Graph Matching on the Patterns of Life

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Bryan Kim

Fall, 2021

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Daniel Graham, Department of Computer Science

Rosanne Vrugtman PHD, Department of Computer Science

Abstract
Though criminals constantly change their phone numbers, making it difficult for law enforcement to trace them, they rarely change their contacts. The process of re-identification can assist law enforcement efforts to trace the identities of unknown individuals by using graph networks from the interactions between the old and new phone numbers. The prototype pipeline I designed during my internship can extract call and message records from recorded logs and convert them to a graph network, creating a way to identify similar networks. The pipeline was tested using data found from Kaggle as well as some sample data that represented the client data. The outcome from the tests shows a promising 8.25% equal error rate in terms of correctly identifying each network with the clients satisfied with the results. This concept can be applied to a wide variety of fields including studies with brain networks and even social media data for relevant pandemic tracing. Next steps for the project should be to implement machine learning models to the pipeline like Graph Neural Networks to reduce the very intensive calculations and find ways to include social media data in the graph networks.

1 Introduction

When a person replaces their phone number, the only aspect that changes in the person's life is the number itself as their contact numbers and their social circles will most likely remain consistent. With the same social group kept around this person, there can be a noticeable number of repeated interactions between them through their phones. A certain pattern of life can be found and mapped out as the repetition of events continue, limited by the scope of the interactions done with the phone number. With every individual having a different pattern of life in some way or another, such patterns can be used as key identifiers regardless of the phone number in use.

This is one of many cases of re-identification. The main idea of this concept is to take anonymized yet unique data and identify the individuals who own it by matching the data with entries from fully identified databases. Such databases would include other instances similar to the anonymized data but with an identifiable label. With the scenario above, the pattern of life extracted from the interactions from the phone would be the anonymized data and the name of the owner for the phone will then be entries from a known database.
One method for storing such complex networks is to use the graph data type. Consisting of nodes representing various entities and edges setting relationships between those nodes, graphs are able to handle containing such non-linear data. There are attributes within the nodes themselves to add unique characteristics and weights to the different nodes, allowing for graphs to be unique.

Law enforcement often faces the problem of reidentification with phone numbers in their cases as the suspects would be swapping their phone numbers, resulting in the difficulty of identifying who is who when looking at call and messaging logs. Currently there is no working solution to this issue, so during my internship I was tasked to create a prototype data pipeline for the client that would extract an interaction-based graph network from call and message logs and compare between various networks to find the most probable match to a specific phone number.

2 Review of Research

The need for graph matching was vital for the solution of the project, resulting in research in that area. This leads to an important piece of work that is actually implemented in the pipeline. The NetLSD package, created by Tsitsulin and his team, is a Network Laplacian Spectral Descriptor that extracts unique descriptors based solely on the graph's structure and allows for straightforward comparisons of large graphs, outperforming in efficiency and expressiveness [1]. This powerful tool is used within the pipeline to provide a point of comparison for the large interaction networks generated from the phone numbers.

Peter and Francois provide an overview on the concept of graph matching and the various techniques using different distance measures [2]. Though the techniques in this source were not used for the actual calculations in the pipeline, it served as a good starting point as the authors go over topics that are found in other sources such as spectral distances and explaining them in a simplified manner.

Another relevant work that provides more context to the data was published in 2015 by a team consisting of Blondel, Decuyper, and Gauiter. The three go over the analysis of social graph networks extracted from anonymized data accumulated over the past decade and explore the vast array of ways this data can be used. This includes urban sensing, tracking of epidemics and geographical partitioning [3]. Much of the information found in the trio's work provided the possibility of where to take the project next as well as providing suggestions on how to format the graphs in the pipeline as well.

3 Project Design

To understand the data pipeline, it's important to understand the project's constraints and the technology used. The client required that the prototype for the pipeline in Python3 and run as a script on a command line. Cloud services were not recommended due to the sensitivity of the data. The pipeline should be able to parse in CSV files provided. Also, the data used by the client were not labeled, so machine learning algorithms like Graph Neural Networks that require labeled data were not feasible. From the sample data used to test the pipeline, each CSV contains call and message logs of one phone number with each entry containing the date, source phone number and destination phone number.

Python includes various data analysis packages that are used, including Pandas and Numpy. NetworkX is another Python package that contains the Graph data structures and graph distance algorithms that is used to create the networks. NetLSD, discussed in the Review of Research section, is also used alongside the distance algorithms for the calculations for determining graph matches. Matplotlib was also used for visualizations of the graphs.

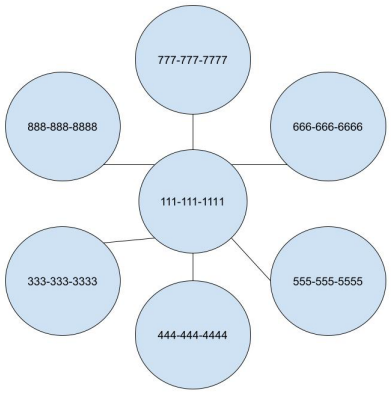

Figure 1: An example network

The pipeline has 3 distinct stages: parsing in CSV files, extracting data into graph networks, and finally calculating the distances from other networks. In the first stage, the CSV files are formatted in a specific way that a simple Python could be used to parse in data into a Pandas Dataframe. The second stage runs a function developed to loop through the Dataframe and create a NetworkX graph with all the phone numbers as nodes and the interactions between the numbers as edges.

An example of an extracted graph network is shown in Figure 1 where the source phone number is 111-111-1111 and the adjacent nodes represent the numbers interacting with the source number. The final stage compares the newly created network to a list of precompiled graph networks, which will be referred to as PG, to determine the most probable match. This stage uses a combination of two distance measures, Graph Edit Distance from NetworkX and the NetLSD, using a simple sum. The Graph Edit Distance returns the number of changes required to convert from one graph to another where changes include adding or removing nodes, attributes, and edges. This stage runs both comparison algorithms to the new network with PG. The values are normalized and added together. This stage then returns a Dataframe with each entry containing both graphs compared and the sum of the normalized distances. The most likely to match will have the lowest distance sum score.

Challenges during this project include having no prior experience with the concept of Graph Matching, a fellow intern not well versed with Python, and having to work remotely due to Covid-19. Reading papers and frequently asking questions of the managers was necessary in order to understand the concepts of graph matching, bringing us interns to the same page. Also, assisting other team members and myself with Python ~~and~~ eventually led my fellow intern to be a sufficient coder near the end of the internship. The pandemic also hindered work schedules as the same intern was located across the country, and the rest of the team was out of state as well. However, the team members provided a welcoming online experience, resulting in ~~the team~~ overcoming the distance and time zones and growing as a unit.

4 Results

The metric used to measure the performance of the data pipeline was the equal error rate. This is the location on the Receiver Operator Characteristic Curve, a plot that shows the diagnostic ability of a binary classifier system, where the false acceptance rate is equal to the false rejection rate, with a lower score resulting in higher accuracy. The pipeline had an 8.25% equal error rate which means it has a good accuracy score that could most definitely be optimized with further adjustments. Stage 3 of the pipeline takes most of the run-time, mainly due to NetworkX's Graph Edit Distance. This algorithm is a NP-Hard problem that can't be optimized by a significant amount. Thus the pipeline requires 10-15 minutes to complete its output. However, with a fairly low equal error rate, the client was satisfied with the outcome and pushed for further research into this project.

5 Conclusion

This data pipeline was created with a simple Python script to help a law enforcement client with reidentifying unknown phone numbers based on patterns of life. It is important to remember that the purpose of this pipeline is solely for law enforcement cases rather than commercial use so lawful citizens would not have to worry about their privacy. This technology could also be used for other industries such as pandemic tracking and brain neuron mapping ~~as well~~.

6 Future Work

Future work includes optimizing the combination of the two distance measures, finding ways to utilize Graph Neural Networks, and using more in-depth data such as social media data. The current method of combining the Graph Edit Distance and NetLSD is a simple sum, which could be optimized to a better combination. This can be done with adding weights adding more attributes to the nodes in the networks to help give more distinguished comparisons. In the long run, Graph Neural Networks can be implemented to replace the inefficient Graph Edit Distance algorithm with most of the work going to labeling the data ~~that was~~ missing from the beginning of the project. This pipeline can be modified and used to experiment with different types of data as well. Exploring rich data like social media can help improve the performance of the pipeline.

7 UVA Course Evaluation

Many of the higher-level courses at UVA promote large amounts of teamwork and project experience that had been applied to this project. Such classes include CS 4750 (Database Systems), CS 3240 (Advanced Software Development), and CS 4640 (Web Programming Languages). Along with teamwork and time management skills, I gained experience with using Big Data and the technology with it in CS 4774 (Machine Learning) that prepared me for a main part of this project.

However, none of my coursework really prepared me to analyze research papers effectively and this took too much time to read during my internship. Providing more hands-on experience with research papers would probably be the one change I would suggest to the UVA CS curriculum.

References

[1] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alex Bronstein, and Emmanuel Müller. 2018. NetLSD: Hearing the Shape of a Graph. In KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219991

[2] Peter Wills and Francois G. Meyer. 2019. Metrics for graph comparison: A practitioner's guide. (December 2019). Retrieved October 21, 2021 from https://arxiv.org/abs/1904.07414

[3] Vincent D. Blondel, Adeline Decuyper, and Gautier Krings. 2015. A survey of results on mobile phone datasets analysis. (February 2015). Retrieved October 21, 2021 from https://arxiv.org/abs/1502.03406