TRƯỜNG ĐẠI HỌC THỦY LỢI



BÁO CÁO thực hành

**Nhận diện thói quen đi lại từ dữ liệu gps bằng phân tích tần số**

Sinh viên:

Nguyễn Thiệu Huy

MSV: 2251262610

Lớp: 64TTNT1

MÔN HỌC: PHÂN TÍCH CHUỖI THỜI GIAN

Đề số 3: Cho dữ liệu tốc độ di chuyển theo thời gian của một người trong vòng 30 ngày (ghi mỗi 15 phút). Sử dụng phân tích phổ để phát hiện chu kỳ hoạt động.

Hà Nội, 05/2025

MỤC LỤC

[CHƯƠNG 1 MÔ TẢ BÀI TOÁN VÀ BỘ DỮ LIỆU 1](#_Toc199333540)

[1.1 Mô tả bài toán 1](#_Toc199333541)

[1.2 Mô tả bộ dữ liệu 1](#_Toc199333542)

[1.2.1 Các tệp và thuộc tính trong bộ dữ liệu 1](#_Toc199333543)

[1.2.2 Tiền xử lý dữ liệu 2](#_Toc199333544)

[CHƯƠNG 2 PHÂN TÍCH PHÁT HIỆN CHU KỲ HOẠT ĐỘNG 5](#_Toc199333545)

[2.1 Xử lý tìm các chu kỳ lặp lại 5](#_Toc199333546)

[2.2 Phân tích phổ để tìm chu kỳ hoạt động 5](#_Toc199333547)

[2.3 Nhận diện chu kỳ chính 7](#_Toc199333548)

[2.4 Trực quan hóa tần suất di chuyển theo ngày và giờ 7](#_Toc199333549)

[CHƯƠNG 3 KẾT LUẬN 9](#_Toc199333550)

[3.1 Kết luận 9](#_Toc199333551)

[3.2 Phụ lục 9](#_Toc199333552)

[TÀI LIỆU THAM KHẢO 9](#_Toc199333553)

# MÔ TẢ BÀI TOÁN VÀ BỘ DỮ LIỆU

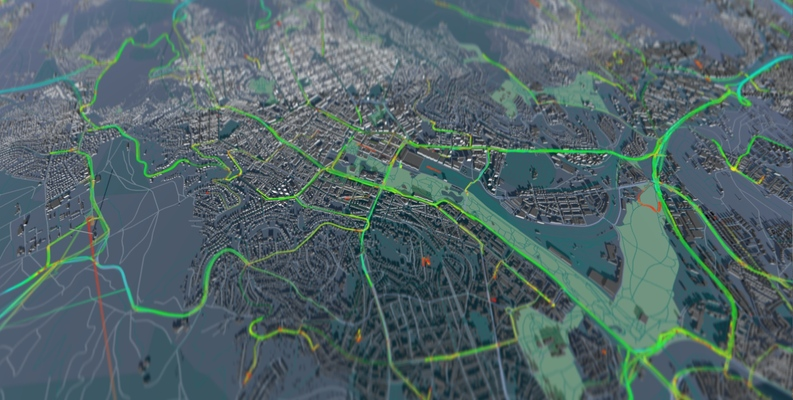
## Mô tả bài toán

Bài toán của buổi thực hành hôm nay là phân tích thói quen di chuyển cá nhân dựa trên dữ liệu GPS bằng phương pháp phân tích phổ (spectral analysis) nhằm mục đích phát hiện các chu kỳ hoạt động lặp lại (daily, weekly) để dự đoán hành vi của người dùng đó.

## Mô tả bộ dữ liệu

Bộ dữ liệu GPS Trajectory từ Kaggle được thu thập trong dự án Geolife của Microsoft từ 178 người dùng trong khoảng 4 năm ( từ 4/2007 đến 10/2011).

Dữ liệu được biểu diễn dưới dạng một chuỗi các điểm được đánh dấu thời gian, mỗi điểm chứa thông tin về **kinh độ, vĩ độ và độ cao**. Bộ dữ liệu này bao gồm **17.621 quỹ đạo**, với tổng quãng đường là **1.251.654 km** và tổng thời gian ghi nhận là **48.203 giờ**. Dữ liệu được ghi lại bằng nhiều thiết bị ghi GPS và điện thoại có GPS khác nhau, với tần suất lấy mẫu khác nhau.



*Microsoft Geolife GPS Trajectory Dataset*

### Các tệp và thuộc tính trong bộ dữ liệu

**Tệp quỹ đạo (.plt)**

Mỗi tệp .plt đại diện cho một quỹ đạo duy nhất và chứa các điểm dữ liệu được ghi lại theo thời gian. Các dòng từ 1 đến 6 trong mỗi tệp .plt chỉ là thông tin bổ sung và có thể bỏ qua. Dữ liệu thực tế bắt đầu từ dòng thứ 7, với mỗi dòng tương ứng một điểm GPS và có các thuộc tính sau:

* Latitude (Vĩ độ): Mô tả tạo độ vĩ độ của điểm GPS
* Longitude (Kinh độ): Mô tả kinh độ của điểm GPS
* Altitude (Độ cao): Mô tả độ cao so với mực nước biển
* Date (Định dạng số): VD: 30/12/2007
* Date (Định dạng chuỗi): có dạng YYYY-MM-DD
* Time (Thời gian chi tiết – Định dạng chuỗi): có dạng HH:MM:SS

**Tệp nhãn (labels.txt)**

Tệp labels.txt (nếu có trong thư mục của người dùng) cung cấp thông tin về phương tiện di chuyển được sử dụng trong các khoảng thời gian nhất định của các quỹ đạo. Mỗi dòng trong tệp này có các thuộc tính sau, thường được phân tách bằng tab (\t):

* Start Time (Thời gian bắt đầu): Mô tả điểm bắt đầu bằng 1 phương tiện cụ thể
* End Time (Thời gian kết thúc): Mô tả thời điểm kết thúc
* Transportation Mode (Phương tiện di chuyển): Mô tả loại phương tiện được sử dụng trong khoảng thời gian trên.

### Tiền xử lý dữ liệu

Các bước chính:

Đọc file labels.txt để lưu các khoảng thời gian di chuyển và phương tiện người đó sử dụng. Sau đó đọc từng file .plt, ánh xạ thời gian từng điểm GPS vào khoảng thời gian tương ứng trong labels. Cuối cùng gán nhãn phương tiện cho từng điểm GPS theo thời gian.

Chi tiết:

# Đọc labels.txt

def read\_labels(label\_file):

    df = pd.read\_csv(label\_file, sep='\t')

    df['start\_time'] = pd.to\_datetime(df['Start Time'])

    df['end\_time'] = pd.to\_datetime(df['End Time'])

    df.rename(columns={'Transportation Mode': 'transport\_mode'}, inplace=True)

    return df[['start\_time', 'end\_time', 'transport\_mode']]

Chuyển đổi cột 'Start Time' trong DataFrame df sang định dạng datetime. Sau đó chuyển đổi cột 'End Time' sang định dạng datetime và lưu vào cột end\_time. Cuối cùng hàm trả về một DataFrame mới chỉ chứa ba cột: start\_time, end\_time, và transport\_mode

# Đọc file .plt

def read\_plt\_file(file\_path):

    df = pd.read\_csv(file\_path, skiprows=6, header=None)

    df.columns = ['lat', 'lon', 'unused', 'alt', 'days', 'date', 'time']

    df['datetime'] = pd.to\_datetime(df['date'] + ' ' + df['time'])

    return df[['lat', 'lon', 'alt', 'datetime']]

Gán tên cho các cột của DataFrame df theo thứ tự chúng xuất hiện trong tệp .plt (sau khi đã bỏ qua 6 dòng đầu).

* 'lat': Vĩ độ (Latitude)
* 'lon': Kinh độ (Longitude)
* 'unused': Trường không sử dụng (luôn là 0)
* 'alt': Độ cao (Altitude) tính bằng feet
* 'days': Số ngày (kể từ 30/12/1899)
* 'date': Ngày ở định dạng chuỗi 'YYYY-MM-DD'
* 'time': Thời gian ở định dạng chuỗi 'HH:MM:SS'

# Gán nhãn

def assign\_labels(gps\_df, labels\_df):

    gps\_df['transport\_mode'] = 'unknown'

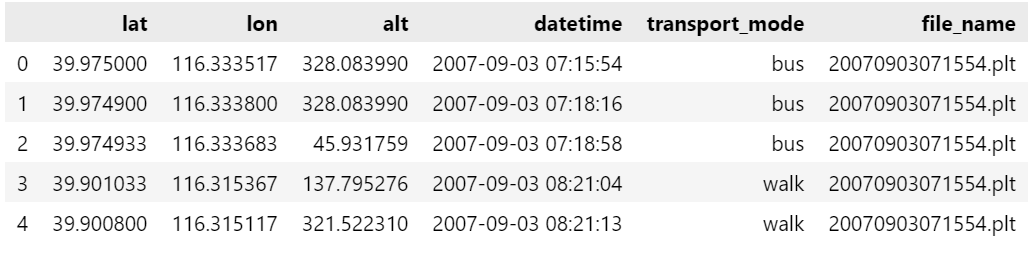
    for \_, row in labels\_df.iterrows():

        mask = (gps\_df['datetime'] >= row['start\_time']) & (gps\_df['datetime'] <= row['end\_time'])

        gps\_df.loc[mask, 'transport\_mode'] = row['transport\_mode']

    return gps\_df

Thêm một cột 'transport\_mode' vào DataFrame GPS và khởi tạo tất cả giá trị là ‘unknown’. Sau đó lặp qua mỗi nhãn, xác định tất cả các điểm GPS có thời gian nằm trong khoảng thời gian của nhãn đó. Cuối cùng gán phương tiện di chuyển của nhãn đó cho các điểm GPS đã xác định.



*DataFrame cuối cùng thu được sau khi tiền xử lý*

# PHÂN TÍCH PHÁT HIỆN CHU KỲ HOẠT ĐỘNG

## Xử lý tìm các chu kỳ lặp lại

Bước này sẽ tính tốc độ di chuyển giữa các điểm và sau đó nội suy để có tốc độ mỗi 15 phút. Chi tiết quá trình xử lý: Tạo một chuỗi thời gian mới bằng cách lấy mẫu lại dữ liệu vận tốc: Tính vận tốc trung bình cho mỗi khoảng 15 phút, sau đó nội suy để điền vào các khoảng 15 phút có thể không có dữ liệu vận tốc.

Bước này làm mịn dữ liệu GPS và chuyển đổi nó sang một tần suất thời gian đều đặn hơn, hữu ích cho việc phân tích xu hướng.

def compute\_speed(row1, row2):

    dist = geodesic((row1['lat'], row1['lon']), (row2['lat'], row2['lon'])).meters

    time\_diff = (row2['datetime'] - row1['datetime']).total\_seconds()

    if time\_diff == 0:

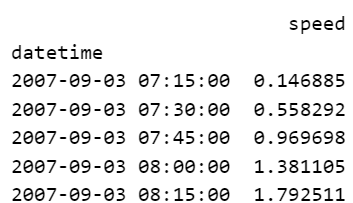
        return 0

    return dist / time\_diff  # m/s

df['speed'] = 0.0

for i in range(1, len(df)):

    df.loc[df.index[i], 'speed'] = compute\_speed(df.iloc[i-1], df.iloc[i])



*Thu được vận tốc mỗi 15 phút*

## Phân tích phổ để tìm chu kỳ hoạt động

Thực hiện phép biến đổi Fourier nhanh (Fast Fourier Transform) trên dữ liệu vận tốc ở phần trước để phân tích thành các thành phần tần suất ( chu kỳ lặp lại) trong tín hiệu vận tốc.

from scipy.fft import fft, fftfreq

import matplotlib.pyplot as plt

speed\_signal = df\_ts['speed'].values

N = len(speed\_signal)

T = 15 \* 60  # 15 phút = 900s

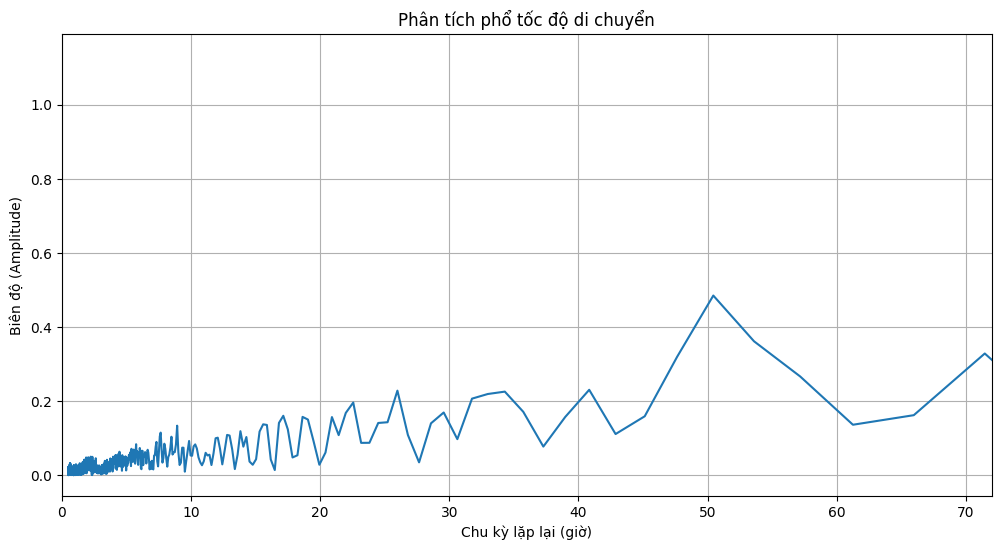
yf = fft(speed\_signal)

xf = fftfreq(N, T)

yf = 2.0/N \* np.abs(yf[1:N//2])

xf = xf[1:N//2]

Trực quan hóa biểu đồ tần suất:



Vài điểm nhận xét:

**Đỉnh rõ rệt nhất (Chu kỳ ~50 giờ):** Rất rõ ràng với biên độ cao nhất (khoảng 0.48 - 0.5) tại chu kỳ lặp lại khoảng 50 giờ. Điều này cho thấy một thói quen hoặc một sự kiện liên quan đến tốc độ di chuyển có tính chu kỳ mạnh mẽ, lặp lại sau mỗi khoảng hơn 2 ngày.

**Chu kỳ hàng ngày (~24 giờ):** Có một đỉnh đáng chú ý với biên độ khoảng 0.2 tại chu kỳ lặp lại xấp xỉ 23-25 giờ. Điều này có thể khớp với các hoạt động lặp lại hàng ngày (ví dụ như: đi làm, về nhà, các thói quen sinh hoạt khác có ảnh hưởng đến tốc độ di chuyển).

## Nhận diện chu kỳ chính

Bước này sử dụng hàm find\_peaks từ thư viện scipy.signal để tự động phát hiện các đỉnh nổi bật trong phổ biên độ yf (đã được tính toán từ bước phân tích FFT trước đó).

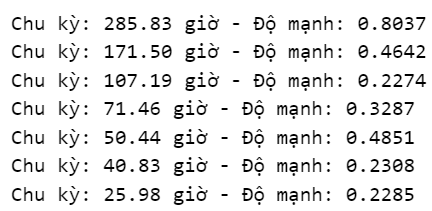
from scipy.signal import find\_peaks

peaks, properties = find\_peaks(yf, height=np.max(yf)\*0.2)  # Ngưỡng

for i in peaks:

    print(f"Chu kỳ: {1/xf[i]/3600:.2f} giờ - Độ mạnh: {yf[i]:.4f}")

Chỉ những đỉnh nào có biên độ (chiều cao) lớn hơn hoặc bằng 20% của đỉnh cao nhất mới được coi là một đỉnh hợp lệ.



*Kết quả*

Chu kỳ: 285.83 giờ - Độ mạnh: 0.8037 Đây là chu kỳ có độ mạnh **cao nhất** trong số các đỉnh được phát hiện, cho thấy đây là thành phần có tính chu kỳ trội nhất với ngưỡng đã đặt (0.2) Chu kỳ này tương đương khoảng **11.9 ngày**. Đây có thể là thói quen như đi làm, hoặc chạy bộ, …

## Trực quan hóa tần suất di chuyển theo ngày và giờ

Tạo biểu đồ với:

Trục x: giờ trong ngày (0 🡪 23)

Trục y: ngày trong tháng

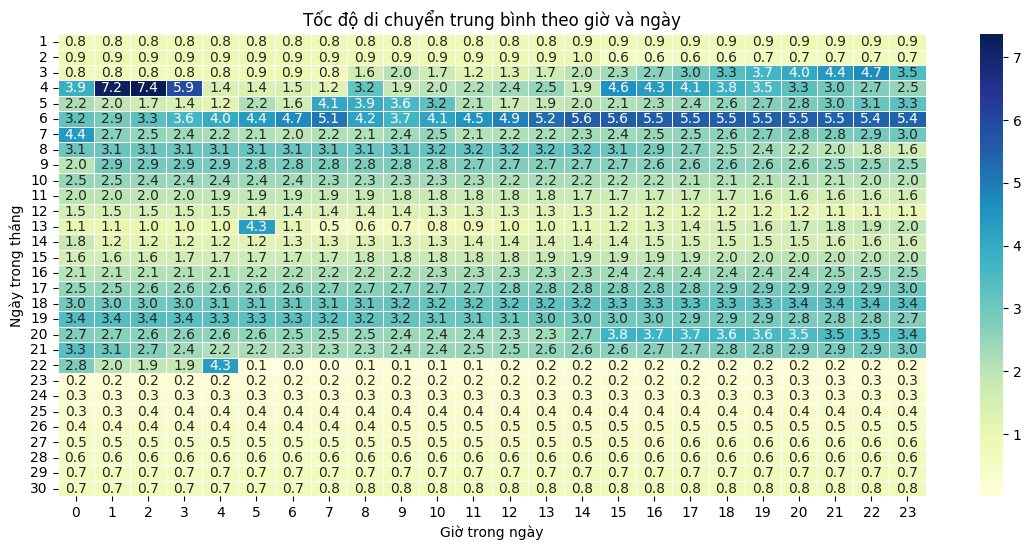
df\_ts = df\_ts.copy()

df\_ts['hour'] = df\_ts.index.hour

df\_ts['day'] = df\_ts.index.day

# Tính tốc độ trung bình theo ngày và giờ

heatmap\_data = df\_ts.pivot\_table(index='day', columns='hour', values='speed', aggfunc='mean')



*Heatmap tốc độ di chuyển trung bình*

Dựa vào biểu đồ heatmap có thể nhận diện thói quen cá nhân như sau:

Giờ cao điểm có tốc độ cao 🡪 Khả năng dùng bus/train đi học hoặc đi làm

Giờ thấp điểm gần bằng 0 🡪 Có thể đang nghỉ, ngủ hoặc làm việc ở nơi cố định

Nếu tốc độ cao về đêm 🡪 Có thể là làm đêm (chạy xe) hoặc đi chơi.

# KẾT LUẬN

## Kết luận

Từ việc trực quan hóa biểu đồ phổ và danh sách các đỉnh được phát hiện tự động, em thấy rằng:

Hành vi di chuyển của người dùng được phân tích có tính chu kỳ và các chu kỳ nổi bật nhất là khoảng 285.83 giờ ( gần 12 ngày) với độ mạnh rất cao là 0.8037. Có thể nhận xét được là trong khoảng thời gian này, người đó đã có một thói quen có thể là đi làm bằng phương tiện như xe máy, ô tô, bus.

## Phụ lục

Github repo: [ThieuHuy43/amazon-revenue-forecasting](https://github.com/ThieuHuy43/amazon-revenue-forecasting) (chưa commit được code mới do lỗi dữ liệu)

Dataset link: [Amazon Reviews'23](https://amazon-reviews-2023.github.io/) (Mục “Clothing\_Shoes\_and\_Jewelry”)

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting: [[2106.13008] Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting](https://arxiv.org/abs/2106.13008) |
| [2] | Long Short-term Memory RNN: [[2105.06756] Long Short-term Memory RNN](https://arxiv.org/abs/2105.06756) |
|  |  |