Multi-Modal Community Detection with Multi-Weighted Graphs

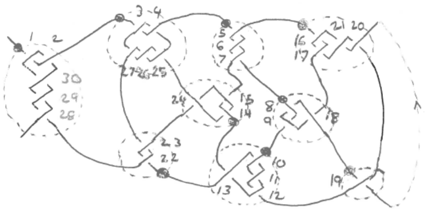
Nicholas Connolly

Research conducted through the NSF-MSGI program
at the USACE Geospatial Research Laboratory

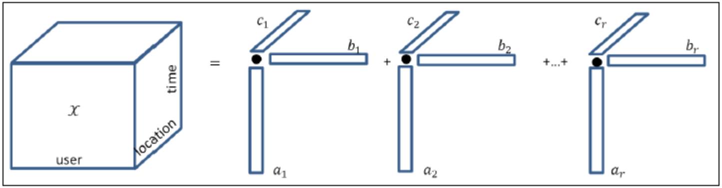
August 7, 2020

About Me (Nick Connolly)

- PhD candidate in mathematics at the University of Iowa
- Research in topology, knot theory, and graph theory
- About to start my final (sixth) year in graduate school



Introduction



[1]

- Program: NSF Mathematical Sciences Graduate Internship
- Mentor: Charlotte Ellison
- Project: Structure Embedding for Heterogeneous ST Data

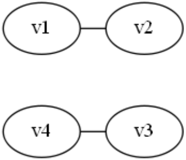
[1] C. Chen, C. L. Ellison, Z. J. Roth, and M. A. Simper. Boolean decomposition of spatiotemporal tensors. Technical report, ERDC Vicksburg United States, 2019

Project Goals

- Develop a **graph theoretic model** to describe trajectories with multi-modal attributes.
- Propose a method of searching for similarity between trajectories based on these attributes (**clustering**).

Graphs as Models

Example Graph

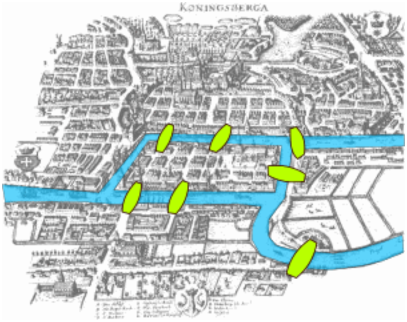


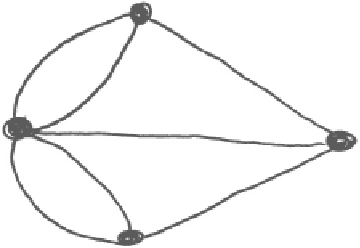
Adjacency Matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Graphs consists of **nodes/vertices** and **edges**.
- Graph models represent **things** (people, trajectories, etc.) with nodes and **relationships between things** with edges.
- A **weighted** graph assigns weights to each edge.
- Graphs can be represented by an **adjacency matrix**.

The First Graph Model: The Seven Bridges of Konigsberg (solved by Euler in 1736)[2][3]





[2] Wikipedia: Seven Bridges of Konigsberg

[3] Maarten van Steen, *Graph Theory and Complex Networks*

Application: Creating a Graph of Trajectories

Relationships between **spatio-temporal trajectories** can be modeled with a weighted graph:

- nodes represent trajectories (an entire time-series);
- edges represent similarity with respect to some attribute.

Examples of Attributes:

- number of time steps
- distance traveled
- average speed
- GPS location
- time of day
- day of week
- month of year

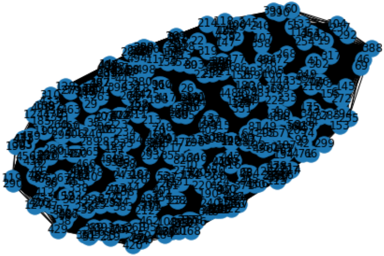
Examples of Trajectories: the Porto Taxi data set

Example: Comparing Trajectories by Total Time Steps

Raw Graph of Trajectories

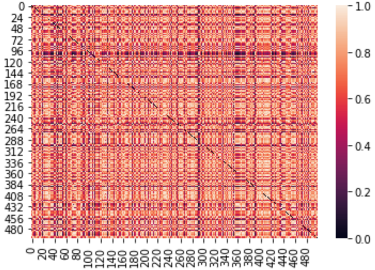
- Nodes are trajectories.
- Weighted edges measure similarity.

Not a helpful visualization!



Adjacency Matrix Heatmap

- The i, j^{th} entry measures the similarity between trajectories i, j.
- Light values $=$ high similarity.
- Dark values $=$ low similarity.



A Side Comment on Attribute Similarity

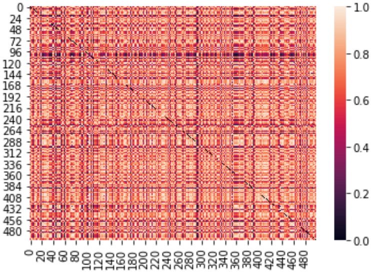
- You can compare using any attribute you want.
- You define your own formula based on the application.

Examples of formulas:

- Categorical: $\delta_{\text{kr}}(x_i, x_j)$
- Continuous: $1 - |x_i - x_j|$
- Cyclic: $1 - \frac{d(x_i, x_j)}{p/2}$
- Coordinate: $||\mathbf{x_i} - \mathbf{x_j}||_2$

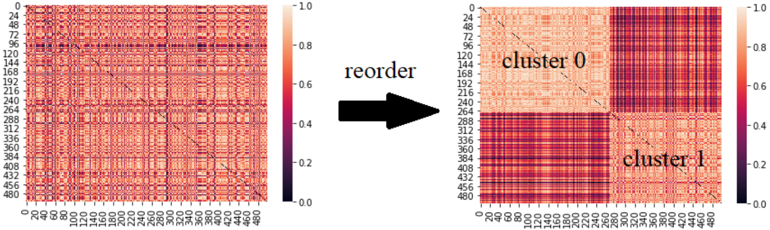
Total Time Steps :

$$a_{ij} = s(x_i, x_j)$$



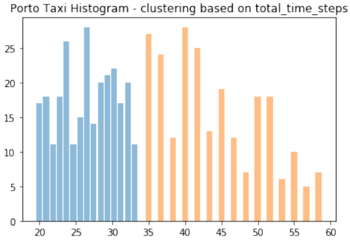
Clustering Trajectories: Graph Community Detection

- **Community detection** means clustering the vertices.
- CD algorithms for graphs are based on **modularity**.
- Sorting the rows and columns in the adjacency matrix based on communities yields a convenient visualization.



Comparing Clustered Trajectories: Total Time Steps

- The distribution of values is divided between clusters.
- We may also visualize the corresponding locations by cluster.

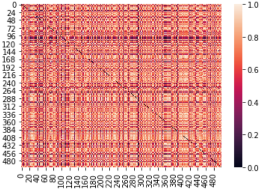


Histogram of time steps

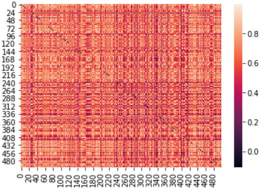
GPS coordinates plotted

How to Combine Multiple Attributes in One Model?

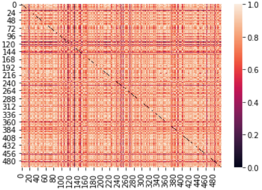
Total Time Steps



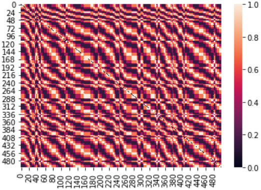
Average GPS Coordinate



Taxi Cab Distance

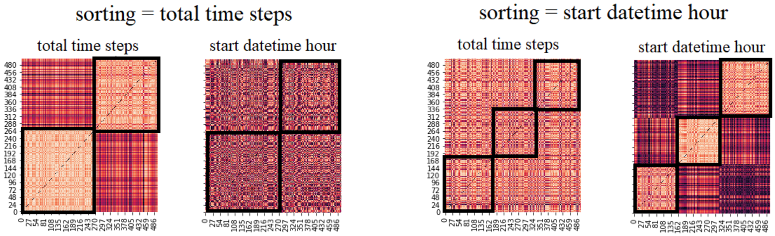


Day of Week

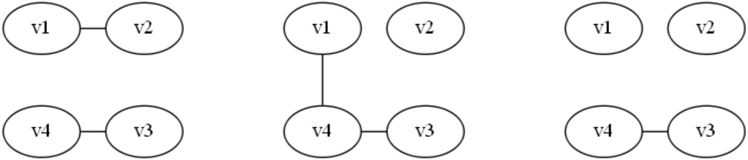


How to Compare Clusters between Attributes?

- Different attributes yield different clusters
- Attribute clusters can be compared by sorting heatmaps.
  - Similar attributes show visually distinct clusters.
  - Dissimilar attributes show noisy clusters.
- **Goal:** Cluster based on multiple attributes.



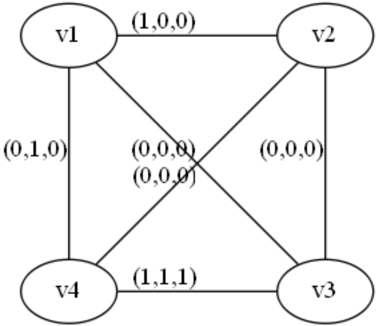
Theoretical Problem: Graphs on the Same Vertex Set

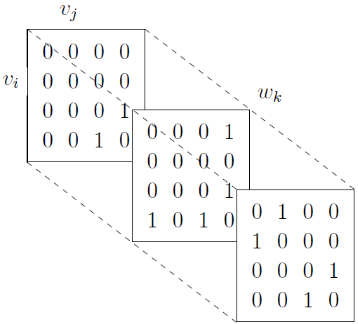


$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Multi-Weighted Graph

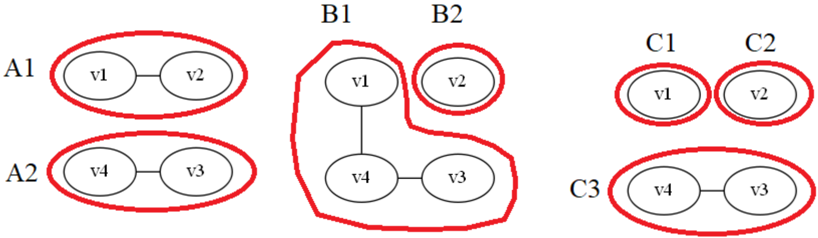
A **multi-weighted graph** is constructed from a sequence of weighted graphs on the same set of vertices (**component graphs**).





A **multi-weight function** $w : E \rightarrow \mathbb{R}^m$ assigns each edge a vector of weights (**composite edges**).

Community Detection for a Multi-Weighted Graph?

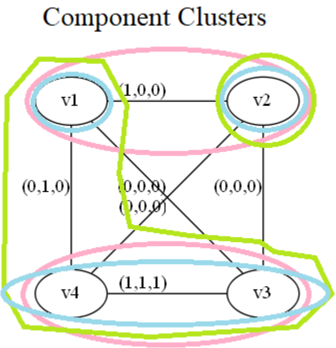


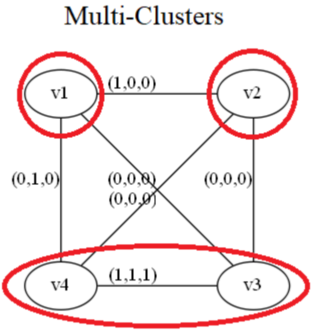
$$A_1 = \{v_1, v_2\}$$
$$A_2 = \{v_3, v_4\}$$

$$B_1 = \{v_1, v_3, v_4\}$$
$$B_2 = \{v_2\}$$

$$C_1 = \{v_1\}$$
$$C_2 = \{v_2\}$$
$$C_3 = \{v_3, v_4\}$$

Multi-Clustering: Non-Empty Intersections



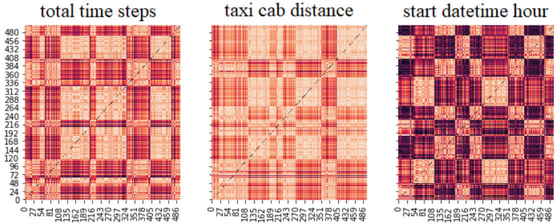


$$
\begin{aligned}
A_1 \cap B_1 \cap C_1 &= \{v_1\} \\
A_1 \cap B_2 \cap C_2 &= \{v_2\} \\
A_2 \cap B_1 \cap C_3 &= \{v_3, v_4\}
\end{aligned}
$$

Multi-Clustering Porto Taxi Trajectories

Example: {short duration} ∩ {long distance} ∩ {morning}

sorting = multi-cluster



Advantages

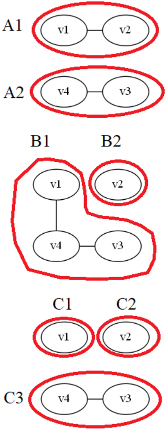
- simple model
- visible across attributes
- easy to interpret

Disadvantages

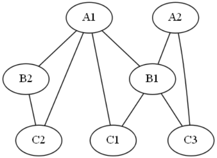
- restrictive model
- very poor scaling
- Attributes are kept separate

The Meta Graph: Combining Component Graph Clusters

Component Graph Clusters

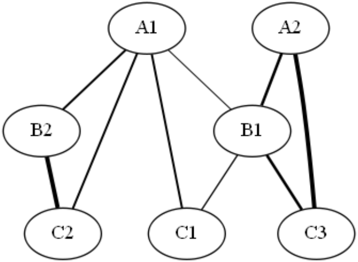


Meta-Graph



- Each cluster from each component is a vertex.
- Connect overlapping clusters with edges.
- **Jaccard score** determines edge weights.
- Meta-graph is k-partite.

The Meta Adjacency Matrix



	A_1	A_2	B_1	B_2	C_1	C_2	C_3
A_1	0	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
A_2	0	0	$\frac{2}{3}$	0	0	0	1
B_1	$\frac{1}{4}$	$\frac{2}{3}$	0	0	$\frac{1}{3}$	0	$\frac{2}{3}$
B_2	$\frac{1}{2}$	0	0	0	0	1	0
C_1	$\frac{1}{2}$	0	$\frac{1}{3}$	0	0	0	0
C_2	$\frac{1}{2}$	0	0	1	0	0	0
C_3	0	1	$\frac{2}{3}$	0	0	0	0

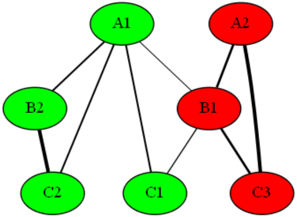
- The **Jaccard** score between two sets is the proportion of elements they have in common.

$$J(A_2, B_1) = \frac{|A_2 \cap B_1|}{|A_2 \cup B_1|} = \frac{|\{v_4, v_3\}|}{|\{v_1, v_3, v_4\}|} = \frac{2}{3}$$

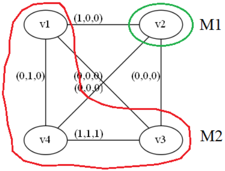
- Clusters from the same partition are automatically disjoint.

Meta Clustering: Cluster the Clusters

Clusters in the Meta Graph



Meta-Clusters in the Multi-Weighted Graph

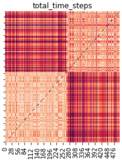


$$
\begin{aligned}
M_1 &= \{A_1, B_2, C_1, C_2\} \\
&\supseteq \{v_1, v_2\} \\
M_2 &= \{A_2, B_1, C_3\} \\
&\supseteq \{v_1, v_3, v_4\}
\end{aligned}
$$

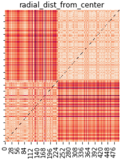
$$
\begin{aligned}
J(v_1, M_1) &:= J(A_1 \cap C_1, M_1) \\
&= \frac{1}{2} \\
J(v_1, M_2) &:= J(B_1, M_2) \\
&= 1
\end{aligned}
$$

Porto Taxi Attribute Clusters

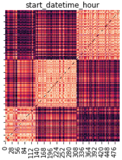
Total Time Steps



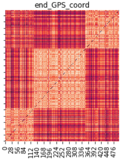
Radial Distance



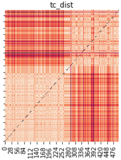
Hour of Day



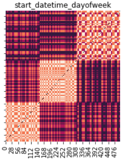
End GPS Coord



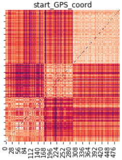
Taxi Cab Distance



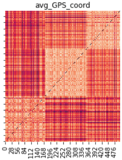
Day of Week



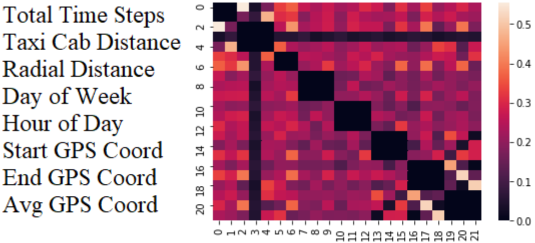
Start GPS Coord



Avg GPS Coord



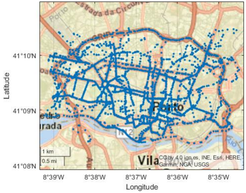
The Porto Taxi Meta Graph



- The meta graph contains 22 vertices (one per cluster).
- The meta graph is 8-partite (one per attribute).
- The dark squares along the diagonal match each attribute.

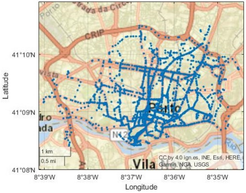
Meta-Clustering Porto Taxi Trajectories

Meta-Cluster 0:
176 Trajectories



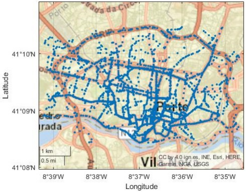
Attributes:
(total time steps, 0)
(tc dist, 0)
(radial dist, 1)
(day of week, 2)
(hour of day, 2)
(start GPS coord, 2)
(end GPS coord', 1)
(avg GPS coord', 1)

Meta-Cluster 1:
161 Trajectories



Attributes:
(total time steps, 1)
(tc dist, 2)
(radial dist, 0)
(day of week, 1)
(hour of day, 1)
(start GPS coord, 1)
(end GPS coord, 2)
(avg GPS coord, 2)

Meta-Cluster 2:
163 Trajectories



Attributes:
(tc dist, 1)
(day of week, 0)
(hour of day, 0)
(start GPS coord, 0)
(end GPS coord, 0)
(avg GPS coord, 0)

Meta-Clustering Porto Taxi Trajectories

Advantages:

- combines attributes
- scales well
- independent of component clustering method

Disadvantages

- less easily interpreted
- requires informed user to draw conclusions
- dependent on clusters

Thanks!

- Thank you to the NSF for funding my research!
- Thank you to Charlotte for mentoring me!
- Thank you to everyone at USACE Geospatial Research Lab for hosting me (remotely)!