CS313
Clustering Taxi Trajectories

Group 9

University of Information Technology
VNU-HCM

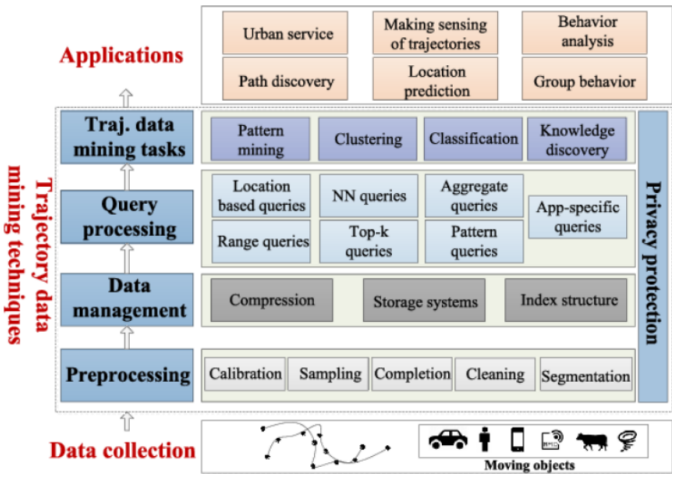
Gia Phúc, Minh Nhựt, Hùng Phát, Thu Phương, Đình Quân, Anh Quân

Supervisor: Võ Nguyễn Lê Duy

Ngày 28 tháng 6 năm 2025

Trajectory Data

- Dữ liệu quỹ đạo (trajectory data) là dữ liệu theo dõi đường di chuyển của các đối tượng theo thời gian, được ghi lại thông qua các thiết bị định vị.
- Sự phát triển nhanh chóng của các thiết bị di động đã dẫn đến việc thu thập một lượng lớn các quỹ đạo GPS (GPS trajectories) từ các dịch vụ định vị, mạng xã hội dựa trên vị trí, ứng dụng giao thông hoặc các ứng dụng chia sẻ phương tiện.

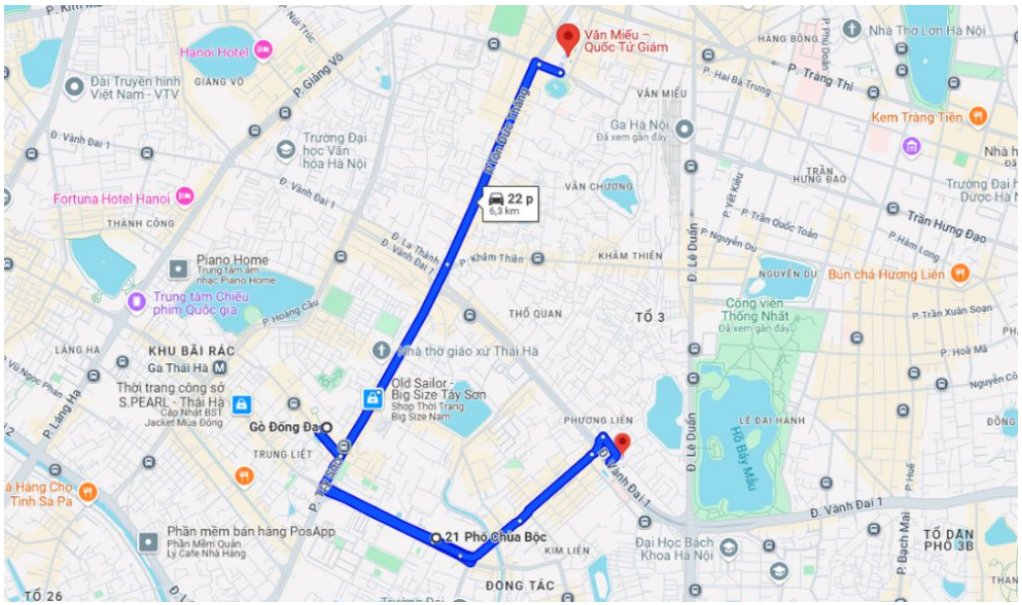


Hình 1: General Framework of Trajectory Data Mining

Trajectory Data

Quỹ đạo GPS (GPS trajectories) mô tả hành trình di chuyển, có thể được sử dụng để:

- Xác định tuyến đường phổ biến
- Theo dõi phương tiện giao thông (xe buýt, xe công nghệ,...)
- So sánh mức độ tương đồng giữa các hành trình

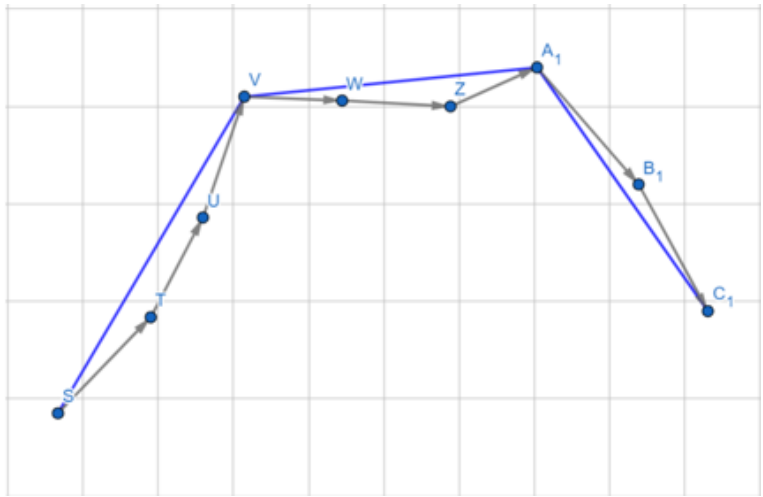


Hình 2: Ví dụ minh họa

Problem

Vấn đề đặt ra

- Dữ liệu có quá nhiều điểm, gây khó khăn khi phân cụm
- Cần phương pháp giảm số điểm nhưng vẫn giữ được hình dạng quỹ đạo ($\rightarrow$ RDP)
- Cần chọn thuật toán phân cụm và phép đo khoảng cách phù hợp
- Cần trực quan hóa kết quả rõ ràng



Hình 3: Vấn đề trong phân cụm quỹ đạo GPS.

Distance Metric

DTW – Dynamic Time Warping

- **Tổng quan:** DTW (Dynamic Time Warping) là phương pháp đo độ tương đồng giữa hai quỹ đạo di chuyển, bằng cách căn chỉnh tối ưu các điểm tọa độ theo trục thời gian. Phương pháp này cho phép các quỹ đạo bị giãn, co lại hoặc lệch pha, giúp so sánh các quỹ đạo dù có độ dài khác nhau hoặc di chuyển với tốc độ khác nhau.
- **Công thức:**

$$DTW(i,j) = \|a_i - b_j\| + \min \begin{cases} DTW(i-1,j), \\ DTW(i,j-1), \\ DTW(i-1,j-1) \end{cases}$$

Hình 4: Công thức của DTW

Distance Metric

DTW – Dynamic Time Warping

- **Trong đó:**
  - **DTW(i, j):**
    Đại diện cho khoảng cách giữa hai chuỗi dữ liệu A và B tại các điểm tương ứng a_i và b_j.
  - **Khoảng cách giữa hai điểm:**

$$||a_i - b_j|| = \sqrt{(a_i - b_j)^2}$$

  - **Phần min:**
    Để tính $DTW(i, j)$, xem xét ba giá trị DTW lân cận:
    - $DTW(i-1, j)$: Khoảng cách từ chuỗi A phía trên.
    - $DTW(i, j-1)$: Khoảng cách từ chuỗi B bên trái.
    - $DTW(i-1, j-1)$: Khoảng cách từ điểm chéo phía trên bên trái.

    Chọn giá trị nhỏ nhất giữa ba giá trị này để tìm con đường tối ưu cho khoảng cách giữa hai chuỗi.

Distance Metric

Fréchet Distance

- **Tổng quan:** Là khoảng cách giữa hai quỹ đạo di chuyển (trajectories), xét đến hình dạng tổng thể và trình tự điểm trên đường đi.
- **Công thức:**

$$\delta_F(A, B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \|A(\alpha(t)) - B(\beta(t))\|$$

Hình 5: Công thức của Fréchet Distance

Distance Metric

Fréchet Distance

- **Trong đó:**
  - $\delta_F(A, B)$: Khoảng cách giữa hai hàm A và B.
  - $\inf$: Tìm giá trị nhỏ nhất.
  - $\max$: Tìm giá trị lớn nhất từ các khoảng cách giữa các điểm tương ứng của hai hàm.
  - $A(\alpha(t))$ **và** $B(\beta(t))$: Các điểm trên hai hàm A và B được xác định bởi các tham số $\alpha(t)$ và $\beta(t)$.

Distance Metric

Thuộc tính	**DTW**	**Fréchet**
Căn chỉnh theo thời gian	Có	Có
Xét hình dạng tổng thể	Không rõ ràng	Có
Độ nhạy với thời gian	Trung bình	Thấp
Độ phù hợp với dữ liệu GPS	Tốt nếu dữ liệu mượt, đều	Tốt nếu cần hình dạng chính xác

Bảng 1: So sánh giữa DTW và Fréchet

DBSCAN

- **Tổng quan:** DBSCAN là một thuật toán phân cụm dựa trên mật độ, được sử dụng rộng rãi trong các bài toán phân tích dữ liệu không giám sát và đặc biệt phù hợp cho bài toán phân cụm quỹ đạo.
- **Các bước hoạt động:**
  - **Bước 1:** Với mỗi điểm, xét xem có đủ số lượng điểm lân cận (MinPts) trong bán kính epsilon ( ) không.
  - **Bước 2:** Nếu có, điểm này là core point và tạo thành cụm với các điểm lân cận.
  - **Bước 3:** Các điểm được mở rộng nếu nằm trong vùng mật độ của cụm.
  - **Bước 4:** Những điểm không thuộc cụm nào là nhiễu (noise).

DBSCAN

- **Ưu điểm:**
  - Không cần chỉ định số cụm trước.
  - Phát hiện cụm có hình dạng không chuẩn.
  - Xử lý tốt dữ liệu có nhiễu.
- **Nhược điểm:**
  - Khó chọn tham số epsilon và MinPts.
  - Không tốt khi mật độ cụm khác nhau quá nhiều.

K-Medoids

- **Tổng quan:** K-Medoids là một thuật toán phân cụm thuộc nhóm unsupervised learning (học không giám sát), tương tự như K-Means, nhưng có một điểm khác biệt quan trọng: Thay vì sử dụng trung bình cộng (centroid) để đại diện cho mỗi cụm như K-Means, K-Medoids chọn một điểm thực trong dữ liệu (gọi là medoid) làm trung tâm cụm.
- **Các bước hoạt động:**
  - **Bước 1:** Chọn k điểm bất kỳ trong tập dữ liệu làm medoid ban đầu.
  - **Bước 2:** Gán mỗi điểm còn lại vào medoid gần nhất (theo khoảng cách).
  - **Bước 3:** Với mỗi cụm, thử hoán đổi medoid với một điểm trong cụm để giảm tổng khoảng cách nội cụm.
  - **Bước 4:** Lặp lại bước 2-3 đến khi không còn cải thiện.

K-Medoids

- **Ưu điểm:**
  - Chống nhiễu tốt hơn K-Means.
  - Kết quả ổn định hơn.
- **Nhược điểm:**
  - Tốn thời gian hơn khi số điểm lớn.
  - Phải xác định số cụm K trước.
  - Không thể phát hiện nhiễu

Agglomerative Clustering

- **Tổng quan:** Agglomerative Clustering là một phương pháp phân cụm theo hướng hệ phân cấp (Hierarchical Clustering). Đây là thuật toán kết hợp từ dưới lên (bottom-up).
- **Các bước hoạt động:**
  - **Bước 1:** Mỗi điểm bắt đầu là 1 cụm riêng lẻ.
  - **Bước 2:** Tính khoảng cách giữa các cụm.
  - **Bước 3:** Gộp 2 cụm gần nhất lại.
  - **Bước 4:** Cập nhật lại ma trận khoảng cách.
  - **Bước 5:** Lặp lại cho đến khi còn số cụm mong muốn.

Agglomerative Clustering

- **Ưu điểm:**
  - Không cần chỉ định số cụm trước.
  - Có thể mô hình hóa các mối quan hệ phân cấp.
- **Nhược điểm:**
  - Tốn nhiều thời gian với dữ liệu lớn.
  - Nhạy cảm với nhiễu và điểm ngoại lệ.

Silhouette Score

- **Tổng quan: Silhouette Score** là một chỉ số đánh giá chất lượng của việc phân cụm, giúp xác định **mức độ hợp lý của các cụm được tạo ra**.
- **Ý nghĩa:**
  - Giá trị Silhouette nằm trong khoảng **[-1, 1]**:
    - Gần **1**: Điểm dữ liệu được phân cụm rất tốt (gần tâm cụm của nó, xa các cụm khác).
    - Gần **0**: Điểm dữ liệu nằm giữa hai cụm, không rõ ràng thuộc cụm nào.
    - Gần **-1**: Điểm dữ liệu có thể bị gán sai cụm.
- **Công thức:**

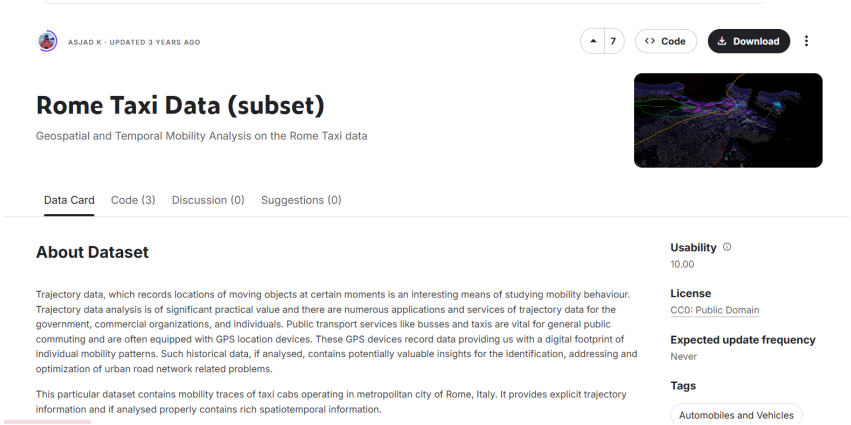
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Trong đó:

- $a(i)$: khoảng cách trung bình từ điểm i đến các điểm khác trong **cùng một cụm**.
- $b(i)$: khoảng cách trung bình từ điểm i đến các điểm trong **cụm gần nhất khác**.

Dataset Overview

Bộ dữ liệu Rome Taxi Data (subset), được tải từ Kaggle 1. Nó chứa các dấu vết GPS của các xe taxi hoạt động tại Rome, Ý — bao gồm tọa độ của khoảng 320 xe taxi được thu thập trong vòng 30 ngày (từ ngày 1 tháng 2 năm 2014 đến ngày 2 tháng 3 năm 2014). Dữ liệu này đại diện cho các tài xế chủ yếu làm việc tại khu vực trung tâm Rome. Tệp hiện tại là một tập con của bộ dữ liệu gốc, được sử dụng cho mục đích phân tích ban đầu.



Hình 6: Illustration of taxi data in Beijing

Dataset Overview

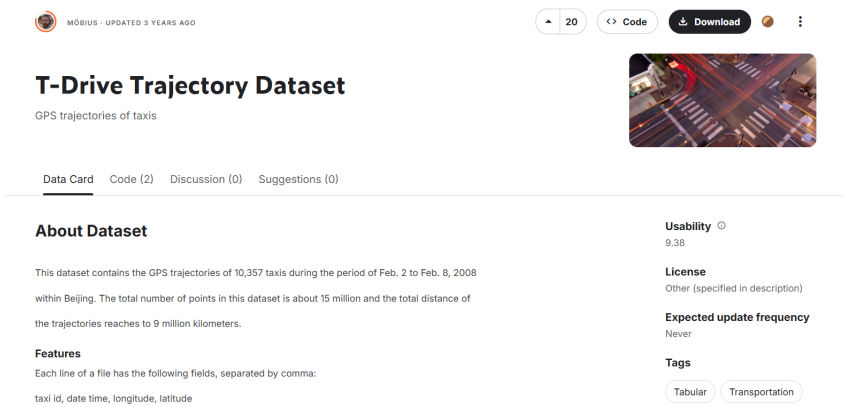
Rome Taxi Data (subset)

Date and Time	Unnamed: 0	DriveNo	Latitude	Longitude
2014-02-01 00:00:00.739166+01:00	1	156	41.883672	12.487778
2014-02-01 00:00:01.148457+01:00	2	187	41.928543	12.469037
2014-02-01 00:00:01.220066+01:00	3	297	41.891069	12.492705
2014-02-01 00:00:01.470854+01:00	4	89	41.793177	12.432122
2014-02-01 00:00:01.631136+01:00	5	79	41.900275	12.462746

Hình 7: Cấu trúc Dataset

Dataset Overview

Bộ dữ liệu **T-Drive Trajectory Dataset**, được tải từ Kaggle [1]. Bộ dữ liệu này chứa các quỹ đạo GPS của 10.357 xe taxi trong khoảng thời gian từ ngày 2 đến ngày 8 tháng 2 năm 2008 tại Bắc Kinh. Tổng số điểm dữ liệu trong bộ này vào khoảng 15 triệu điểm và tổng chiều dài các quỹ đạo lên tới 9 triệu km.



Hình 8: Illustration of taxi data in Beijing

[1] https://www.kaggle.com/datasets/arashnic/tdriver?fbclid=IwY2xjawJs5BlleHRuA2FlbQIxMAABHg4Rr_ZX-qEDDnkxyhj2oGN4oNsGvikYfkeDiP8olcjuHeqXARf0aKz9tbzJ_aem

Dataset Overview

T-Drive Trajectory Dataset

TimeStamp	TaxiID	Longitude	Latitude
2008-02-03 00:00:32	1	116.69171	39.85184
2008-02-03 00:10:32	1	116.69170	39.85184
2008-02-03 00:20:32	1	116.69170	39.85184
2008-02-03 00:30:32	1	116.69168	39.85146
2008-02-03 00:40:32	1	116.69172	39.85165

Hình 9: Cấu trúc Dataset

Demo

Datasets	Distance \ Algorithm	K-medoids	AgglomerativeClustering	DBSCAN
Beijing Taxi	DTW	0.4913	0.4813	x
	Frechet	0.1739	0.6519	x
Rome Taxi	DTW	0.1443	0.835	x
	Frechet	0.1112	0.7599	x

Hình 10: Bảng so sánh hiệu suất

Thanks for your attention!