[DRAFT] Entity Linkage in Evolving Temporal Graphs

Data Intensive Systems Course, Utrecht University, 2024-2025

Instructor: Prof. Yannis Velegrakis

DEADLINE: Nov 7th, 2024 **Number of Persons: max 3**

Situation Description:

A telecommunication company is recording data about its clients. The **clients** at different points in time call each other by phone. Each **phone cal**l has a duration. The company is recording information on who has called whom, and for each call that have been performed it keeps the **starting** and the **ending** time. The phone calls are always between two clients (no three-way or more calls are allowed). Each client has in possession more than one phones (devices). This means that it is possible for a client to call someone even if they are already on a call with another client. For instance, at 3:00AM client A calls client B, and they keep talking until 7:00. But at 4:30, the client A is also calling client C with whom they keep talking until 5:00. (so from 4:30 to 5:00, the client A talks in parallel to client B (through the first call) and to client C (through the second call). At 5:30, client A calls again person C and they keep talking until 10:00 (meaning that they hang up the phone after the person A hangs up the phone with B (which happened at 7:00). For a phone call we are not interested who called whom first, just the fact that two people are talking on the phone.

The company stores the information of all the calls in a repository which is basically a CSV file. Each line of the file corresponds to a call. The first two fields are the ids of the two clients involved in the call, the third is the moment the call started and the last is the moment that the call ended. A moment is represented as an 8-digit number YYMMDDHHMM, where YY is the year, MM is the month, DD the day of the month, HH the hour and MM the minute. For example, the entry:

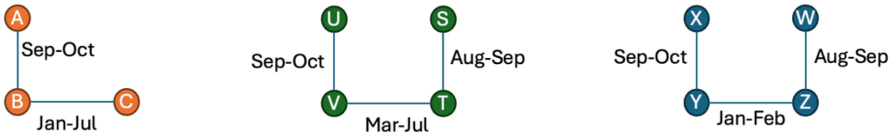
5, 7, 2409080630, 2409101140

means that there was a call between client 5 and 7 which started at 6:30 on 8th of Sept 2024 and finished at 11:40 on the 10th of Sept of the same year. Note that to preserve the privacy of the clients, they are simply presented as numbers. No other information is provided about the clients. It is perfectly natural to have more than one phonecalls between two clients as long as they are not temporarily overlapping.

We say that a client A is **related** to a client B if at some point in time there is a phone call between A and B, or if there is a set of persons p_1, p_2, p_3, …, p_{N-1}, p_N, such that there is a phone call from A to p_1, another phone call from P_N to B, and a phone call from p_i to p_{i+1} for each i from 1 to N-1. We call **community** the maximal set of clients for which any 2 persons selected are **related**. In other words, a community is a set of clients that are connected in a direct or indirect way though phone calls.

The information of the dataset can be easily represented as a graph where each node represents a client and an edge between two nodes represents a phone call between the respective clients. An edge is also annotated with the time that the phone call took place.

The company is interested in identifying (finding) communities that are spatiotemporally similar, meaning communities in which their clients are connected in similar ways in space an in time. In terms of space, this means that the connections (phone calls) have been performed between the clients in each community is similar. In terms of time, it means that the temporal variation between the phone calls of the clients is similar. For example, consider the 3 communities you see below, the red, the green and the blue. In terms of space, the similarity to the green and blue to the orange are the same (They both have 3 clients connected in the same way but have also one extra). However, in terms of time, the green is closer since the duration of the call between U and V is closer to the duration between B and C than the one between Y and Z.



Goal

In this project you are asked to develop a solution that identifies communities that are spatiotemporarily close enough so that the company treats them the same. (It is ok if the clients are different. We care only on the way the clients are related and the time of their phone calls. You need to design, implement, evaluate and write a report about it.

Your solution should assume that it is given a database (CSV file as described previously) and identify the communities that are similar enough. It is important to provide all the details of your solution and perform an extensive evaluation that should demonstrate how good (or bad) your solution is.

Programming language

This is left totally to you.

Number of persons

The project is for 3 persons max.

Input Dataset

There is no specific data that is given to you. But you can create one (this is called synthetic Dataset Generation) in a way that you test your program. You will need to devise a program that generates the necessary datasets that will allow you to stress test your work. You need to create a large dataset (do not create toy examples by hand. The toy examples will be only when you develop the code, but then when you need to evaluate the program you need serious examples.). You most likely do not need one dataset only but you need to create multiple datasets in order to be able to perform a complete evaluation of the developed solution.

Output - Visualization

The output of part1 consists of groups of communities that have been found to be similar. It is a text file of the form:

Group 1:

Community 1:

```
5, 7, 2409080630, 2405101140
7, 10, 209080630, 2402101140
7, 8, 2409080630, 1409101140
```

Community 2:

… etc.

Group 2:

Community 11:

….

Community 12:

….

etc.

Delivery

You need to deliver the code of the program you developed, the dataset you used, instructions on how the program runs and a report in which you describe the solution you have devised and the results of the experiments you have performed to prove the effectiveness and efficiency of your solution.

You should form your group and send an email to the instructor (i.velegrakis@uu.nl) with the names AND emails of the members of your group. The instructor will create a private channel on MS Teams which is where you will have to put any deliverables.

In the Files of the teams channel of your group, create the following directories (if not already there):

1. Src
2. Report
3. Presentation
4. Data

In the Presentation folder have a power point presentation called XX.pptx (where XX is the name of your group).

In the Src folder place the code of your project. Have also a README.txt file that explains how it runs.

In the Data folder place any datasets you used for the experiments and any output they generated

In the Report folder place a file XX.pdf where XX is the number of your group. The number of your group is in the name of your MS Teams channel.

PLEASE USE THE NAMING of files and folders EXACTLY AS REQUESTED. Do NOT put any extra information, characters of change something. The files are collected by a script program. If you use a different name, the script will copy nothing and the project will be marked incomplete.

Presentation

On the last week of the lectures (see the lecture schedule for the exact day) there will be a presentation in which a representative of every team will make a 5 min (maximum) presentation of the solution the group is developing. At least one slide is required for each of the following topics.

1. Project/group name and members (**with photos of the members** so that we know who is with whom) (0.3 points)
2. Solution (0.4 points)
3. Datasets + Experiments (to be done or have been done (0.3 points)

The presentation should be in line with the final project that will be delivered. Changes are of course allowed but one cannot have a dramatically new solution. The name of the power point file should be X.pptx where X is the number of your group. The number of your group is in the name of your channel.

Evaluation Criteria

The project is evaluated according to the following evaluation criteria:

1. Novelty & sophistication of the idea as well as how well it solves the problem.
2. Technical Depth: Detailed description of the approach and its challenging choices. How well data intensive technologies (Spark, Map-Reduce, NoSQL) have been exploited.
3. Presentation: Clarity and Completeness of the report & the presentation.
4. Experimental Evaluation
   - 4.1. The dataset(s) that have been used in the evaluation
   - 4.2. The evaluation tests that have been made (what has been tested and how)
   - 4.3. The comments on the results of the evaluation

Structure of the report

The final report should be written in Latex, using the following template which is available on overleaf: https://www.overleaf.com/read/bgrgzbqhqkjr#412001 It should contain the following sections:

1. **Introduction** (maximum 1 page) in which you introduce the problem you are solving, its importance and the main highlights of your solution (1 paragraph) and the results of your experiments (1 paragraph). Provide a

motivation for this work. (Why you think that such a study is important? (you already have an application so it is clear that it is important, but maybe you can think additional applications to make the statement stronger). And why it is challenging (i.e., not trivial) to perform this processing? What were the hard/challenging parts in developing a solution?) Note that a “hard/challenging” part should be generic and not personal to the authors. They should apply to everyone and are challenging due to the nature of the problem at hand. They should not be challenging just because of the capabilities of the author. For example, if the solution is developed in python and the programmer does not know python, then clearly the difficulty is only for the specific author and not for everyone.

2. **Related work** and technologies (maximum 1 page): Any information you think is important for the reader to know but IS NOT your own work. For example, you could describe there what Spark is, what map reduce is, etc. Do not waste space getting into details that everyone else knows already from the lectures or other online sources. Keep it to the basics and to the minimum.
3. **Solution**. In this section you describe in detail what your solution is (or what your solutions are, in case of more than one). The more detailed you are in this section the better the section is. Imagine that you give your report to someone else, and you ask her/him to implement your solution. Will that person be able to do it by looking only at what is written in the document? If yes, then the document is successful. Also explain the reasoning behind every choice you are making. You are free to include some pseudocode because it makes it much easier for people to understand what the text is saying.
4. **Experimental evaluation**. This section contains a detailed description of all the experiments you have done to understand how well your solution works. How does it compare to some baseline? The more things you are testing, the more it helps to understand the performance of the solution, and the better the report is. Since the company refuses to share its datasets, the experiments will be on synthetic data. The size of the section is up to you, since it depends on the complexity of the solution you are proposing and the details you would like to study. Make sure that you also provide a description of the datasets you used as input.
5. **Conclusion**. A recap of what you did in your work (the main highlights). Maximum half a page.

On blackboard there are two sample reports from previous years.