its mean GM, which summarizes the expected travel time statistics of each member of that class.

3.2.3 Step 2: Plates Clustering.

Let N be the number of routes clusters and M be the number of Gaussians in the GM modeling the travel times statistics of each route. Plates are classified using K-means clustering, associating a $N \times M$ matrix to each plate and using the norm of the difference of their matrices as the distance between two plates. The i, j element of the matrix associated to a plate is the fraction of data for that plate which refers to a route belonging to cluster i and that are assumed to have been generated by

[7] see `http://twin.iac.rm.cnr.it/manuale.tbz`

[8] To avoid excessive repetitions we will interchangeably use "cluster" and "class" (sometimes even "type") to denote the partitions obtained using our classifier.

the Gaussian j associated to that route. In other words, the matrix associated to a plate summarizes the available information about the behavior of that plate in terms of a travel time distribution per cluster, and the more two plates have a similar behavior the closer their matrices are expected to be. In order to have comparable matrices we need M and N to be fixed for all routes. Additionally, having a fixed M is consistent with the use of an EMD-based metrics. We experimentally found $M = N = 5$ to be a good choice, while the optimal parameter K is found at each run relying on a silhouette score.

4 Our Findings

The main outputs produced at each run of our tool can be summarized as follows:

- For each possible pair of gates, which we call a *route*, the tool identifies the Gaussian Mixture (GM) that best fits the measured travel times on that route. Being, *de facto*, a generative model for transits over that route, this GM is a synthetic yet descriptive representation of behaviors on that route.
- The tool produces a classification of routes into clusters based on their GMs. This classification is accompanied by an aggregate GM which tells us what are the communal/distinctive features of that set of routes.
- Finally, the tool clusters plates based on their behavior, defined as the distribution over the Gaussians of each of the identified classes of routes. This behavior is also combined on a *per*-class level to obtain a single cumulative distribution describing the characteristics of each class.

4.1 Tuning the Classifier

Despite the ultimate goal of our tool is to classify plates, in order to measure the quality of our tool it is fundamental to also evaluated the results of the routes classifier. Indeed, while we do not have any prior knowledge about the plates, we can use the map provided along with the Trap2017 dataset as a source of information to evaluate the obtained routes clusters obtained. Specifically, since we aim at identifying the options/parameters that provide a more accurate classification, we define two reasonable classes of routes and evaluate whether our classifier is able to recognize them. The two classes are:

- **Adjacent-Gates Routes (AGR)**: routes composed by two gates which are adjacent on the map. For instance, route (27,9) is in AGR, while route (27,26) is not.
- **No-Exit Routes (NER)**: routes which are compliant with the direction of travel, and can be therefore travelled without exiting the highway. For instance, route (27,26) is in NER, while (27,4) is not.

Of course, AGR is a subset of NER.

To measure how the accuracy of our route classifier is affected by different options we rely on the well known *Precision* (P) and *Recall* (R) scores, computing P and R for each cluster for both AGR and NER. Ideally, finding a cluster with both large P and large R means being able to recognize routes of the considered class (either AGR or NER) with only a few false positives and false negatives. In general, the existence of one or more clusters with large P means that our classifier was able to detect the difference between the considered class and all other routes, whereas the existence of a single cluster with large R means that it was able to recognize the similarity among routes of the considered class.

In Table 1 we report the results of a preliminary set of experiments aimed at assessing the effects of four possible alternatives for data preprocessing: (a) doing nothing; (b) considering only travel times not larger than 10 hours to remove episodic/irrelevant events; (c) using the median of travel times of a route as a "normalizing constant" to remove the dependence on the route length; (d) taking the logarithm of all travel times to adjust their density. The information gained by Table 1 can be summarized as follows:

- When using no preprocessing at all, AGR is nicely captured by Cluster 0, although a non negligible 28% of AGR routes are scattered over other clusters. The many clusters with

P=1 for NER mean that NER is split into many sub-classes, the largest of which contains only 43% (R=0.43) of all NER routes.

- Cutting out all travel times larger than 10 hours provides an excellent classification of AGR, with Cluster 2 having optimal P=1 and R=0.9. Unfortunately, NER is not equally well clustered, since 85% of NER routes belong to Cluster 0 but they only sum up to 21% of the members of that cluster.
- Normalizing, a choice apparently consistent with the underlying logic of our metrics, yields Cluster 4 having P=0.87 and R=0.67 for AGR, and Cluster 3 having P=1 and R=0.58 for NER. However, non negligible percentages (10-17%) of each of the two classes are associated with other clusters.
- If we take the log of all travel times, an option which impacts on the GMM algorithm by modifying temporal distances, we see that Cluster 0 has P=0.60 and R=0.93 for AGR, while Cluster 2 has P=0.99 and R=0.62 for NER. Interestingly, Cluster 0 also has P=1 and R=0.27 for NER, meaning that the sub-optimal P of Cluster 0 for AGR and the sub-optimal R of Cluster 2 for NER are both due to part of NER routes being associated with AGR routes and thus being mapped to Cluster 0 instead of Cluster 2.

As we aimed at having two clusters respectively representing AGR and NER, in the following we will focus on option (d), *i.e.*, taking the logarithm of all travel times. Indeed, we believe it is reasonable to associate the shortest NER routes to AGR, and we therefore pick the classification produced with option (d) as the neatest one. However, we point out that this choice is somewhat discretionary, and other preprocessing options could be preferred. For instance, a more detailed analysis of the map could lead to the conclusion that the classification provided by option (a) is a more refined partitioning of NER into meaningful sub-classes. Any deeper investigation is however left to future work.

Table 1: Accuracy of our route classifier

	Cluster 0		Cluster 1		Cluster 2		Cluster 3		Cluster 4	
	P	R	P	R	P	R	P	R	P	R
AGR	**1.00**	**0.72**	**1.00**	0.03	0.00	0.00	0.03	0.07	0.15	0.17
NER	**1.00**	0.12	**1.00**	0.01	0.07	0.24	**1.00**	**0.43**	**1.00**	0.20

(a) Raw travel times

	Cluster 0		Cluster 1		Cluster 2		Cluster 3		Cluster 4	
	P	R	P	R	P	R	P	R	P	R
AGR	0.00	0.10	0.00	0.00	**1.00**	**0.90**	0.00	0.00	0.00	0.00
NER	0.21	**0.85**	0.00	0.00	**1.00**	0.15	0.00	0.00	0.00	0.00

(b) Travel times $\leq$ 10 hours

	Cluster 0		Cluster 1		Cluster 2		Cluster 3		Cluster 4	
	P	R	P	R	P	R	P	R	P	R
AGR	0.00	0.00	0.33	0.03	0.03	0.10	0.19	0.17	**0.87**	**0.67**
NER	0.04	0.12	0.33	0.01	**1.00**	**0.58**	**1.00**	0.15	**1.00**	0.14

(c) Normalized travel times

	Cluster 0		Cluster 1		Cluster 2		Cluster 3		Cluster 4	
	P	R	P	R	P	R	P	R	P	R
AGR	**0.60**	**0.93**	0.00	0.00	0.02	0.07	0.00	0.00	0.00	0.00
NER	**1.00**	0.27	0.00	0.00	**0.99**	**0.62**	0.03	0.11	0.00	0.00

(d) Logarithm of travel times

4.2 Classifying Routes

Let us now focus on the ability of our classifier to model and cluster routes. As mentioned before, all following results refer to logarithmic preprocessing.

Figure 2 shows the travel time statistics, equipped with the inferred GM, for nine different routes. Routes (11,19), (24,17), (2,7) and (7,6) belong to the AGR class, whereas the other five routes are "anomalous" for some reason: routes (4,26) and (3,1) connect two gates located at the same Km of the highway, but on opposite directions; route (27,1) connects two gates which are both far away and on opposite directions; route (26,9) connects two adjacent gates, but on the wrong direction of travel; finally, route (9,9) is a loop.

Looking at the plots, a few observation can be made:

- Reasonable and anomalous routes can be easily told apart, since the former have a peak at a few minutes, while the latter are unbalanced towards a day or a week. This difference is well captured by their respective GMs.
- Among reasonable routes, we can clearly distinguish routes (11,19) and (2,7) from routes (24,17) and (7,6), despite all four routes are in AGR and contain exactly one service area. The main divergence between these two pairs is that (11,19) and (2,7) are around 10 Km

long, compared with (24,17) and (7,6) being more than 40 Km long, a difference which impacts on both the mean time and the variance of behaviors of road-users on these routes and explains why (24,17) and (7,6) are the only two routes of AGR associated to routes Cluster 2. Notably, routes (24,17) and (7,6), which correspond to the very same highway section but in the two opposite direction of travel, have a almost identical GM.

- Among anomalous routes, (3,1) and (4,26) can be easily paired and distinguished from the others, which is desirable since they are two completely analogous routes, as previously described. In general, the behaviors captured by the GMs of these routes are somewhat expectable, except for route (27,1) having a non-negligible Gaussian with mean time less than 2 minutes. This is completely unrealistic as the shortest feasible path between gates 27 and 1 is surely several tenths of Km long. This aspect may be the proof of one or more cloned plates, and it would require further investigation which lies beyond the scope of this paper.

Finally, let us focus on the routes clusters identified by our classifier. Table 2 reports a synthetic description of these clusters, where for each of them we report its size and a aggregate GM obtained averaging the GMs of all the members of the cluster. For these aggregate GMs, we report the mean (μ) and the weight (w) of each Gaussian, ordered by the latter. Clusters 1 and 4 are composed by just a few routes. Significantly, all but one of these routes are loops, *i.e.*, routes delimited on both sides by the same gate, but even considering both clusters together we only get about a third of all possible loops. All in all, these two clusters seem of scarce interest. Conversely, Clusters 0 and 2, as already emerged in Section 4.1, group together all routes that are compliant with the direction of travel, and the only significant difference between routes appearing in the two clusters seem to be their length. This is also visible in the aggregate GMs of these two clusters: Cluster 0 contains shorter routes whose travel times are strongly unbalanced towards values in the order of a few minutes, which is the expected travel time of a vehicle running across that route at fast but licit speed; on the other hand, along longer routes, such as those belonging to Cluster 2, we detect a flatter distribution, in which travel times around an hour or even half a day are notably common. Finally, Cluster 3 contains all other routes. As previously highlighted, for this larger class of routes much larger travel times, ranging from half a day to as much as two months, are extremely recurring.

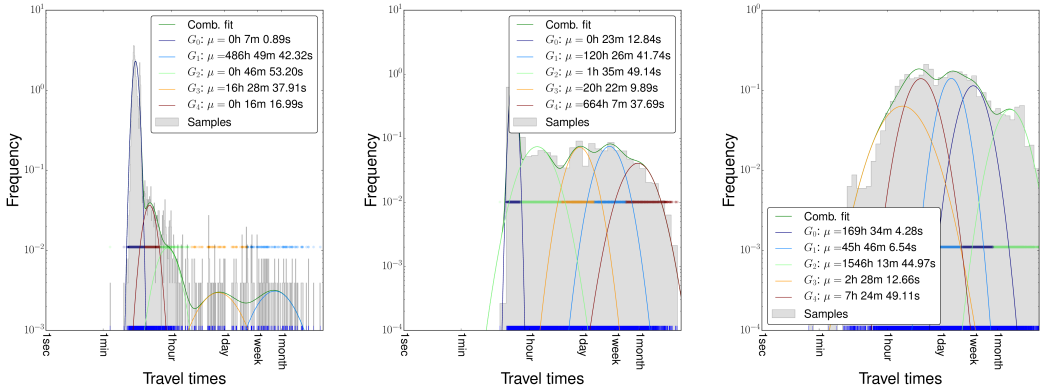
Table 2: Routes Clusters

Cluster	Size	G_0		G_1		G_2		G_3		G_4	
		μ	w	μ	w	μ	w	μ	w	μ	w
0	45	09m02s	0.69	16m11s	0.20	2h13m37s	0.07	34h12m44s	0.03	542h38m17s	0.02
1	9	179h19m16s	0.34	46h3m29s	0.24	893h48m24s	0.17	19h50m32s	0.13	12h18m54s	0.12
2	106	1h16m28s	0.34	14m36s	0.32	12h27m24s	0.13	158h31m21s	0.12	1226h24m23s	0.09
3	568	258h50m18s	0.30	60h37m2s	0.26	1446h59m20s	0.21	10h22m19s	0.16	1h13m27s	0.07
4	1	155h4m30s	0.44	1233h35m15s	0.28	1h22m20s	0.17	23h25m20s	0.09	1s	0.01

4.3 Classifying Plates

Finally, we can switch to our plates classifier and assess what can be inferred about the behavior of different classes of plates based on its outcomes.

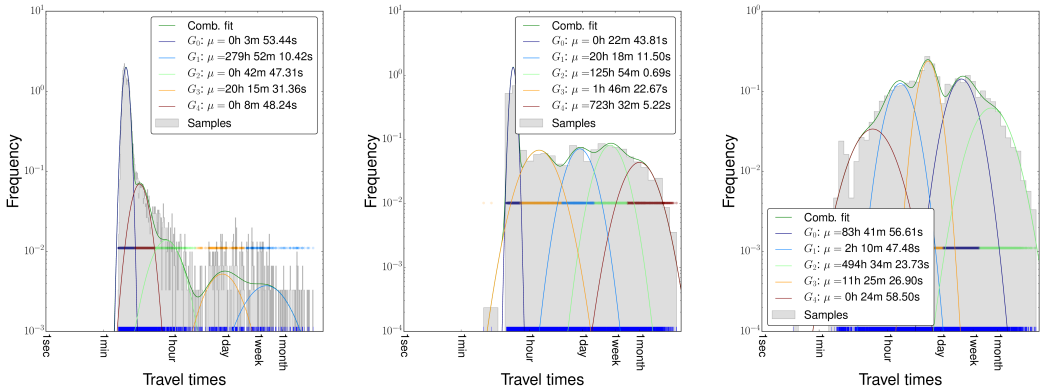
Table 3 presents a summary of the clusters obtained by our plates classifier. We recall that each plate is described by means of the distribution of its travel times over the aggregate GMs we produced for each routes cluster. Indeed, GMM *de facto* provides a generative model for travel times measured over a route, allowing each recorded event to be interpreted as an outcome of a more general behavior. Considering aggregate GMs for an entire routes cluster is a way to define a restricted set of possible behaviors that may occur over all routes of that class. Now, plates can be classified based on how frequently they behave in a specific way over routes of a specific type. In Table 3, other than the size of each cluster, we report highlights of the typical behavior of plates belonging to that cluster, in



(a) Route (11,19)

(b) Route (24,17)

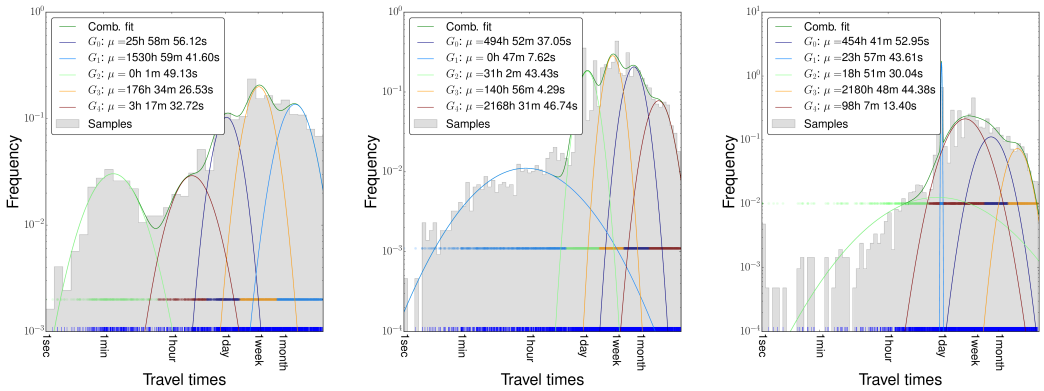
(c) Route (4,26)



(d) Route (2,7)

(e) Route (7,6)

(f) Route (3,1)



(g) Route (27,1)

(h) Route (26,9)

(i) Route (9,9)

Figure 2: Comparison of routes statistics

terms of a short description of the Gaussians they are most frequently associated with. For instance, the first quadruplet of Cluster 0 says that plates belonging to that cluster are associated 13% of the time to Gaussian 2 of Routes Cluster (RC) 3, which means that 13% of the events involving those plates are travels along routes of RC 3 each having average temporal length of 60h37m2s.

A few significant insights can be deduced from a closer look at Table 3:

- Cluster 2 contains more than 3.5M plates which exhibit a very standard behavior: they almost exclusively travel over routes of type 0, which are the most reasonable type, and half of these travels occur at the most frequent/expected speed. We argue that these plates

Table 3: Plates Clusters

Cluster	Size	Most relevant Gaussians											
0	2380633	RC	G	μ	w	RC	G	μ	w	RC	G	μ	w
		3	2	60h37m2s	0.13	0	3	34h12m44s	0.12	3	4	1446h59m20s	0.12
1	635257	RC	G	μ	w	RC	G	μ	w	RC	G	μ	w
		3	0	1h13m28s	0.47	3	1	10h22m19s	0.32	3	3	258h50m18s	0.03
2	3680336	RC	G	μ	w	RC	G	μ	w	RC	G	μ	w
		0	0	9m2s	0.45	0	3	34h12m44s	0.16	0	4	542h38m17s	0.09
3	247398	RC	G	μ	w	RC	G	μ	w	RC	G	μ	w
		3	3	258h50m18s	0.79	3	1	10h22m19s	0.04	3	2	60h37m2s	0.03

are vehicles driven by ordinary road-users, and thus they can be pruned from the dataset if the goal is identifying criminals. If our intuition is correct, this would mean reducing by more than a factor 2 the total number of plates to be classified.

- Cluster 0 is also very large, summing up to more than 2M plates. Contrarily to Cluster 2, however, plates belonging to this cluster exhibit a very diverse and odd behavior, which probably deserves a deeper investigation (*e.g.*, through a second layer of clustering).
- Cluster 3 is composed of plates that, altogether, are associated almost 80% of the time with travel times of approximately 10 days over anomalous routes. A possible explanation is that these are plates for which we only have a few records, thus what we interpret as travels are actually occasions in which they left the highway to only come back a few days after.
- Finally, Cluster 1 is probably the most interesting one, since it is characterized by only two very recurring habits: travelling over routes of RC 3 with travel times either close to 1 hour or to 10 hours, which are both reasonable values despite implying two different activities. We do not have enough information to really understand the meaning of similar behaviors. However, it would not be surprising to discover they can be associated with known suspicious/criminal patterns, such as moving goods from one point of the highway to another.

5 Related Work

5.1 Traffic Monitoring and Analysis

A large amount of work on traffic monitoring and analysis has been carried out in the past. In fact, making sense and extracting meaningful information from the large data flows collected by cameras and GPS tracking is not a trivial task.

Sivaraman's et al. work [1] is an interesting starting point over on-road vision-based vehicle detection, tracking, and behavior understanding. They provide a survey of recent works in the literature, placing vision-based vehicle detection in the context of sensor-based on-road surround analysis. Authors detail advances in vehicle detection, discussing monocular, stereo vision, and active sensor-vision fusion for on-road vehicle detection. Authors also characterize on-road behavior, introducing common performance metrics and benchmarks. Our present work is much different, as we are not in control of the way data is captured. However, we have tried to infer *a posteriori* some information from the classification of the given raw data. Interesting work on traffic pattern analysis and optimization can be found in a work by Koller et al. [2]. They leverage machine vision-based technology and high-level symbolic reasoning to develop a system for detailed, reliable traffic scene analysis. Their symbolic reasoning approach uses a dynamic belief network to make inferences about traffic events such as vehicle lane changes and stalls. Koller's work is complementary to ours, and interesting future work can be foreseen by integrating mutual results.

5.2 Pattern Mining and Clusterization

Jindal et al. [7] analyze the problem of mining frequent patterns from road traffic data by developing a method to mine spatiotemporal periodic patterns in the traffic data and use these periodic behaviors to summarize the huge road network. Their first step is to find periodic patterns from the speed data of individual road sensor stations, then use their periods to represent the station's periodic behavior using probability distribution matrices. Jindal uses density-based clustering to cluster the sensors on the road network based on the similarities between their periodic behavior as well as their

geographical distance, thus combining similar nodes to form a road network with larger but fewer nodes. This work is somewhat similar to our present work. However, our approach is somehow more universal as it can be applied to heterogeneous data with little or no prerequisites. Elfeky et al. [8] use periodicity mining to predicting trends in time series data. They address the problem of detecting the periodicity rate of a time series database. They define different types of periodicities and propose scalable algorithms performing in O($n \log n$) time for a time series of length n. Also Kiran et al. [9] aim at discovering partial periodic itemsets in temporal databases. They introduced a new measure (periodic-frequency) to determine the periodic interestingness of itemsets by taking into account their number of cyclic repetitions in the entire data. These two contributions are interesting and can be leveraged for a given analysis following our approach described here.

Giannotti et al. [10] leverage a knowledge discovery process to mine frequent travel patterns, big attractors and extraordinary events influence on mobility. They aimed at predicting dense traffic areas in the near future. They also defined M-Atlas, a querying and mining language that eases the analytical process by transforming raw GPS tracks into mobility knowledge. Giannotti's work will be considered for future work in combination to our present results. Necula [11] performed R-based statistical analysis to identify contiguous set of road segments and time intervals which have the largest statistically significant relevance in forming traffic patterns. He mined vehicle traces to extract outlier traffic patterns. Similarly to what we did, he organized the road infrastructure as segments in a graph and tracks the visits for each vehicle. He found that over time, the visited segments settle into a pattern and vary periodically. Grossi et al. [12] address the problem of clustering data as a machine learning problem, as well as an optimization problem. They present a constraint programming model for a centroid based clustering and one for a density based clustering. In particular, as a key contribution, they show how the formulation of the density-based clustering by constraint programming makes it very similar to the label propagation problem and they propose a variant of the standard label propagation approach. Their approach is quite different from the one presented here. Future work will investigate benefits and pitfalls of our approch in comparison to Grossi's.

6 Conclusions and Future Work

In this paper we have described the efforts, pitfalls, and successes of applying a purely automated classification/clustering approaches to analyze the TRAP-2017 challenge dataset. All work was performed leveraging open source tools, self-written (python, perl, bash) code and state of the art scientific software libraries. Various approaches to data filtering/cleaning have been manually applied and compared, and all obtained results and figures have been analyzed and discussed. Our findings show that unsupervised clustering is a viable approach to extract meaningful information about the composition of the dataset. Further, by building our classifier upon a formal and descriptive definition of behavior of a plate, we created the conditions for police officers to fully characterize the clusters produced by our tool. Additional analysis and understanding of the results will be part of future work. We believe the results described here can pave the way to interesting research on the matter.

References

[1] S. Sivaraman and M. M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, Dec 2013.

[2] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proc. 33rd IEEE Conf. on Decision and Control*, volume 4, pages 3776–3781 vol.4, Dec 1994.

[3] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[4] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *Intl. journal of computer vision*, 40(2):99–121, 2000.

[5] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[6] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[7] Tanvi Jindal, Prasanna Giridhar, Lu An Tang, Jun Li, and Jiawei Han. *Spatiotemporal periodical pattern mining in traffic data*. 2013.

[8] Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. Periodicity detection in time series databases. *IEEE Trans. on Knowl. and Data Eng.*, 17(7):875–887, July 2005.

[9] R. Uday Kiran, Haichuan Shang, Masashi Toyoda, and Masaru Kitsuregawa. Discovering partial periodic itemsets in temporal databases. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM '17, pages 30:1–30:6, New York, NY, USA, 2017. ACM.

[10] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pinelli, Chiara Renso, Salvatore Rinzivillo, and al. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20(5):695–719, oct 2011.

[11] Emilian Necula. Analyzing Traffic Patterns on Street Segments Based on GPS Data Using R. *Transportation Research Procedia*, 10:276 – 285, 2015. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands.

[12] Valerio Grossi, Anna Monreale, Mirco Nanni, Dino Pedreschi, and Franco Turini. Clustering formulation using constraint optimization. In *Selected Papers of SEFM 2015 Worksh on Software Engineering and Formal Methods - Volume 9509*, pages 93–107, New York, NY, USA, 2015. Springer-Verlag New York, Inc.