

Graph Embeddings

Development Workshop

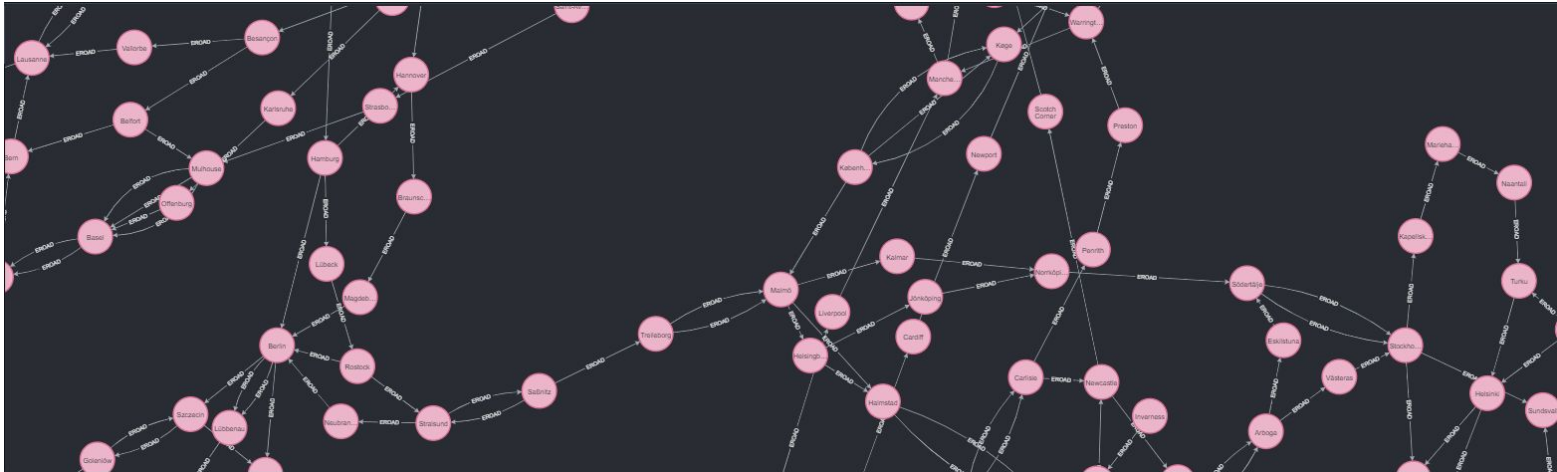
Filip Jachura
Klaudia Klepacka

Stages of work

1. Analysis of different types of graph embedding algorithms
2. The application for finding the most similar cities to the given ones based on graph embedding algorithms

Neo4j model

- 895 nodes (cities)
- 1250 edges (roads)

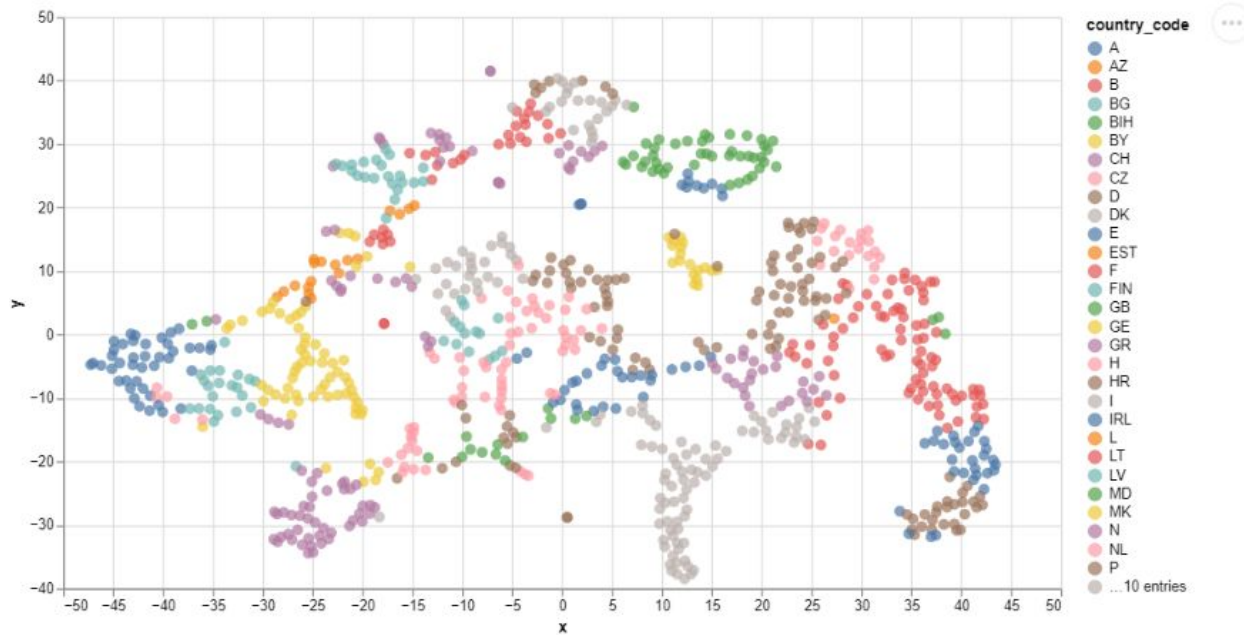


Embedding algorithms in Neo4j

Graph Data Science Library

- **Node2Vec** - computes a vector representation of a node based on random walks in the graph
- **FastRP** - uses a linear projection chosen in a random way
- **GraphSage** - uses node feature information to generate node embeddings on unseen nodes or graphs. Instead of training individual embeddings for each node, the algorithm learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood

The best results – Node2Vec 10D



Application for finding the most similar cities

Graph Embeddings

Warszawa

Kraków

Third city

Fourth city

Fifth city

Embedding algorithm

Node2Vec-100

SHOW THE MOST SIMILAR CITIES

Three most similar cities

City: Radom
Distance: 0.4703537612822087
City: Przemyśl
Distance: 2.100097514466581
City: Gliwice
Distance: 2.3759674709234086

Problems

- The documentation for embedding algorithms applies to the old version of neo4j

Run FastRP in stream mode on a named graph. Cypher

```
CALL gds.fastRP.stream(  
  graphName: String,  
  configuration: Map  
) YIELD  
  nodeId: Integer,  
  embedding: List of Float
```

```
: with driver.session(database="neo4j") as session:  
    fastRP_2d = pd.DataFrame(session.run("""CALL gds.alpha.randomProjection.stream({  
        nodeProjection: "ReferencePlace",  
        relationshipProjection: {  
            eroad: {  
                type: "EROAD",  
                orientation: "UNDIRECTED"  
            }  
        },  
        embeddingSize: 2,  
        maxIterations: 1  
    })  
    YIELD nodeId, embedding  
    RETURN nodeId, gds.util.asNode(nodeId).name AS name, gds.util.asNode(nodeId).country_code as countryCode, embedding;""")  
    fastRP_2d = fastRP_2d.rename(columns={0: "nodeId", 1: "name", 2: "countryCode", 3: "embedding"})
```

- It's not easy to obtain satisfying results for each algorithm

Thank you for your attention