Analysis of the Floating Car Data of Turin Public Transportation system

Roberta Ravanelli, Mattia Crespi

University of Rome “La Sapienza”
Department of Civil, Constructional and Environmental Engineering
Geodesy and Geomatics Division

PRIN meeting
Naples, July 23rd 2019
Introduction

- The largest part of *movements* in an *urban environment* is constrained to the *road network*

- In the field of transportation, *GNSS data* collected from *vehicles* are frequently referred as *Floating Car Data (FCD)*

- *FCD* are *Urban Geo Big Data* and contain the key information for estimating traffic *impedance maps*, potentially in real-time

Aim of the work

To develop a reliable *methodology* able to perform the preliminary analyses needed for computing the *impedance maps from FCD*

- *management* and *visualization* of a *huge data amount*
- preliminary tests for *projecting* the *raw FCD* to the *route lines*
Features of the analysed FCD

The analysed FCD:

▶ acquired in the month of April 2017 by the on board units installed on the vehicles of the Turin Public Transportation company (Gruppo Torinese Trasporti - GTT)

▶ include the pairs of WGS84 geographical coordinates (longitude, latitude) along with a set of attributes (vehicle code, line code, turn, timestamp, ecc.)
  ▶ variable acquisition rate (from few to tens of seconds)

▶ provided in the CSV format
  ▶ the original file is very heavy (2.19 GB)
  ▶ converted in a database through a Python script based on the sqlite3 and pandas libraries
# Database generation

About **30,000,000** records!
Velocity analysis

- The **FCD** were **organized** for **lines**, then for **vehicles** and finally they were chronologically ordered.

- For every line of the transportation network:
  - the *Vincenty* formula was used to compute the **planimetric displacement** $\Delta s$ between **two positions** of the specific vehicle in two **consecutive time moments**;
  - the **velocities** were computed as $v = \frac{\Delta s}{\Delta t}$. 
Velocity analysis

The computed **velocities** were represented as **arrows** and plotted on top of the Turin drive network graph, automatically downloaded from Open Street Map through the OSMnx Python library.
Example of computed velocities

Line 11, vehicle 3063
Outlier removal

Before proceeding with the time analysis, the outliers were removed by eliminating all the records:

1. whose $\Delta t$ are higher than 99.5\textsuperscript{th} percentile and lower than 0.5\textsuperscript{th} (statistically not significant)
2. characterized by a velocity higher than 5 times the mean
Outlier removal

Before proceeding with the time analysis, the outliers were removed by eliminating all the records:

1. whose $\Delta t$ are higher than 99.5$^{th}$ percentile and lower than 0.5$^{th}$ (statistically not significant)
2. characterized by a velocity higher than 5 times the mean

![Histogram of dt values]
After the **outlier removal**, the reconstructed path follows more closely the actual line route: the **longest arrows**, probably due to the bus routes from and to the depot, are **eliminated**
Line 11: velocities

After the **outlier removal**, the reconstructed path follows more closely the actual line route: the **longest arrows**, probably due to the bus routes from and to the depot, are **eliminated**.
Temporal analysis

- Once the outliers were removed, a **temporal analysis** was performed.

- The data were divided into **working** and **weekend days**, considering the following **time intervals** during the day:
  - 0 - 5
  - 5 - 7
  - 7 - 9
  - 9 - 11
  - 11 - 13
  - 13 - 15
  - 15 - 17
  - 17 - 19
  - 19 - 21
  - 21 - 24
Line 11: time slot velocities in working days
Line 11: time slot velocities in weekend days
Considerations

▶ The **highest velocities** occur at **night** and in **late evening**, with a **local peak** shortly after the **lunch hour**

▶ The **lowest velocities** occur during the **peak hours**, in correspondence of the **office entrance** and **exit hours**

▶ The differences between **working** and **weekend days** are more **evident** in the **peak hour** time slots

▶ During the **0-5** and **21-24** time slots, the **difference** is **small**, since in these hours the **traffic** level is low also in the **working days**
FCD projection to line networks

A preliminary strategy was implemented to **assign** the **velocities to** the **line network topology**:

- for **every FCD point**, the **closest tree** of the specific line network is selected.
FCD projection to line networks

A preliminary strategy was implemented to assign the velocities to the line network topology:

- for every FCD point, the closest tree of the specific line network is selected
FCD projection to line networks

A preliminary strategy was implemented to **assign** the **velocities to** the line network topology:

- for every FCD point, the **closest tree** of the specific line network is selected.
FCD projection to line networks

A preliminary strategy was implemented to assign the velocities to the line network topology:

▶ for every FCD point, the closest tree of the specific line network is selected
Topological issues occur when the FCD point is located in a segment in which the distance between two (or more) arcs is comparable to the GNSS measurement errors.
Topological issues

It is rather improbable that the FCD point 4 and point 5 may be assigned to the tree 206-207 of the network, since the vehicle was located in the tree 77-78 few moments before.
Topological issues

A possible solution is to consider:

- the **temporal information** contained in the **FCD**
- the **cardinality information** contained in the **line network**

Select the segment closest to the previous selected tree
The **assignment errors** can be identified considering that:

1. the bus cannot travel back in time $t_{i+1} > t_i$
2. the bus cannot travel big distances in a short time interval $nodo_{t+1} < nodo_t + 20$
3. the bus cannot move in the wrong direction: $node_{t+1} > node_t$ (possible problem when a new lap begins)
Projection algorithm

The **temporal trend** of the **node IDs** must be **constant** or **increasing** with small slopes (constant or positive derivative)
Projection algorithm: error removal

For every projected point, if the **hypothesis number 2 is not verified**, the arc incorrectly selected is **removed** from the network together with the **following arcs** and the projection is newly performed.
Projection algorithm: error removal

For every projected point, if the hypothesis number 2 is not verified, the arc incorrectly selected is removed from the network together with the following arcs and the projection is newly performed.
Projection algorithm: error removal

For every projected point, if the hypothesis number 3 is not verified, the arc incorrectly selected is removed from the network together with the previous arcs and the projection is newly performed.
Projection algorithm: error removal

For every projected point, if the hypothesis number 3 is not verified, the arc incorrectly selected is removed from the network together with the previous arcs and the projection is newly performed.
Identification of a new lap

- A **new lap** can be identified on the basis of a **peak** in the trend of the **node IDs**
- Once the last point of the lap is identified, the following points are newly projected forcing the algorithm to consider only nodes with low values of IDs
Identification of a new lap

▶ A new lap can be identified on the basis of a peak in the trend of the node IDs

▶ Once the last point of the lap is identified, the following points are newly projected forcing the algorithm to consider only nodes with low values of IDs
Results

- The designed and implemented algorithm is quite effective
- Few assignment errors still remain, nevertheless a solution has already been designed and is under implementation

Problem in visually validating such huge amount of data
Conclusions

- A **first strategy** to analyse the **FCD** of the Turin Public Transportation system was implemented, in view of an automatic and possible real-time impedance map generation.

- A **huge amount** of **FCD** were processed to compute the **vehicles velocities**.

- A **visualization** approach based on Osmnx library was adopted.

- A preliminary **temporal analysis** was carried out.

- A method to **assign** the **velocities** to the line **network topology** was developed and successfully tested.
Further developments

- To refine the outlier removal process in order to all the *velocities not referable* to the *actual path* of the lines

- To *test* the developed *topological procedure* on all the *velocity data*, by checking the *effective reliability* and *real-time feasibility* of the designed methodology

- To *compute* the *impedence maps* and deliver the corresponding *metadata*

- To extend the developed methodology to other cities
Thank you for your kind attention!

This work was supported by **URBAN-GEO BIG DATA**, a Project of National Interest (**PRIN**) funded by the Italian Ministry of Education, University and Research (**MIUR**) id. 20159CNLW8