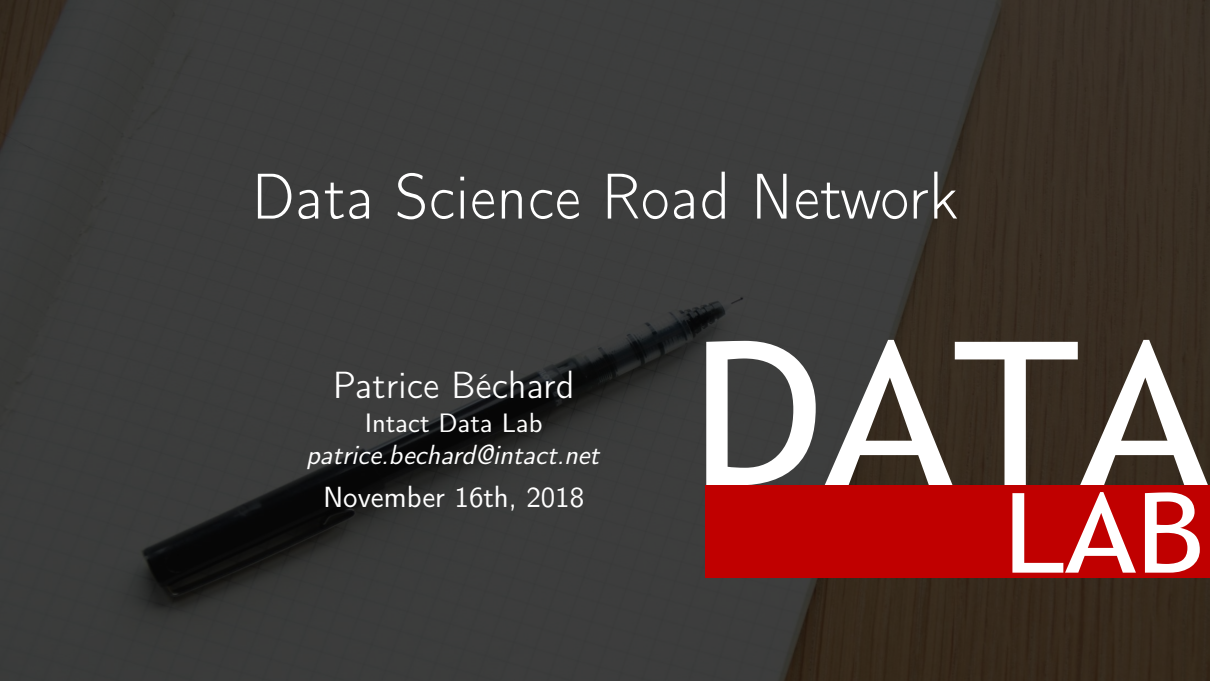
Data Science Road Network

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

Finding hotspots in the city

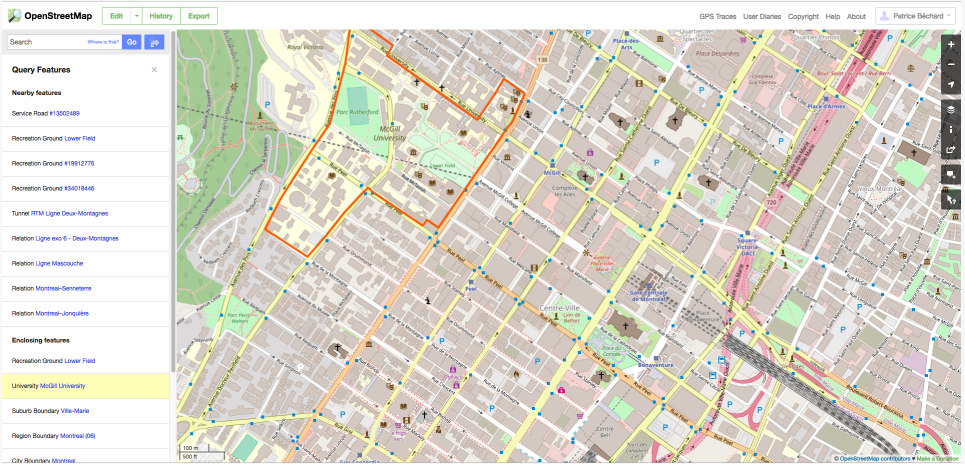
Predicting a driver's destination

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/

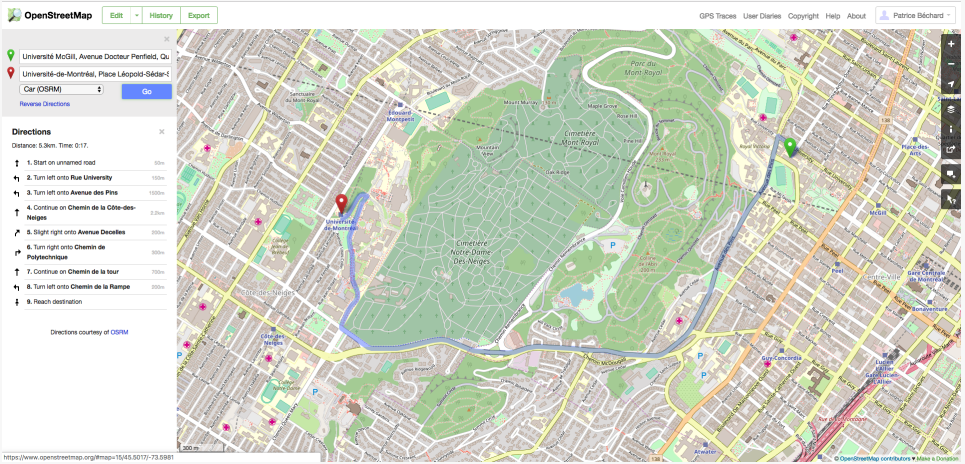
Open Street Map (OSM) [4]

Example : Querying features nearby



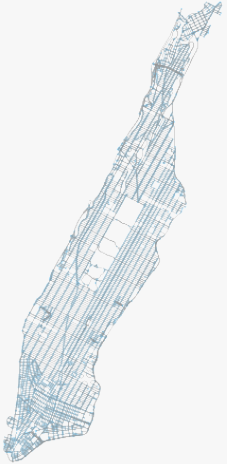
Open Street Map (OSM) [4]

Example : Find optimal route between two points



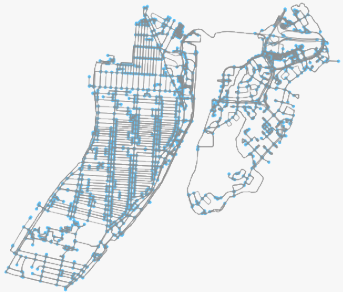
OSMnx [2]

- 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 the road network 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 the road network from a point

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

OSMnx [2]

Example : Computing statistics from the network

```
import osmnx as ox
G = ox.graph_from_address("Arc de Triomphe, Paris")
stats = ox.basic_stats(G)
```

```
{
    "circuity_avg": 1.0267881322837478,
    "edge_length_avg": 54.606206202850004,
    "edge_length_total": 130290.40800000011,
    "intersection_count": 990,
    "k_avg": 4.156794425087108,
    ...
}
```

Example : Computing statistics from the network

```
import osmnx as ox
G = ox.graph_from_address("Arc de Triomphe, Paris")
stats = ox.basic_stats(G)
```

```
{
    "circuity_avg": 1.0267881322837478,
    "edge_length_avg": 54.606206202850004,
    "edge_length_total": 130290.40800000011,
    "intersection_count": 990,
    "k_avg": 4.156794425087108,
    ...
}
```

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)

Finding hotspots in the city

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

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.