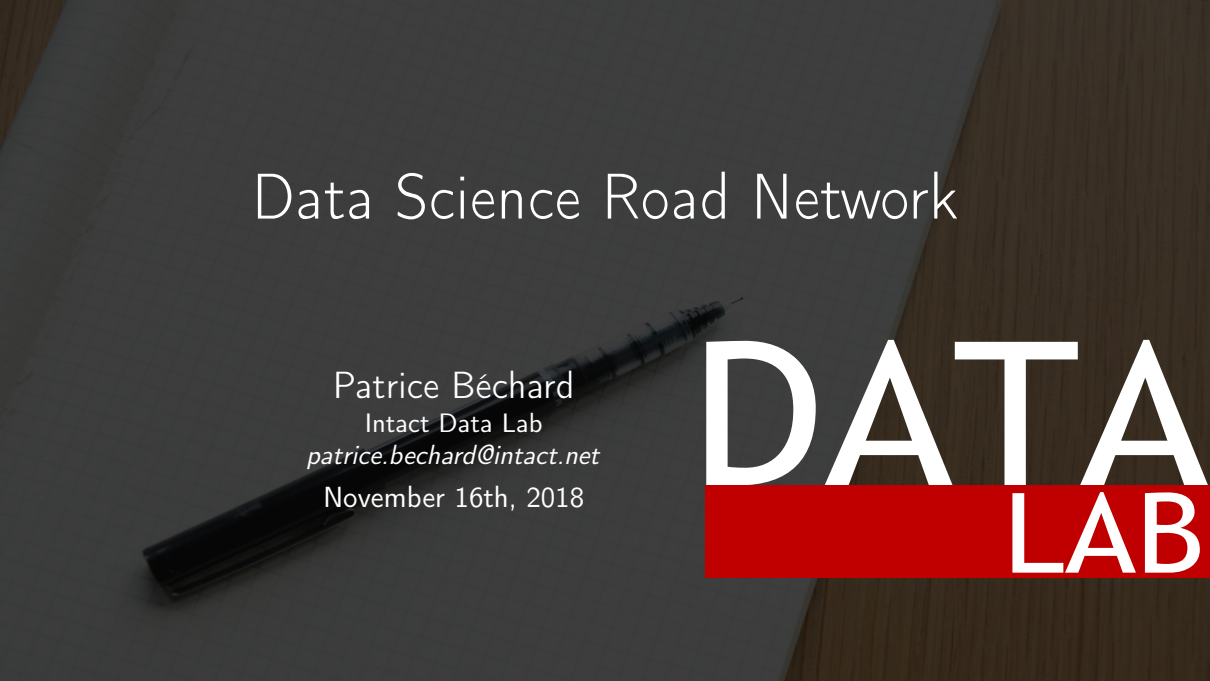
Data Science Road Network

Patrice Béchard
Intact Data Lab
patrice.bechard@intact.net

November 16th, 2018



Why is it an interesting problem ?

- Lots of GPS data from users
- May want to do things such as
  - Find patterns in the driving habits of users
  - Detect dangerous road sections
  - Optimize fastest route based on traffic
  - ...

Plan

Open Street Map

Building and visualizing road networks with OSMnx

GeoLife GPS Trajectories Dataset

Finding hotspots in Beijing

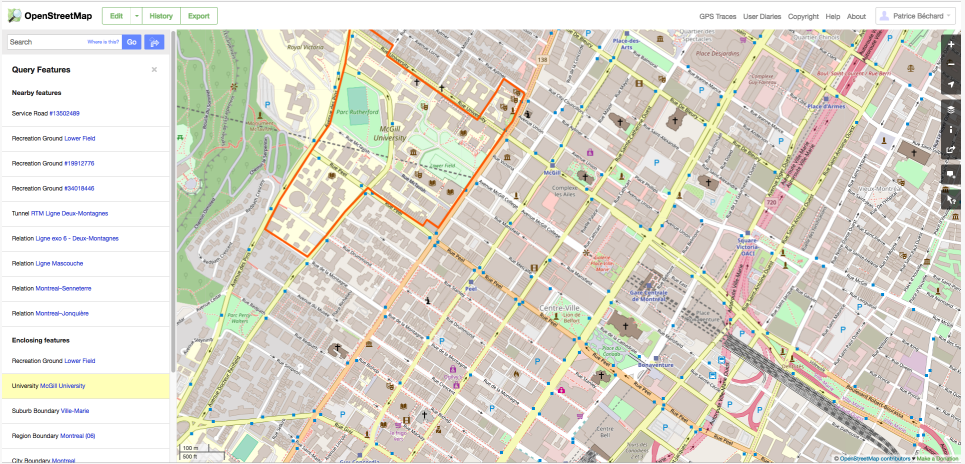
Predicting a driver's destination

Open Street Map (OSM) [4]

- Open-source map maintained by users
- Contains various informations about :
  - road segments
  - intersections
  - landmarks
  - ...
- Contains a routing engine similar to Google Maps
- https://www.openstreetmap.org/

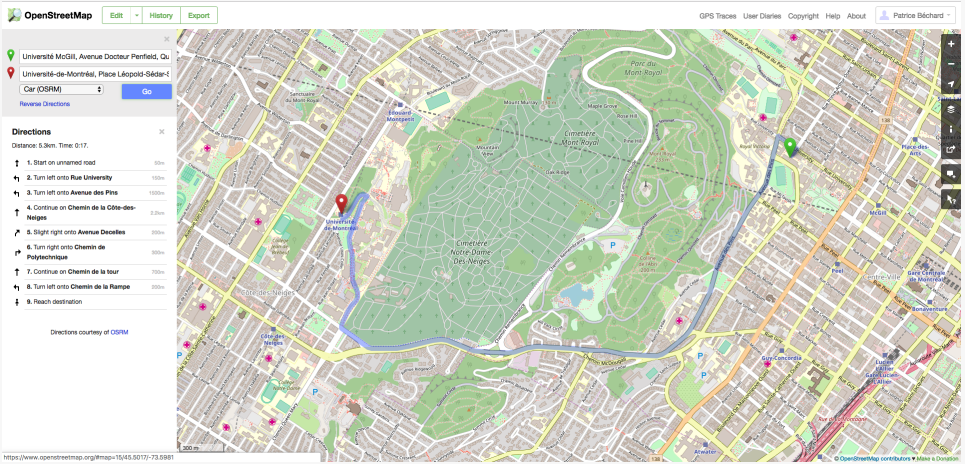
Open Street Map (OSM) [4]

Example : Querying features nearby



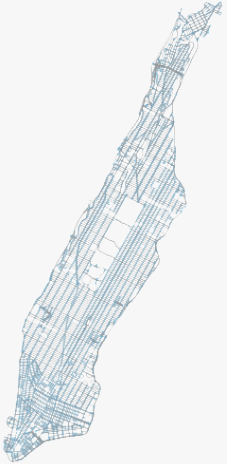
Open Street Map (OSM) [4]

Example : Find optimal route between two points



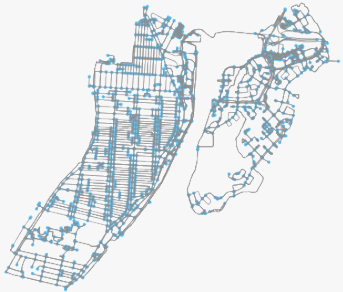
OSMnx [2]

- Open-source Python library
- Represents the road network as a directed
- Allows us to
  - Create the road network of a given location
  - Visualize this network easily
  - Simplify the road network by removing non-intersection nodes
  - Compute statistics about the road network
  - Find the shortest path between two nodes of the graph
  - ...
- https://github.com/gboeing/osmnx



Example : Creating the road network for Verdun

```
import osmnx as ox
G = ox.graph_from_place("Verdun, Montreal, Canada", network_type="all")
ox.plot_graph(G)
```



Example : Creating the shape of the Island of Montreal

```
import osmnx as ox
S = ox.gdf_from_place("Island of Montreal, Canada")
ox.plot_shape(S)
```

OSMnx [2]

Example : Creating a graph from a bounding box

```
import osmnx as ox
bbox = (45.52, 45.49, -73.55, -73.58)
G = ox.graph_from_bbox(bbox)
ox.plot_graph(G)
```

Example : Creating a graph from a single coordinate

```
import osmnx as ox
coord = (48.87378, 2.29504)
G = ox.graph_from_point(coord, distance=1000)
ox.plot_graph(G)
```

OSMnx [2]

Example : Finding the shortest path between two locations

```
import osmnx as ox
import networkx as nx

start_coord = (45.5049756, -73.5736905) # McGill University
end_coord = (45.5035380, -73.6176820) # Universite de Montreal
north, south, east, west = (45.5181450, 45.4854686, -73.5681800, -73.6279802)

G = ox.graph_from_bbox(north, south, east, west, network_type='drive')

start_node = ox.get_nearest_node(G, start_coord)
end_node = ox.get_nearest_node(G, end_coord)

route = nx.shortest_path(G, start_node, end_node)
ox.plot_graph_route(G, route)
```

Example : Finding the shortest path between two locations

For more examples and things to do with OSMnx, check out these links :

- https://geoffboeing.com/2016/11/osmnx-python-street-networks/ (overview)
- https://osmnx.readthedocs.io/en/stable/ (documentation)
- https://github.com/gboeing/osmnx-examples/ (more examples)

The GeoLife GPS Trajectories Dataset [8, 9, 10]

Dataset containing GPS trajectories from 181 users mostly around Beijing, China.

- **Number of unique trips** : 18,670
- **Total distance** : 1,292,951 km
- **Total duration** : 50,176 hours

For a full overview of the dataset :

- https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/User20Guide-1.2.pdf

Finding hotspots in Beijing

We can use trip origins and destinations to find where drivers tend to go while driving.

- We use the GeoLife GPS Trajectories Dataset
- We use the *Scikit-Learn* python library for the clustering [7]

Predicting a driver's destination

We can predict the destination of a driver based on its origin time and place.

References I

[1] D. Ashbrook and T. Starner, *Using gps to learn significant locations and predict movement across multiple users*, Personal and Ubiquitous computing, 7 (2003), pp. 275–286.

[2] G. Boeing, *Osmnx : New methods for acquiring, constructing, analyzing, and visualizing complex street networks*, Computers, Environment and Urban Systems, 65 (2017), pp. 126–139.

[3] A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, *Artificial neural networks applied to taxi destination prediction*, arXiv preprint arXiv :1508.00021, (2015).

[4] M. Haklay and P. Weber, *Openstreetmap : User-generated street maps*, Ieee Pervas Comput, 7 (2008), pp. 12–18.

References II

[5] J. Krumm and E. Horvitz, *Predestination : Inferring destinations from partial trajectories*, in International Conference on Ubiquitous Computing, Springer, 2006, pp. 243–260.

[6] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, *Automatic differentiation in pytorch*, (2017).

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., *Scikit-learn : Machine learning in python*, Journal of machine learning research, 12 (2011), pp. 2825–2830.

[8] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, *Understanding mobility based on gps data*, in Proceedings of the 10th international conference on Ubiquitous computing, ACM, 2008, pp. 312–321.

References III

[9] Y. Zheng, X. Xie, and W.-Y. Ma, *Geolife : A collaborative social networking service among user, location and trajectory.*, IEEE Data Eng. Bull., 33 (2010), pp. 32–39.

[10] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, *Mining interesting locations and travel sequences from gps trajectories*, in Proceedings of the 18th international conference on World wide web, ACM, 2009, pp. 791–800.