Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 231–236

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Artificial Intelligence Methods for Smart Cities (AISC 2021)
November 1-4, 2021, Leuven, Belgium

Private or Public Parking Type Classifier on the Driver's Smartphone

Emanuele Panizzi[a,*], Alba Bisante[a]

[a] *Department of Computer Science, Sapienza University of Rome, Rome, Italy*

Abstract

As the number of vehicles and car owners rapidly increases, how to resolve problems related to on-street parking is gradually becoming one of the most challenging tasks of smart cities. Smart parking solutions exploit existing technology to provide drivers with information about available parking spots, often leveraging physical, expensive road infrastructures. Our goal is to analyze previous user's parking spots and discern private places from on-street ones. We identified the main characteristics of the private parking spots and trained a machine learning model that clusters them and classifies the driver's last parking. The applicability of our work is in smart parking systems, which can benefit from this classification in three ways: i) tagging parking spots in the city without the need of sensor infrastructure; ii) ease parking exchange when a public parking spot gets available; iii) predict when a user is approaching a public or a private spot based on his last parking in the same area.

Keywords: smartphone; parking; sensing; machine learning; curb; smart city; on-street parking; private parking;

1. Introduction

Smart parking systems allow exchanging parking spots between users of the same community, generally by leveraging sensors or other infrastructures that detect a new vacancy in a parking spot [9]. Such systems deploy a large number of sensors in public parking spots at a high cost. An alternative to deploying external infrastructures is to use context-aware information [10], such as data sensed by drivers' smartphones [11], which greatly reduces costs but requires determining if the driver's place is a good parking spot for another user. Private parking spots, for example, are not suitable to be used by drivers other than the owner.

This work aims at classifying automatically private and public parking on the driver smartphone using the data collected during the last minutes of travel, as this can help to build low-cost smart parking systems.

We based our research also on the observation that it is widespread among drivers to have a parking routine. Each driver often visits the same places and regularly searches for a parking spot in the same way, following repetitive

* Corresponding author.
E-mail address: panizzi@di.uniroma1.it

1877-0509
10.1016/j.procs.2021.12.233

patterns. Searching habits for public parking are different from private parking: the driver does not need to perform a long parking search at arrival in the latter. Therefore, a byproduct of our research could be ease automated parking searches by distinguishing if the driver is approaching a private parking. To this matter, it is essential to specify that the private/public distinction does not represent the driver's actual ownership of that parking area. It could be a private parking lot or an area where locating a parking spot is very fast, such as a dedicated parking facility or a countryside area.

We organized our work as follows. By leveraging a prototype parking mobile app, it was possible to collect easily, for nine monitored users, the GPS coordinates of places where they parked their cars. Secondly, we processed those locations with a clustering algorithm to identify where the users parked more frequently. Finally, by observing the collected data we inferred information that we used to train a machine learning model to classify the parking clusters as public or private.

2. Related Work

A common problem faced in research regarding smart parking systems is to classify a given parking space as free or occupied (among the most recent publications, [1], [2]). On the other hand, there are very few works that address parking areas classification with different labels. We did not find any reference to studies regarding the specific classification of private/public parking spots; however, this research is still relatable to other parking classification systems.

As an instance, the work presented in [3] aimed at labelling parking spots as parallel, angle or perpendicular through machine-learning. The proposed classifier runs on the driver smartphone and leverages its inertial sensors. However, it does not rely on GPS coordinates and does not address a clustering problem. Other classification studies, which exploit GPS parking locations as this work, have been proposed by [4], [5] and [6].

In [4], the authors analyzed the parking locations of BMW cars in Munich city with the purpose of categorizing parking behaviour. They distributed the parking events over quadkeys and analyzed parked-in time and parking duration through unsupervised learning technology. As a result, they categorized areas by quadkeys into a combination of residential, business, eating, and shopping areas. Therefore, the classification was based on the purpose of the car trips and aimed at classifying city areas, not at identifying single users' habits.

In [5], Zheng et al. proposed a method to recognise potentially interesting trends by applying automated clustering and anomaly detection techniques to real-time parking data. For example, by analyzing data obtained by parking sensors deployed in San Francisco, they identified clusters of heavy usage parking spots. Also in this case the classification is done over city areas and is not user-centred. Moreover, this study lacks a specific classification objective, and the areas are labelled based on post-analysis observations.

Finally, the authors of [6] developed an exhaustive classification scheme for HCV parking locations. In this case, however, the classification method is offered as a framework and the authors do not propose an automated implementation of it. As an example, they present a case study where they apply the scheme to a collection of GPS locations, but they do not specify if the application is done manually or automatically.

A significant difference between the cited studies and the work presented in this paper is that the first ones rely on external sensors (in-car technology in [4], on-street parking sensors in [5] and not specified in [6]) instead of taking advantage of the highly portable and low-cost smartphones technology. Also, none of them proposes a classifier that runs on smartphones.

With respect to the adopted clustering techniques, the cited authors developed variations of known algorithms. In [4], it is leveraged a combination of DBSCAN and K-means algorithms, which are some of the most widespread algorithms for clustering GPS positions. Other more specialized algorithms are instead adopted in [5], but in this case, the authors have selected clustering techniques specifically suitable for anomaly detection, which is beyond the scope of our research.

At present, we believe that modelling the parking data as a graph and applying a community detection algorithm better fits our context. In this way, we avoid the problem of deciding a priori the k parameter, and we still obtain satisfying results. However, for future developments, we plan to refine the clustering step by comparing the obtained results with that of other techniques.

To the best of our knowledge, our work is the first that offers an automated classifier of public and private parking spots, which runs on the driver smartphone and only relies on its sensors. In the long term, we aim to improve user experience on smart parking applications with personalized location-based services, without requiring additional sensors.

3. Collection of Data

We collected data using a smart parking prototype application that we developed, which allowed us to differentiate each monitored car trip by user. This application detects parking every time the user's smartphone disconnects from the car BlueTooth and stores the parking GPS location. The user needs to keep the app in background mode, where it silently gathers information. The testing group consisted of nine drivers who use their car daily to reach destinations of interest, such as work or study places.

Over two months, we collected the GPS locations of all the parking spots occupied by the testers. We conducted the test in Rome, Italy. We collected **1472** parking locations (table 1).

4. Experiment

4.1. Parking Clusters

For each user, we built a graph, where each parking made was treated as a graph node; an edge was placed between pairs of nodes only where the two associated parking locations were less than 50m apart. The edges were weighted inversely to the distance between the associated parking spaces. In addition, we assigned greater importance to the connections of parking carried out in the same time slot.

As we modelled the problem with a graph, the clustering technique adopted mainly leveraged the Louvain heuristics [7] for finding communities among graphs with weighted edges. In addition to that, for each cluster, we computed a mid-point coordinate, then we merged the pairs of clusters where such mid-points were less than 200m apart.

Table 1. Collected Data

User	Parkings	Clusters	Public	Private
0	318	23	22	1
1	626	33	31	1
2	274	24	19	5
3	62	5	5	0
4	51	6	5	1
5	20	6	6	0
6	40	5	2	3
7	5	1	1	0
8	76	7	7	0
Total	1472	109	98	11

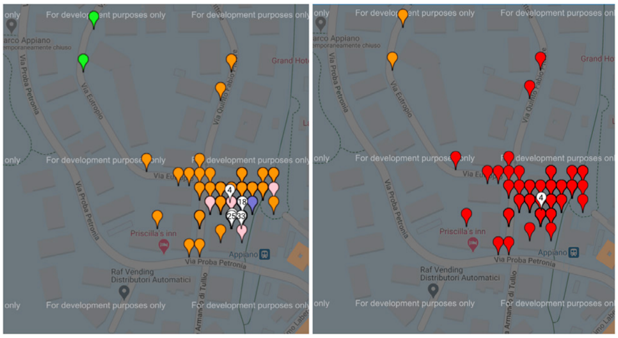

Fig. 1. Before and after the extra merging step.

This extra clustering step and the metric choices were due to manual checks and observations made on the collected data. Figure 1 shows an example of a user's Parking Cluster before and after the merging process. Each pin marks one parking spot (some pins overlap); different colours identify different clusters; the white pins mark the mid-point of the cluster named after the pin label. In this example, before merging, there were four different clusters in the same zone (the clusters named 4, 18, 25, and 33). After the merging procedure, they were unified in cluster number 4, coloured in red. Please note that the algorithm did not merge the two green parking spots at the left-top corner of the first image to cluster number 4 because they were too distant.

The clustering process identified **109** different Parking Clusters with at least two parking spots.

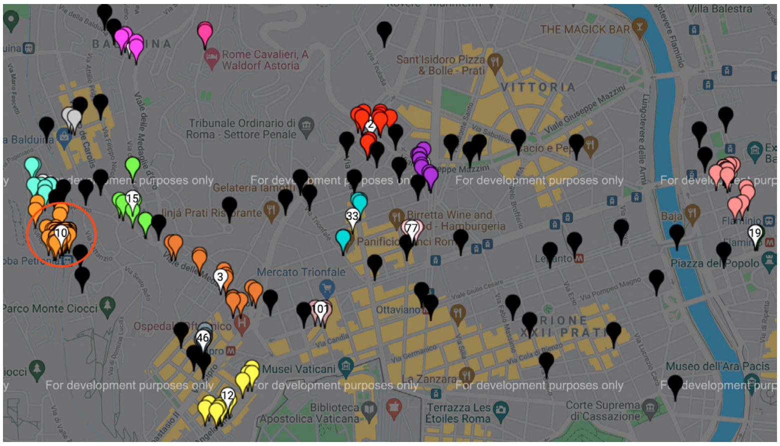

Fig. 2. Parking Clusters

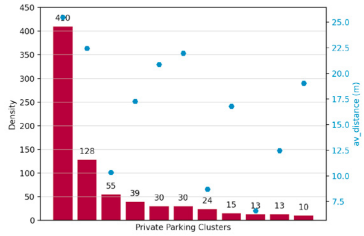

Fig. 3. Private Parking Clusters

4.2. Clustering Example

Picture 2 presents part of the parking clusters resulted from the application of the clustering algorithm to the data collected by User 0. As explained above, each pin on the map represents a parking spot, and pins that share the same colour are part of the same cluster. Black pins indicate singleton clusters. The only private cluster of the user is the one on the left, circled in red.

This example shows both the quality of the adopted algorithm and some of its criticalities. Indeed, all the identified clusters represent actual parking areas of interest for the user. However, even after the extra merging step (showed in 1), still a lot of outliers parking spots persist that could be part of adjacent clusters, as in the case of the black pins around cluster 46 on the bottom-left corner of the picture. In other cases, the algorithm correctly identified isolate parking spots as singletons. Unfortunately, there is no standard way to evaluate the accuracy of a clustering algorithm that aims to find parking areas, if not manually checking each cluster and eventually questioning the user. At present, we believe that the adopted algorithm satisfies the requirements of this research, and we leave for future work the possibility of further refining its performance.

4.3. Discern Private Parking from On-Street Parking By Observing the Data

Once identified the Parking Clusters for each user, it became necessary to find some descriptive characteristics of the clusters useful for the machine learning model.

We define the *avDistance* of a cluster as the average distance between each parking space in a cluster and the cluster midpoint, and the *density* of a cluster as the number of nodes it contains. Graphs 3 and 4 can be compared in order to appreciate the descriptiveness of these two combined values.

Firstly, we noticed that the collected private clusters tend to have an *avDistance* value smaller than 30m. This finding can be explained by reasonably assuming that if the parking area is private, the user will tend to park in the same spot, with minimal variation in the distance between one parking and another, mainly due to GPS location error. However, it is also necessary to consider the number of parking spaces carried out in the same area. In clusters with few parking spaces, the distance between them can be accidentally very small and therefore mislead the resulting identification.

Therefore, we took into consideration also the *density* parameter. By looking at the first two clusters in graph 3, in private clusters high *density* values are related to small values of *avDistance* (at most 25m). On the other hand, looking at public clusters, graph 4, when the number of parking spots in a cluster raises, the associated *avDistance* raises as well, with peaks between 100 and 120 meters.

By this first observation, we assume that private parking clusters have *avDistance* lower than 30m. Looking at figure 4, however, we also notice that some public Parking Clusters that count less than 20 nodes have *avDistance*

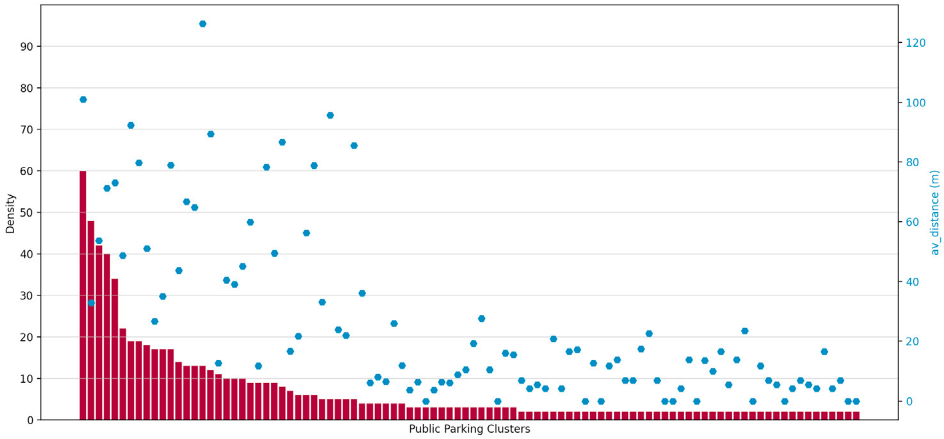

Fig. 4. Public Parking Clusters

values lower than 30m. As already observed, the lower the number of nodes, the less significant is the resulting average distance. Hence, we consider the previous assumption as valid if the cluster has a minimum number of 10 nodes.

Starting from these two observations, we developed a machine learning classifier that exploits *avDistance* and *density* to distinguish private parking clusters from public ones.

5. Parking Cluster Type ML Classifier

Table 2. Classifier Parameters

Max Depth	6
Max Iterations	10.0
Min Loss Reduction	0.0
Min Child Weight	0.0
Random Seed	42
Row Subsample	0.3
Column Subsample	1.0

We trained a machine learning classifier with the aim of discerning private clusters from public ones. The features adopted to describe the clusters were *avDistance* and *density*. We divided the total number of clusters obtained with the clustering process between the training and testing phases. Unfortunately, we had very few private clusters: only 11 out of 109 clusters were private parking areas. Therefore, we used 4 private clusters for training and the remaining 7 for the testing phase.

We will present the results obtained from a classifier implemented by adopting the XCode CreateML [8] library. This library allows both a simple and easily reproducible implementation, and the construction of a model that can run directly on the driver smartphone. CreateML accepts data tables in .csv format and automatically identifies the optimal classifier type for the given problem. In our case, the optimal solution resulted to be a BoostedTreeClassifier, and the adopted parameters are listed in table 2.

6. Results

Table 3. Classifier Evaluation

Class	Precision (%)	Recall (%)
private	77.78	100.0
public	100.0	97.2

Table 4. Confusion Matrix

True/Pred	private	public
private	7	0
public	2	70

The obtained results appear promising. We reached **97.47%** accuracy over 79 clusters tested. Our model recognized all the 7 private clusters correctly and classified as private only 2 out of the 72 public clusters. Table 3 displays the precision and recall evaluation parameters, whilst table 4 presents the confusion matrix during the testing phase.

7. Conclusions

The ML classifier presented in this research paper can recognize, with nearly flawless accuracy, if a cluster of parking spots is private or public. We conducted the experiment by collecting the coordinates of 1412 parking spots occupied by nine users (3); for each user, the areas where he/she parked more frequently were identified by modelling the coordinates as the nodes of a graph and by adopting the Louvain clustering algorithm (4.1). We discussed the quality of such an algorithm in (4.2). Starting from some observations made over the collected data (4.3), we trained the ML classifier taking into account the *avDistance* and the *density* parameters, and tested it (5) over 72 clusters.

The applicability of the presented work is primarily in smart parking system applications, and the Parking Cluster Type Classifier is suitable to run directly on a smartphone. Our contribution aims at improving user experience on smart parking applications with personalized location-based services, without requiring additional sensors.

As an example of employment, we developed an iOS application that registers the coordinates of the parking spots occupied by the users and clusters them to identify frequent parking areas. Once a new parking spot is registered, it is added to the closest cluster and is tagged as public or private by running the Parking Cluster Classifier.

8. Future Work

The final version of the Parking Cluster Type Classifier presented to date still offers room for improvement. Mainly, future research will focus on collecting new and more varied data to refine the training phases and consequently increase the significance of the results. In addition to that, we expect the clustering algorithm to be subject to further tests, including a comparison with other well-known clustering techniques.

Finally, in ongoing research, we are using the present results in a cruising-for-parking recognition system.

References

[1] Farley, Albertus Ham, Hanry Hendra,. (2021). Real Time IP Camera Parking Occupancy Detection using Deep Learning. Procedia Computer Science. 179. 606-614. 10.1016/j.procs.2021.01.046

[2] Chen, Wei Sheu, Peng, Wu, Tseng. (2020). Video-Based Parking Occupancy Detection for Smart Control System. Applied Sciences. 10. 1079. 10.3390/app10031079

[3] Bassetti E., Luciani A., Panizzi E. (2021) ML Classification of Car Parking with Implicit Interaction on the Driver's Smartphone. In: Ardito C. et al. (eds) Human-Computer Interaction – INTERACT 2021. INTERACT 2021. Lecture Notes in Computer Science, vol 12934. Springer, Cham. https://doi.org/10.1007/978-3-030-85613-721

[4] Gomari, Syrus and Knoth, Christoph and Antoniou, Constantinos. (2021). Cluster analysis of parking behaviour: A case study in Munich. *Transportation Research Procedia.* 52. 485-492. 10.1016/j.trpro.2021.01.057.

[5] Zheng, Yanxu and Rajasegarar, Sutharshan and Leckie, Christopher and Palaniswami, Marimuthu. (2014). Smart car parking: Temporal clustering and anomaly detection in urban car parking. 1-6. 10.1109/ISSNIP.2014.6827618.

[6] Nevland, E. A., Gingerich, K., Park, P. Y. (2020). A data-driven systematic approach for identifying and classifying long-haul truck parking locations. Transport Policy, 96, 48-59.

[7] Vincent D Blondel and Jean-Loup Guillaume and Renaud Lambiotte and Etienne Lefebvre. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 10. P10008. IOP Publishing

[8] Apple. Retrieved on December 12, 2020. CreateML Framework. Url: *https://developer.apple.com/documentation/createml*

[9] Paidi, V., Fleyeh, H., Håkansson, J. and Nyberg, R.G. (2018), Smart parking sensors, technologies and applications for open parking lots: a review. IET Intell. Transp. Syst., 12: 735-741. https://doi.org/10.1049/iet-its.2017.0406

[10] Mahfooz Ul Haque, Hafiz Zulfiqar, Haidar Khan, Sajid Haque, Muneeb. (2019). Context-Aware Parking Systems in Urban Areas: A Survey and Early Experiments: 7th EAI International Conference, ICCASA 2018, and 4th EAI International Conference, ICTCC 2018, Viet Tri City, Vietnam, November 22–23, 2018, Proceedings. 10.1007/978-3-030-06152-43.

[11] R. Salpietro, L. Bedogni, M. Di Felice and L. Bononi, Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 18-23, doi: 10.1109/WF-IoT.2015.7389020.