Scooter Trajectories Clustering

Machine Learning and Deep Learning

Mirco De Marchi VR445319

University of Verona

2020/2021

Introduction

Motivation

Trajectory clustering is a problem really difficult to be treated but can be useful for several applications:

- Monitoring
- Forecasting
- Viability
- Smart City
- Security

State of art

The current researches can be divided into 5 categories:

- Spatial based clustering: *DBSCAN* algorithm.
- Time depended clustering: *OPTICS* algorithm.
- Partition and group based clustering: *Lee partition & group*.
- Uncertain trajectory clustering: *Fuzzy C-Means* algorithm.
- Semantic trajectory clustering: *Stops and Moves* model.

Starting point

Dataset size: **2GB**.

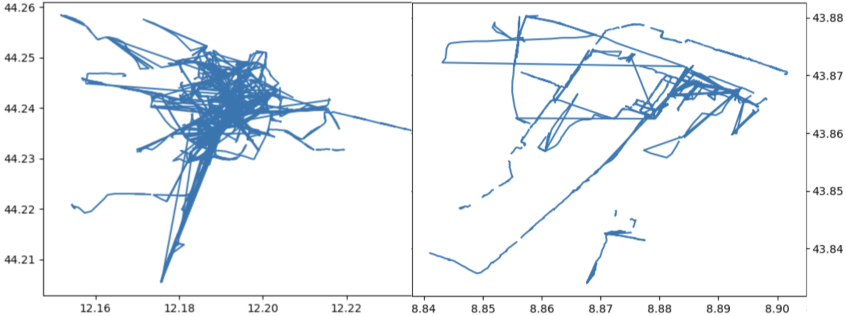
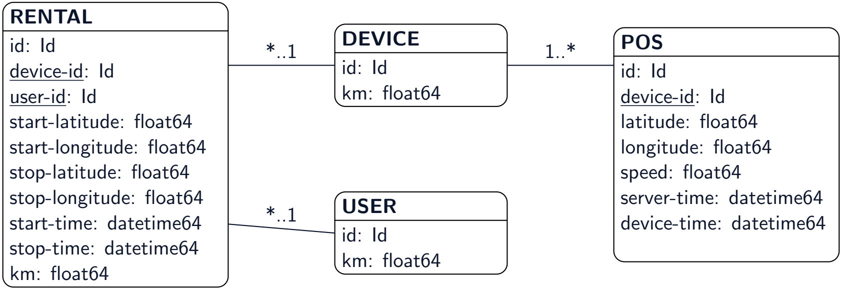


Figure: Rentals showed: 200.

Original dataset diagram

Dataset entities: position, rental, device and user. The dataset has been previously processed in order to delete sensitive informations.



RENTAL

id: Id
device-id: Id
user-id: Id
start-latitude: float64
start-longitude: float64
stop-latitude: float64
stop-longitude: float64
start-time: datetime64
stop-time: datetime64
km: float64

DEVICE

id: Id
km: float64

USER

id: Id
km: float64

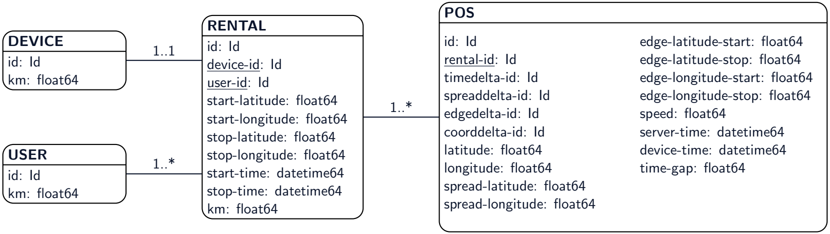
POS

id: Id
device-id: Id
latitude: float64
longitude: float64
speed: float64
server-time: datetime64
device-time: datetime64

Methodology

Generated dataset diagram

Dataset	Samples	Features
rental	14826	10
pos	817076	18
merge	817076	18
dataset	14826	13
partition city 1	608251	18
partition city 2	202795	18



DEVICE
id: Id
km: float64

USER
id: Id
km: float64

RENTAL
id: Id
device-id: Id
user-id: Id
start-latitude: float64
start-longitude: float64
stop-latitude: float64
stop-longitude: float64
start-time: datetime64
stop-time: datetime64
km: float64

POS
id: Id
rental-id: Id
timedelta-id: Id
spreaddelta-id: Id
edgedelta-id: Id
coorddelta-id: Id
latitude: float64
longitude: float64
spread-latitude: float64
spread-longitude: float64
edge-latitude-start: float64
edge-latitude-stop: float64
edge-longitude-start: float64
edge-longitude-stop: float64
speed: float64
server-time: datetime64
device-time: datetime64
time-gap: float64

Rentals trajectories

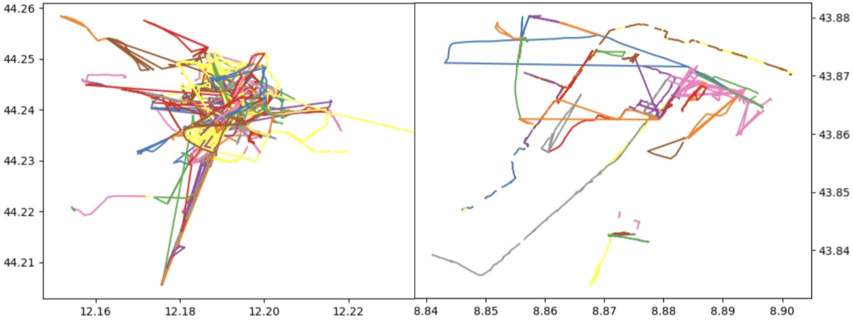


Figure: Rentals showed in the 2 cities: 200 (left), 50 (right).

HEURISTICS: *timedelta* AND *spreaddelta*

The following heuristics methodologies use a `delta` value that is valued with the statistic's empirical rule.

- **timedelta heuristic**: a rental trajectory can be divided in a sequence of trajectories if the time gap between a position and previous one exceeds a *timedelta* value.

$$TIMEGAPS = \{p.time - p[-1].time \mid \forall p \in POS\} \tag{1}$$

- **spreaddelta heuristic**: a rental trajectory is similar to another one if they spread a similar amount of area in relation with *spreaddelta* value.

$$SPREADS = \{max(t) - min(t) \mid \forall t \in TRAJ\} \tag{2}$$

HEURISTICS: *edgedelta* AND *coorddelta*

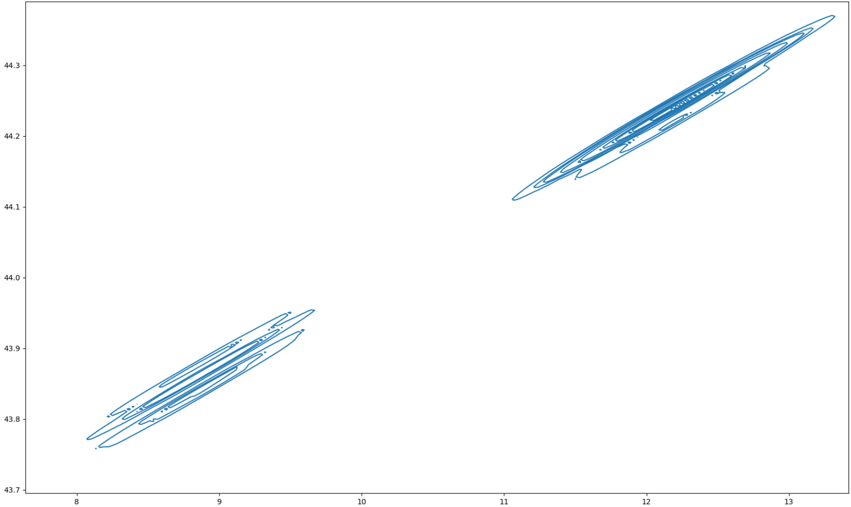
- **edgedelta heuristic**: acts as the *spreaddelta heuristic*, but it considers the edges of a trajectory, or rather the first position and the last position of a trajectory in relation with *edgedelta* value. The main issue is the bimodal distribution of edges.

$$EDGES = \{concat(p[0], p[-1]) \mid \forall t \in TRAJ\} \tag{3}$$

- **coorddelta heuristic**: combination of spread and edge heuristics in order to combine the main advantages.

Position distributions

The positions are concentrated in 2 distant cities:



Feature extraction

Pipeline: integration of heuristic data as features, *Standardization*, *Normalization* and than *Principal Component Analysis (PCA)*.

The component extracted by *PCA* can be decided in 3 different ways:

- By a number a priori;
- By the cumulative variance with 80% cover;
- Concatenation of columns produced by *PCA* for different subset of features;

$$\{\{\mathit{latitude}\}, \{\mathit{longitude}\}, \{\mathit{spread\,latitude}, \mathit{spread\,longitude}\},$$
$$\{\mathit{edge\,latitudestart}, \mathit{edge\,latitudestop},$$
$$\mathit{edge\,longitudestart}, \mathit{edge\,longitudestop}\}\}$$

Clustering

- **K-Means**: simple technique with distance based metric, fast and cheap in memory terms. $O(n * k * I)$
- **Mean Shift**: density based, automatically sets the number of clusters, but it needs a bandwidth parameter. $O(n^2)$
- **Gaussian Mixture**: estimation of linear combination of a finite number of Gaussian distributions with unknown parameters and *expectation-maximization (EM)* algorithm. $O(I * n^3)$
- **Full Hierarchy Agglomerative**: hierarchical clustering with bottom up approach and minimization metric on the maximum distance between observations in pairs of clusters. $O(n^3)$
- **Ward Hierarchy Agglomerative**: hierarchical clustering with bottom up approach and minimization metric on the sum of squared differences between all clusters. $O(n^3)$

Results

Timedelta heuristic results

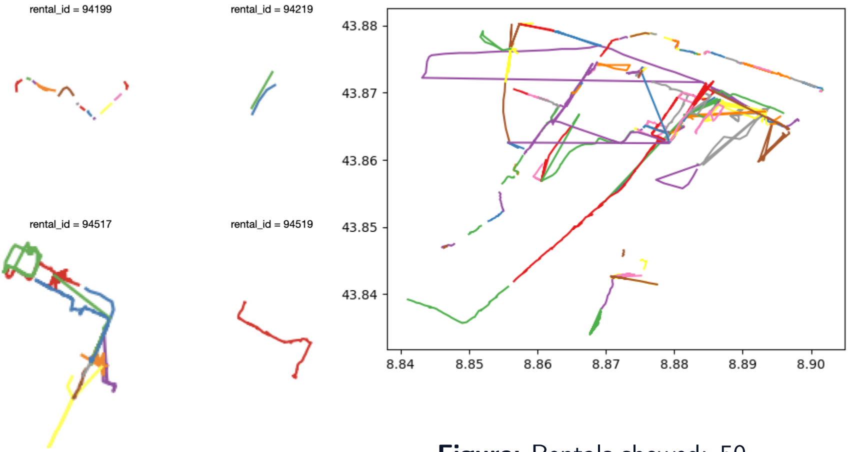


Figure: Rentals showed: 50.

Spreaddelta heuristic results

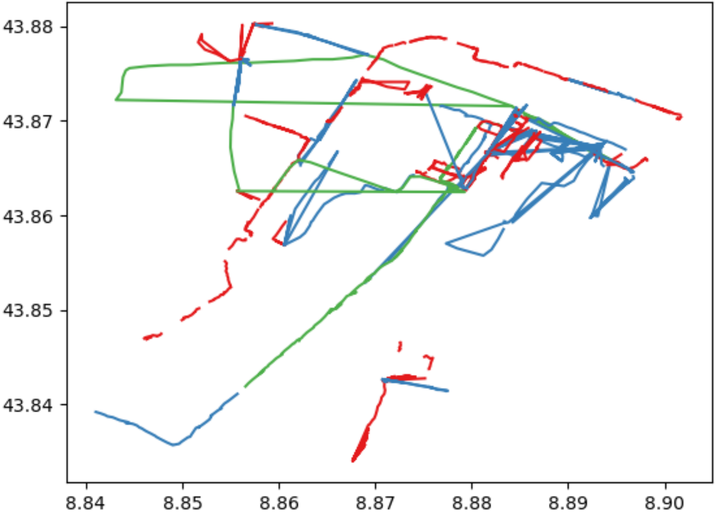


Figure: Rentals showed: 50.

Edgedelta heuristic results

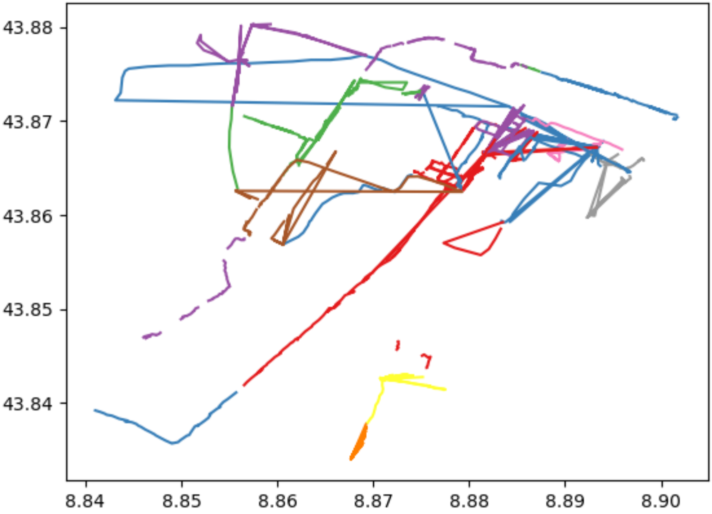


Figure: Rentals showed: 50.

Coorddelta heuristic results

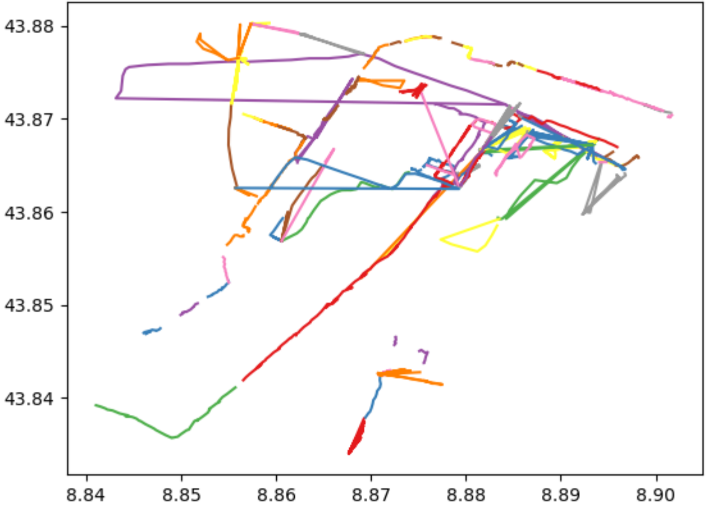
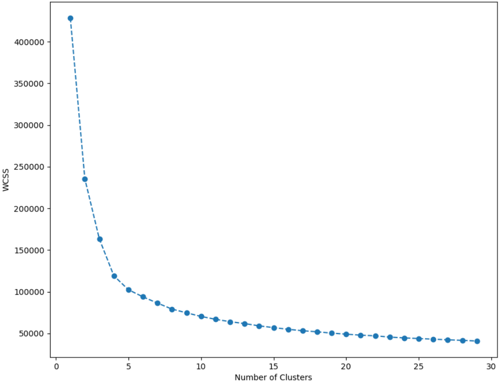


Figure: Rentals showed: 50.

WCSS and Elbow method

Within Cluster Sum of Squares (WCSS) graph for *Elbow method* with number of clusters in range from 1 to 30 and *K-Means* algorithm.



Number of clusters estimated: **5**.

Gaussian Mixture clustering

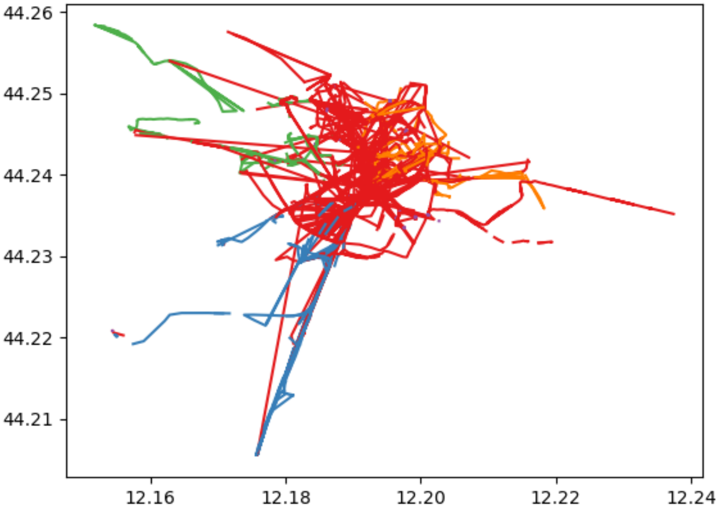


Figure: Silhouette: -0.02. Rentals showed: 200.

Mean Shift clustering

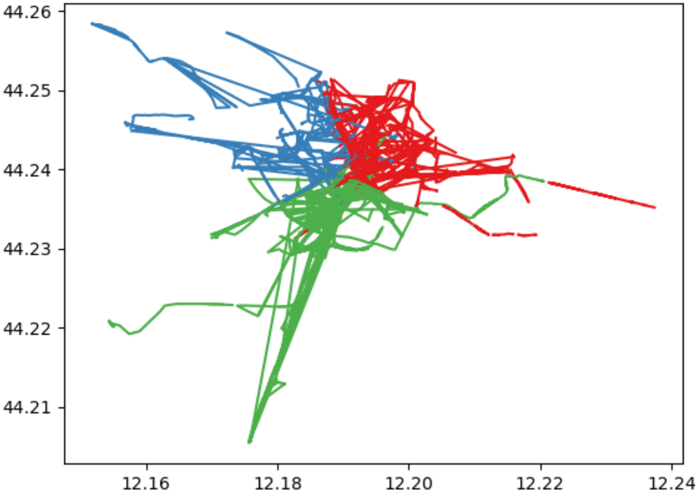


Figure: Silhouette: 0.40. Rentals showed: 200.

Full Hierarchical Agglomerative

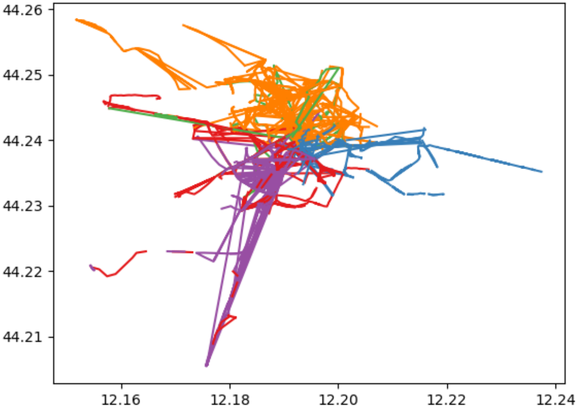


Figure: Silhouette: 0.16. Rentals showed: 200.

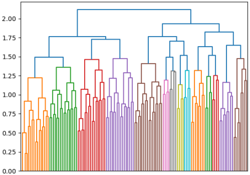


Figure: Dendrogram up to level 5 of merge

Ward Hierarchical Agglomerative

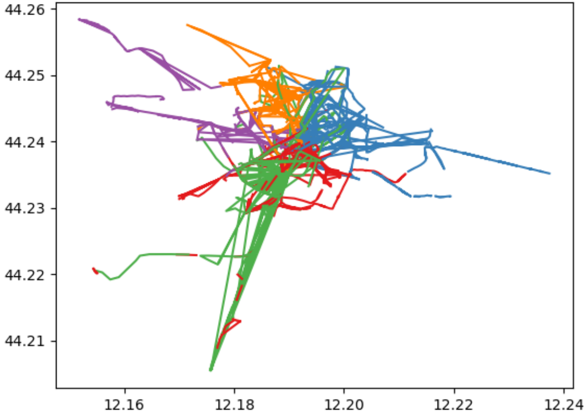


Figure: Silhouette: 0.28. Rentals showed: 200.

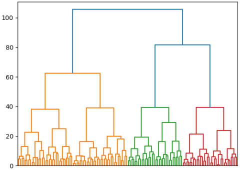


Figure: Dendrogram up to level 5 of merge

K-Means clustering

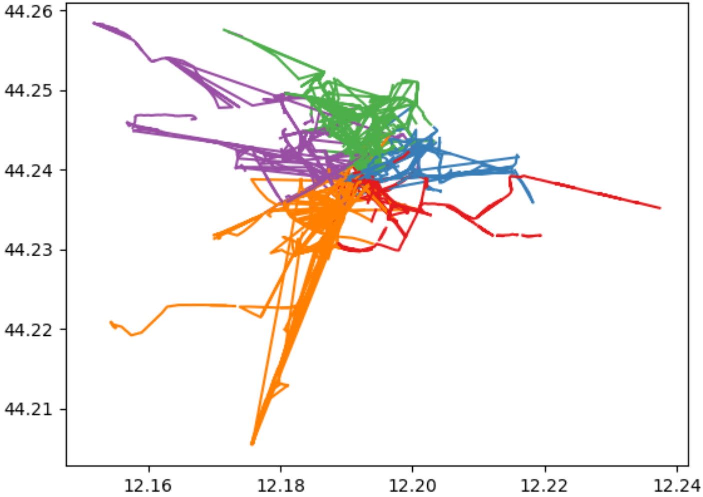


Figure: Silhouette: 0.352. Rentals showed: 200.

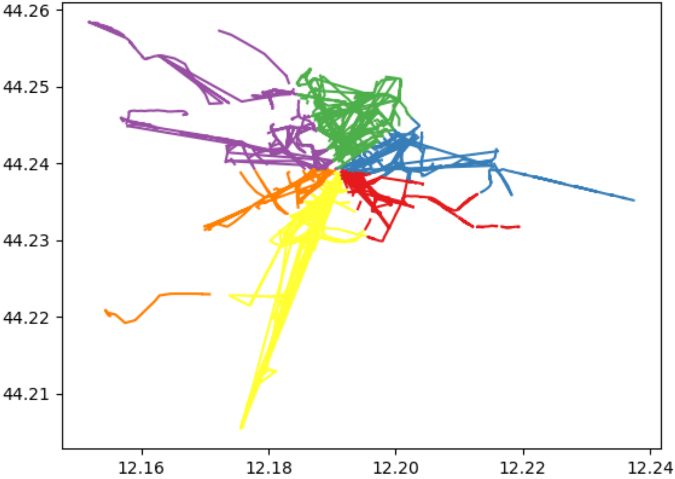
Conclusion

Final considerations

- *K-Means* is the best result in terms of plot representation and *Silhouette score*;
- Custom *PCA* implementation results similar to traditional *PCA* approach based on the 80% of cumulative variance;
- Time useful for partition and group, but not for bottom-up clustering techniques;
- Clustering with *PCA* shows better results in variance terms;
- Clustering with heuristic features maintains the rental information;
- Clustering has always to be performed on a specific region of interest in order to optimize the results;
- *Silhouette score* is not a validation methodology so reliable, because it depends a lot on the data you are dealing with;

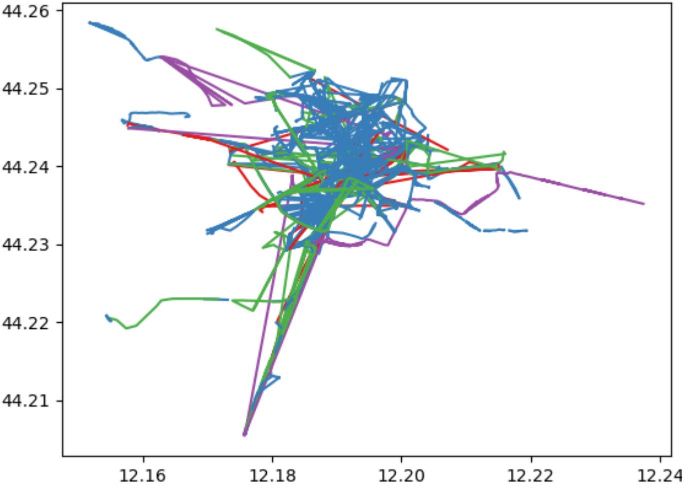
K-Means slices

K-Means without *PCA* with only latitude and longitude features.



K-Means bad result

K-Means with 5 clusters performed on all positions showed on one city.



K-Means 3D