Enhancing Synthetic Transaction Data Generation with Graph Clustering

G. Leal Cardoso Pita[1] Y. Lawryshyn[2]

[1]Department of Engineering Science
University of Toronto

[2]Department of Chemical Engineering
University of Toronto

Complex Networks in Banking and Finance, June 2024

Table of Contents

Relevance of the problem

Bank Transaction Data (account-level credits and debits)

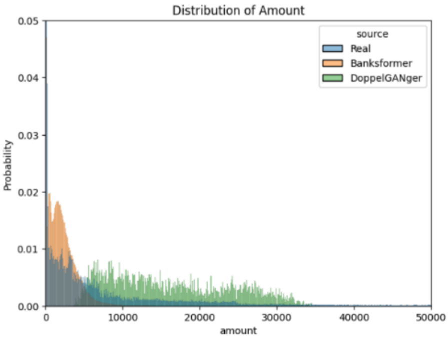
- Highly useful, e.g. to train fraud detection or product recommendation models.
- Highly sensitive, with PII on spending habits; as little as four credit card transactions can be enough to de-anonymize (de Montjoye et al., 2015).

Account 1						
Day	Cash Withdrawal	Collection from Another Bank	Credit Card Withdrawal	Credit in Cash	Interest Credited	Balance
1	0	0	0	700	0	700
2	0	0	0	7268	0	7968
3	0	0	0	14440	0	22408
⋮	⋮	⋮	⋮	⋮	⋮	⋮
432 rows elided						

Account 2						
Day	Cash Withdrawal	Collection from Another Bank	Credit Card Withdrawal	Credit in Cash	Interest Credited	Balance
1	0	0	0	1800	0	1800
2	0	0	0	1800	0	3600
3	2414	0	0	0	0	1186
⋮	⋮	⋮	⋮	⋮	⋮	⋮
713 rows elided						

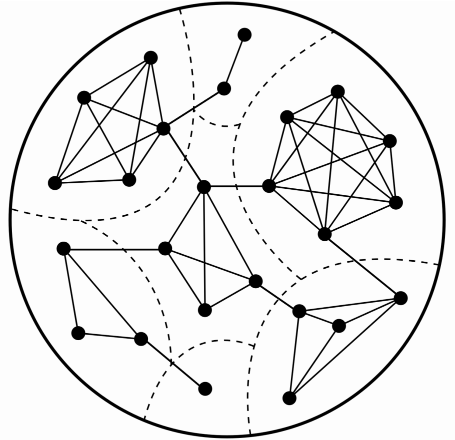
Current Work and Hypothesis

- **Current models** use GANs (e.g. DoppelGANger) and transformers (e.g. Banksformer) and work to a certain extent, but there is room for improvement in the periodicity or value distributions, for example (Liu, 2023).



Current Work and Hypothesis

- **Idea**: **separate** real, training, data into clusters, **generate** new data based on **each section** individually.
  - **Less variability**.
  - **Potentially closer to reality!**



Why use graphs?

Advantages

- No need for previous knowledge on client behaviours.
- Robustness against bad choices of summary statistics of transaction series, like frequency of transactions or average amount transacted.

Problem

How do we build edges between nodes?

Data overview

- Not many open data sources for banking transactions, ones available are old and not representative or current trends. With that in mind...
- Use **real open-source transaction data** from Czech Banks in the 90s. https://data.world/lpetrocelli/some-translatedreformatted-czech-banking-data

Column	Description	Property	Features
operation	Mode of transaction	Categorical	**Categories (num. entries)** Cash Withdrawal: 434918 Remittance to Another Bank: 208283 Credit in Cash: 156743 Collection from Another Bank: 65226 Credit Card Withdrawal: 8036
amount	Transaction amount	Numerical (Czech koruna)	Min: 0.0 Mean: 5924.15 Max: 87400.0 Standard Deviation: 9522.74
balance	Account balance after transaction	Numerical (Czech koruna)	Min: -41125.7 Mean: 38518.33 Max: 209637.0 Standard Deviation: 22117.87

Data overview

Methodology

- Group transactions by accounts as time series, categorical data flipped to columns. Every account becomes a node.
- Add edges between accounts, using combinations of account features and distance measures, to be explored soon.
- Measure the *clusterability* of the resulting graph with the Eigengap heuristic.

Creating nodes and edges

- With every node being an account, we can calculate the distance between nodes using, for example, Dynamic Time Warping (DTW) or Compression-based Algorithms.
- Once we have the distances, edges are added between nodes if they are below a certain threshold. In our case, we tested multiple thresholds: median distance +/- 0, 1, 2 and 3 standard deviations.

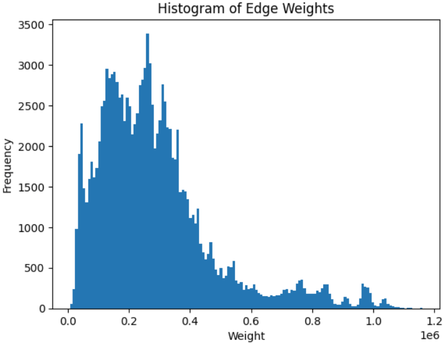
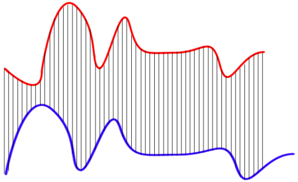
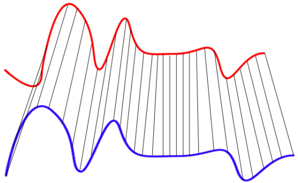


Figure: Distribution of edge weights, using only amount value and DTW

Dynamic Time Warping

- DTW is a technique used to measure the dissimilarity between two time series that may vary in time or speed.
- Expandable to multivariate time series, by combining the distances for each column (Petitjean et al., 2011).


Euclidean Matching


Dynamic Time Warping Matching

Compression-based distance algorithms

Compression-based algorithms leverage the file compressing power of regular compression algorithms to measure how many times larger is the compressed file of both series concatenated, compared to compressing each series individually and summing their sizes.

```
function Dist = CDM(A,B)
save A.txtA–ASCII                                   % Savevariable A as A.txt
zip('A.zip', 'A.txt');                              % Compress A.txt
A_file = dir('A.zip');                              % Get file information
save B.txt B –ASCII                                 % Save variable B as B.txt
zip('B.zip', 'B.txt');                              % Compress B.txt
B_file = dir('B.zip');                              % Get file information
A_n_B = [A; B];                                     % Concatenate A and B
save A_n_B.txt A_n_B –ASCII                         % Save A_n_B.txt
zip('A_n_B.zip', 'A_n_B.txt');                      % Compress A_n_B.txt
A_n_B_file = dir('A_n_B.zip');                      % Get file information
                                                    % Return CDM dissimilarity
dist = A_n_B_file.bytes / (A_file.bytes + B_file.bytes);
```

Figure: Compression-based dissimilarity measure (Keogh et al., 2007)

The Eigengap Heuristic: Math Background

Recall,

The graph of N time series can be represented by the adjacency matrix A such that

$$\begin{aligned} \underset{(N\times N)}{A} &= [w_{ij}]\,, \\ w_{ij} &\in \{0,1\}, \\ w_{ii} &= 0,\ \forall i, \end{aligned} \tag{1}$$

with $w_{ij} = 1$ if the distance d_{ij} between nodes i and j is smaller than the threshold chosen, $w_{ij} = 0$ otherwise.

The Eigengap Heuristic: Math Background

Then,

The normalized Laplacian matrix $\mathcal{L}$ is defined such that

$$\text{defining } \mathbf{D} = \begin{bmatrix} \mathbf{d}_1 & 0 & \dots & 0 \\ 0 & \mathbf{d}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{d}_n \end{bmatrix}, (\text{where}, \mathbf{d}_i = \sum_j w_{ij}), \tag{2}$$

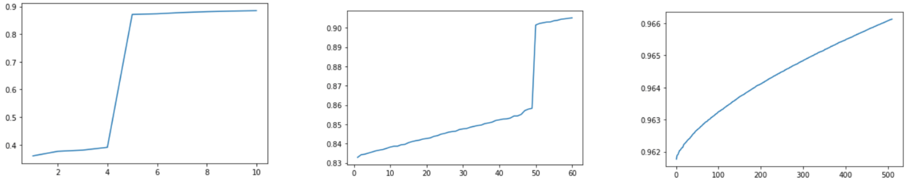
$$\text{and } L = \mathbf{D} - A,$$

$$\text{yields } \mathcal{L} = \mathbf{D}^{-1/2} L \mathbf{D}^{-1/2}.$$

Finally,

If A is fully connected, the eigenvalue λ_0 is 0; all other eigenvalues are in $(0, 2]$ (Chung, 2001). **Eigengap heuristic**: the ideal number of clusters for the graph is around the index of the first spike in eigenvalues.

The Eigengap Heuristic: Math Background



(a) N=250 **(b)** N=2500 **(c)** N=25000

Figure: Plots of eigenvalues of adjacency and Laplacian matrices for varying number of nodes (Miasnikof et al., 2024)

The Eigengap Heuristic: Clusterability interpretation

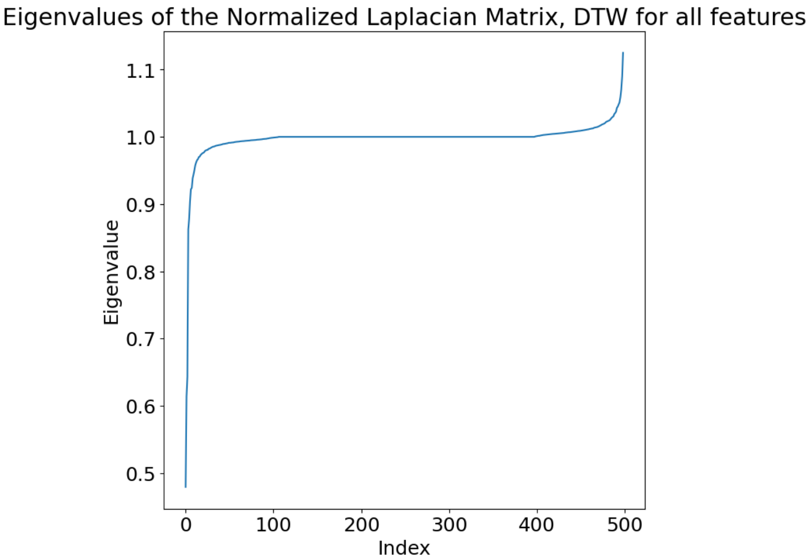
Idea

Although the original point of the heuristic is to yield the ideal number of clusters, in the case of the index being 1 or $N-1$, we can interpret it as a *clusterability* metric: the graph is unclusterable.

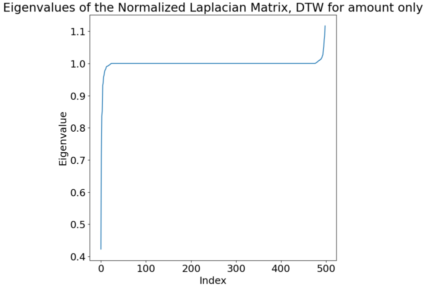
Table: Index of largest spike of eigenvalues of $\mathcal{L}$

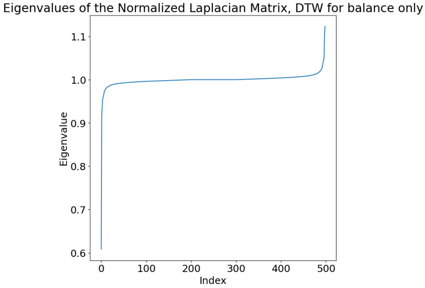
		Standard deviations from the mean distance						
		-3	-2	-1	0	1	2	3
DTW	all features	N/A	N/A	N/A	N/A	3	2	1
	amount only	N/A	N/A	N/A	N/A	N/A	1	1
	balance only	N/A	N/A	N/A	1	1	1	1
Compression	amount only	N/A	N/A	N/A	1	498	498	498
	balance only	N/A	N/A	N/A	1	1	1	1

Eigenvalues of the normalized Laplacian matrices for DTW

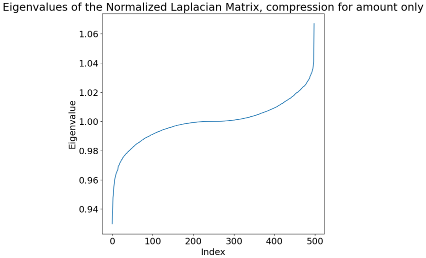


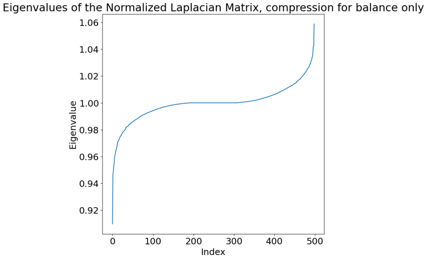
Eigenvalues of the normalized Laplacian matrices for DTW





Eigenvalues of the normalized Laplacian matrices for Compression-based





Clusterability conclusion and thoughts

- Training data appears to be unclusterable, but this conclusion depends entirely on the original dataset and how the tabular data was converted into a graph.
- Current and future work should focus on similarity metrics and edge representation on this and other datasets. The clusterability measure using the Eigengap heuristic could also be used to determine the fidelity of synthetic transaction data.

Acknowledgements

I would like to thank everyone at the CMTE group and RBC for the valuable conversations we had during our weekly meetings. I would also like to express special thanks to Professor Yuri Lawryshyn, Dr. Lucy Liu, and Dr. Pierre Miasnikof for all your mentorship and support throughout this research.

Bibliography

Chung, F. R. K. (2001). Lectures on spectral graph theory.

de Montjoye, Y.-A., Radaelli, L., Singh, V., and Pentland, A. (2015). Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539.

Keogh, E., Lonardi, S., Ratanamahatana, C. A., Wei, L., Lee, S.-H., and Handley, J. (2007). Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14(1):99–129.

Liu, A. (2023). *Generating Synthetic Transaction Data with Deep Learning Models*. University of Toronto, Toronto.

Miasnikof, P., Shestopaloff, A. Y., Bravo, C., and Lawryshyn, Y. (2024). Empirical study of graph spectra and their limitations. In Cherifi, H., Rocha, L. M., Cherifi, C., and Donduran, M., editors, *Complex Networks & Their Applications XII*, pages 295–307, Cham. Springer Nature Switzerland.

Petitjean, F., Ketterlin, A., and Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693.