# Project 2

## 1 Data Description

This project involves a dataset comprising human mobility data from four metropolitan areas of Japan. All data have been anonymized before use to protect individual privacy.

Data link:

https://entuedu-my.sharepoint.com/:f:/r/personal/shuai004_e_ntu_edu_sg/Documents/13237029?csf=1&web=1&e=RHtfVL

### Grid Division

- **Spatial Resolution:** Each area is divided into 500-meter x 500-meter grid cells.
- **Grid Layout:** Each metropolitan area consists of a 200 x 200 grid.

### Time and Spatial Data

- **Time Span:** The data captures human activities over a period of 75 days, with a temporal resolution of 30 minutes.
- **Spatial Coordinates:** The location data is normalized to 500-meter grid cells.
- **Data Integrity:** Datasets B, C, and D have missing data for the last 15 days (from day 61 to 75), during which the position coordinates (x, y) are marked as -999.

### Data File Structure

- **Mobility Data Files:** Consist of four compressed csv files, each recording the mobility data of residents in one city. The data columns include:
  - uid: User ID
  - d: Current date
  - t: Specific time of the record
  - x: x-axis coordinate (grid column number, counted from left to right, starting from 1)
  - y: y-axis coordinate (grid row number, counted from top to bottom, starting from 1)

### Points of Interest (POI) Data

- **POI Category Table:** A csv file named POI_datacategories.csv, which includes 85 types of POI categories and their corresponding IDs.
- **POI Distribution Data:** Four csv files that record the distribution of POIs in each grid of every city, with columns named:
    - `x`, `y`: The x and y-axis coordinates of the grid
    - `category`: POI category
    - `POI_count`: The number of that POI category present in the grid

# 2 Task Description

## 2.1 Analysis of Co-occurrence Patterns of Points of Interest (POI)

- Objective:

The goal of this task is to analyze the co-occurrence patterns of different Points of Interest (POI) within each city. Specifically, the task aims to identify types of POIs that frequently appear together within the same grid cell.

- Method:

Implement the Apriori algorithm to analyze the POI data for each city, identifying common combinations of POIs that frequently co-occur within the same grid.

- Tips:

1. **Data Preparation for Apriori Algorithm:** Organize the data so that each grid cell represents a "basket." The "items" in this basket are the POI categories that appear in that grid.
2. **Analysis of Frequent Itemsets:** Analyze the frequent itemsets to determine which POI categories tend to co-occur within the same grid.
3. **Data Handling Choices:** You may choose to ignore the specific quantities of each POI, focusing only on the presence of POI categories. Alternatively, you can consider the quantity of each POI by representing each grid's POIs in the form "Category: Quantity."

**Code Reuse:** You can refer to online code, and reuse part of online code. But you must understand the code. **If you reuse some online code, you need to tell what online code you use in your report. In the report, you need to find a way to show the result of your code is correct**

**In the report, you need to document what you have done, and report the results.**

## 2.2 Mining Sequential Patterns

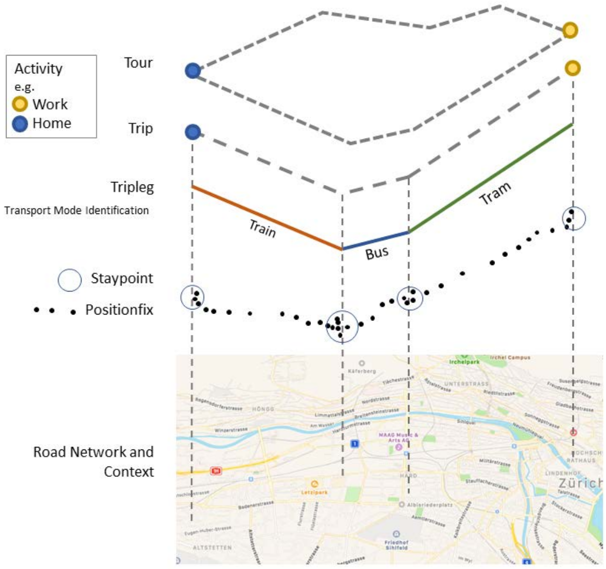
- Objective:

The objective of this task is to analyze the movement sequences of residents in each city to uncover common patterns of mobility through sequential pattern mining.

- Steps

1, Data Preprocessing

Utilize the Python library `trackintel` to generate "triplegs," which represent movement segments from one location to another. For detailed preprocessing steps, refer to the GitHub project: mie-lab/location-prediction: [TRC] Context-aware next location prediction.



2, Data Analysis:

Consider each tripleg as a sequence where each element is a two-dimensional coordinate (x,y). Implement the Generalized Sequential Pattern (GSP) algorithm to mine sequential patterns from the triplegs.

- Example

Consider the following tripleg sequence:

- <{2,4}, {5,6}, {1,2}>

Using the GSP algorithm, we can mine frequent subsequences such as:

- <{2,4}, {1,2}>

This indicates that moving from coordinate {2,4} to {1,2} is a common mobility pattern.

- Tips

  - **Handling Large Datasets:**
    - Pay attention to performance and computation time, especially when tuning parameters to optimize the execution of the GSP algorithm.
    - Consider using a simplified dataset to test the accuracy and efficiency of the algorithm to ensure it is well-adjusted before processing the full dataset.
  - **Scope of Data Analysis:**
    - Due to the scale of the data, limit the frequency mining to only the first month's data (30 days) to manage computational demands effectively.
  - **Handling too long triplegs**
    - For convenience, if a tripleg is too long, you can decide whether to split it into shorter sub-triplegs.

**In the report, you need to document what you have done, and report the results.**

## Task 3: Open Advanced Tasks

- Objective

Define an application (e.g., predicting the next location) based on the datasets, and give a solution. The solution Includes but not limited to traditional analytics, machine learning, deep learning, and LLM-related tasks. Implement your solution, and provide some experimental results to show your solution works.

**In the report, you need to explain the algorithm you design. You also need to document the results of your algorithm.**

Weight of grading:

- Task 1: 30%
- Task 2: 40%
- Task 3: 30%

**You are expected to improve your problem solving, deep thinking, and self-learning ability through the project, which are very important skills to acquire in universities.**

**What to deliver:**

**Code:** for the three tasks.

**Report:** The final report is up to 8 A4 pages (not necessary to write 8 pages. The page limit does not include front page).

**In the end of report, please include the individual contribution claims (which should be agreed by all of you) in the following format:**

Member name 1: list of contributions to the project.

**Up to 5 minutes video.** In the short video, you can capture screen to do a demo of your code to show it works, and you can also highlight any other part of your report in the video.
Project are done in groups. Discussions with other students are allowed, but each group has to write your own code.

## Code submission + report +video: Due in the end of Thursday, 21 Nov. Only softcopy is required, and submission will be through NTUlearn.

Grading will consider report + code + video

**NOTE:**

1. **MOSS**: Sharing code with your classmates is not acceptable!!! All programs will be screened using the Moss (Measure of Software Similarity.) system.
2. **You are not allowed to share your project code on the web publicly**.

TA for projects:

- Liu Shuai (shuai004@e.ntu.edu.sg)

**If you have questions, please email the email above and cc to me (gaocong@ntu.edu.sg). TAs can only provide some consultation for projects, but you should NOT expect TAs to help to do any part of your project.**