

Graph Powered Machine Learning - Exercise 1

Jörg Schad

September 21, 2020

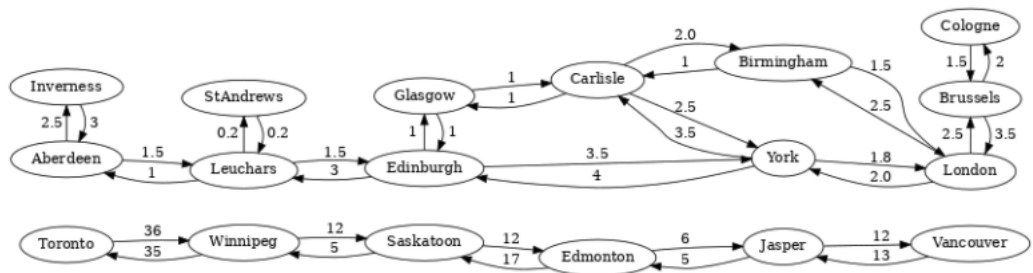
Thank you for participating in the Graph Powered Machine Learning Workshop! This is the first of two Exercises.

Logistics

- You can submit into groups up-to 4 people, please include:
 - Full Name
 - Student ID and Affiliation
 - Email Address
- for all group members
- Deadline: October 2nd, 23:59 CET
- Submit to joerg@arangodb.com
- Code can be either submitted in .py file or as .ipynb notebook (will give you the chance to add additional information)
- Feel free to use existing notebooks as starting point

1 Property Graph vs RDF/SPARQL

Recall the Train Network from the Graph Query Notebook.



Using rdflib (e.g., RDF and SPARQL Notebook)

1. Create an RDF Graph representing the same road network and travel times
2. Implement a SPARQL query returning all cities which can be reached from London. Bonus: all cities which can be reached within less than 5 hours. Hint: You might want to consider property paths.
3. Implement generic python code (i.e., the algorithms don't have to be specified in SPARQL, but could be) for the Single Source Shortest Path algorithm and return the shortest paths to all other cities starting from London. You can choose either Dijkstra's or Bellman-Ford's algorithm.

2 Pagerank

For a given directed networkx Graph (e.g., `G = nx.DiGraph(nx.path_graph(4))`) write a PageRank algorithm using python from scratch (i.e., don't use `nx.pagerank()`). Your PageRank algorithm should consider a parameter `alpha`, representing the damping factor and return a dictionary of nodes with their PageRank as value.