ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRỊNH BÁ QUÝ

PHÂN TÍCH VÀ MÔ PHỎNG TÌNH TRẠNG GIAO THÔNG DỰA VÀO KHAI PHÁ DỮ LIỆU CỦA PHƯƠNG TIỆN VẬN TẢI

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

HÀ NỘI - 2018

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRỊNH BÁ QUÝ

PHÂN TÍCH VÀ MÔ PHỎNG TÌNH TRẠNG GIAO THÔNG DỰA VÀO KHAI PHÁ DỮ LIỆU CỦA PHƯƠNG TIỆN VẬN TẢI

Ngành: Khoa học máy tính

Chuyên ngành: Khoa học máy tính

Mã Số: 8480103.01

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC: **PGS.TS PHAN XUÂN HIẾU**

TS. NGUYỄN VĂN TĂNG

HÀ NỘI - 2018

MỤC LỤC

LỜI CẢM ƠN

Lời đầu tiên, tôi xin bày tỏ sự cảm ơn chân thành đối với Thầy giáo, Tiến sĩ Phan Xuân Hiếu và Thầy giáo, Tiến sĩ Nguyễn Văn Tăng – hai giáo viên hướng dẫn của tôi. Hai thầy đã cho tôi những gợi ý và chỉ dẫn quý báu, cũng như nguồn dữ liệu để thực nghiệm trong đề tài, tôi đã không thể hoàn thành luận văn nếu không có sự chỉ bảo của hai thầy.

Tôi xin cảm ơn Công ty Trách nhiệm hữu hạn phát triển Công nghệ Điện tử Bình Anh và Công ty Cổ phần Công nghệ AIB Việt Nam đã cung cấp dữ liệu phục vụ cho nghiên cứu trong luận văn.

Tôi xin gửi lời cảm ơn tới các Thầy Cô trong khoa Công nghệ thông tin, trường Đại học Công nghệ, Đại học Quốc gia Hà Nội đã dìu dắt, hướng dẫn, dạy dỗ cũng như chỉ bảo và tạo điều kiện cho tôi học tập và nghiên cứu tại trường trong suốt thời gian vừa qua.

Tôi xin cảm ơn những người thân trong gia đình, bạn bè, đồng nghiệp đã quan tâm, động viên giúp đỡ, tạo điều kiện cho tôi trong thời gian học tập và nghiên cứu luận văn tốt nghiệp.

Mặc dù đã cố gắng hoàn thành luận văn nhưng chắc chắn sẽ không tránh khỏi những sai sót, tôi kính mong nhận được sự thông cảm và chỉ bảo của các thầy cô và các bạn.

Xin chân thành cảm ơn.

LỜI CAM ĐOAN

Tôi là Trịnh Bá Quý, học viên lớp Khoa học máy tính K22 xin cam đoan báo cáo luận văn này được viết bởi tôi dưới sự hướng dẫn của Thầy giáo, Tiến sĩ Phan Xuân hiếu và Thầy giáo, Tiến sĩ Nguyễn Văn Tăng. Tất cả kết quả đạt được trong luận văn này là quá trình tìm hiểu, nghiên cứu của riêng tôi. Trong toàn bộ nội dung của luận văn, những điều được trình bày là kết quả của cá nhân tôi hoặc là được tổng hợp từ nhiều nguồn tài liệu khác. Các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày….tháng….năm 2018

Người cam đoan

Trịnh Bá Quý

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

Phân tích dữ liệu giao thông là một công việc quan trọng và có nhiều ý nghĩa trong thực tiễn. Bài toán này đang thu hút sự quan tâm của các đơn vị quản lý và vận hành hạ tầng giao thông cũng như các nhà khoa học trong lĩnh vực liên quan. Phân tích dữ liệu giao thông giúp ích rất nhiều cho các ngành như ngành vận tải: vận chuyển người và hàng hóa đến đích một cách an toàn, tiết kiệm; ngành giao thông: điều phối lưu lượng giao thông, trách ùn tắc giao thông; ngành quy hoạch đô thị: đưa ra những giải pháp trong việc quy hoạch các tuyến đường, nhà ga, bến xe.

Trong khoảng thời gian gần đây, các đối tượng kinh doanh vận tải đều bắt buộc gắn thiết bị giám sát hành trình, và cách thức kinh doanh vận tải cũng được hiện đại hóa bằng cách áp dụng công nghệ thông tin, đặc biệt là những thiết bị di động thông minh. Dữ liệu từ những hệ thống giám sát hành trình, hệ thống nghiệp vụ này phần nào cho phép ta biết được vị trí hiện thời của phương tiện vận tải, biết được những thông tin đi kèm của phương tiện vận tải như vận tốc, người lái, các sai phạm của phương tiện vận tải. Tuy nhiên việc khai thác dữ liệu này còn đang gặp khá nhiều thách thức do lượng dữ liệu lớn, dữ liệu nhiễu nhiều.

Luận văn này nêu phương pháp: (1) phân vùng và phân cụm các cung đường di chuyển theo thời gian để tìm ra quy luật di chuyển của các phương tiện vận tải; (2) Mô phỏng luồng di chuyển của các phương tiện vận tải theo vùng; (3) Xếp hạng các khu vực đón, trả khách; (4) Dự đoán luồng giao thông trong các vùng; (5) Đưa ra gợi ý di chuyển cho tài xế dựa vào mật độ giao thông và kết quả xếp hạng của các vùng. Các bài toán này được thực hiện theo tiếp cận phân tích dữ liệu giao thông, cụ thể là phân tích dữ liệu hành trình thu nhận từ taxi theo thời gian thực và gần thời gian thực.

Bố cục của luận văn được tổ chức như sau:

- **Chương 1: Khái quát bài toán khai phá dữ liệu phương tiện vận tải** giới thiệu tổng quan về bài toán khai phá dữ liệu phương tiện vận tải, định nghĩa và các hướng tiếp cận.
- **Chương 2: Một số nghiên cứu về phân tích, mô phỏng tình trạng giao thông** giới thiệu một số phương pháp, kỹ thuật đã được nghiên cứu và áp dụng cho bài toán phân tích, mô phỏng tình trạng giao thông.

- **Chương 3: Xây dựng hệ thống phân tích, mô phỏng tình trạng giao thông** trình bày mô hình bài toán phân tích và mô phỏng tình trạng giao thông dựa vào khai phá dữ liệu vận tải, quy trình thực hiện giải quyết các bài toán trong luận văn, các đề xuất bổ sung cho các nghiên cứu ở chương 2 để giải quyết các bài toán đặt ra.
- **Chương 4: Thử nghiệm và đánh giá** trình bày quá trình thử nghiệm mô hình đã xây dựng dựa trên hai bộ dữ liệu về taxi, và thực hiện đánh giá độ chính xác của mô hình dự báo
- **Kết luận:** Tổng kết các đóng góp và kết quả đạt được trong quá trình nghiên cứu và thực hiện luận văn, cũng như hướng phát triển trong tương lai để hoàn thiện hơn kết quả nghiên cứu.

Chương 1: Khái quát bài toán khai phá dữ liệu phương tiện vận tải

Ngày nay, với sự phát triển mạnh mẽ và vượt bậc về Công nghệ thông tin, cũng như hạ tầng cơ sở giao thông, việc hiện đại hóa quá trình khai thác, kiểm soát phương tiện vận tải đang được chú trọng triển khai sâu rộng. Điều này thúc đẩy sự gia tăng về dữ liệu của phương tiện vận tải. Các dữ liệu này đến từ các thiết bị giám sát hành trình cũng như các thiết bị đi kèm trong quá trình thực hiện giải quyết các bài toán nghiệp vụ. Vì vậy, nhiều nhà khoa học đã nghiên cứu các công nghệ, thuật toán để giải quyết bài toán về khai phá dữ liệu cách nhanh nhất đáp ứng được những yêu cầu thực tế mà các tổ chức hay doanh nghiệp đưa ra. Chương này sẽ mô tả khái quát về dữ liệu từ phương tiện vận tải [1] cũng như vai trò và ứng dụng của nó.

1.1 Tổng quan về dữ liệu GPS

GPS - Hệ thống định vị toàn cầu là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.

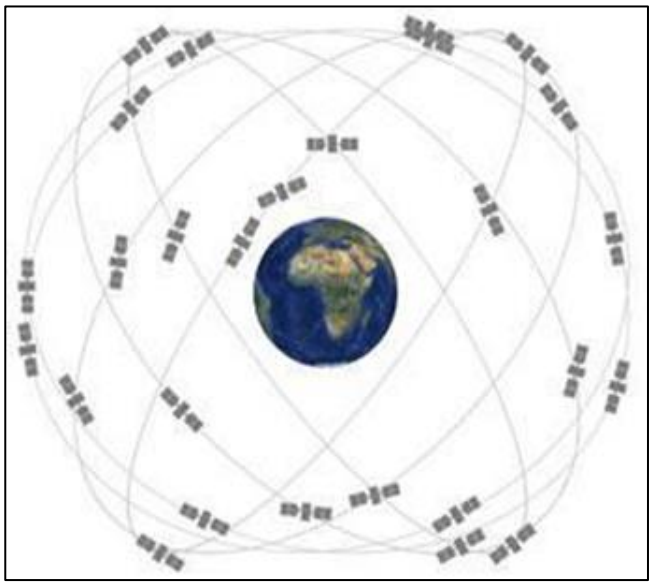
GPS sử dụng nguyên tắc hướng thẳng tương đối của hình học và lượng giác học. Mỗi vệ tinh liên tục phát và truyền dữ liệu trong quỹ đạo của nó, do đó, mỗi thiết bị GPS nhận sẽ liên tục truy cập dữ liệu quỹ đạo chính xác từ vị trí của tất cả vệ tinh. Từ đó tín hiệu hoặc sóng vô tuyến di chuyển ở vận tốc hằng số (thường bằng vận tốc ánh sáng – C), các thiết bị GPS thu có thể tính toán khoảng cách liên quan từ GPS đến các vệ tinh khác bằng cách máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian mà thiết bị GPS thu nhận được tín hiệu do các vệ tinh khác. Nguyên lý xác định toạ độ của hệ thống GPS dựa trên công thức quãng đường bằng vận tốc x thời gian. Vệ tinh phát ra các tín hiệu bao gồm vị trí của chúng, thời điểm phát tín hiệu.

Máy thu tính toán được khoảng cách từ các vệ tinh, giao điểm của các mặt cầu có tâm là các vệ tinh, bán kính là thời gian tín hiệu đi từ vệ tinh đến máy thu nhân vận tốc sóng điện từ là toạ độ điểm cần định vị.

GPS hiện tại gồm 3 phần chính: Phần không gian, phần kiểm soát và phần sử dụng.

1.1.1 Phần không gian

Phần không gian gồm 27 vệ tinh (24 vệ tinh hoạt động và 3 vệ tinh dự phòng) nằm trên các quỹ đạo xoay quanh trái đất. Chúng cách mặt đất 20.200 km, bán kính quỹ đạo 26.600 km.



Hình 1.1 Vệ tinh GPS

Chúng chuyển động ổn định và quay hai vòng quỹ đạo trong khoảng thời gian gần 24 giờ với vận tốc 7 nghìn dặm một giờ. Các vệ tinh trên quỹ đạo được bố trí sao cho các máy thu GPS trên mặt đất có thể nhìn thấy tối thiểu 4 vệ tinh vào bất kỳ thời điểm nào.

Các vệ tinh được cung cấp bằng năng lượng Mặt Trời. Chúng có các nguồn pin dự phòng để duy trì hoạt động khi chạy khuất vào vùng không có ánh sáng Mặt Trời. Các tên lửa nhỏ gắn ở mỗi quả vệ tinh giữ chúng bay đúng quỹ đạo đã định.

1.1.2 Phần kiểm soát

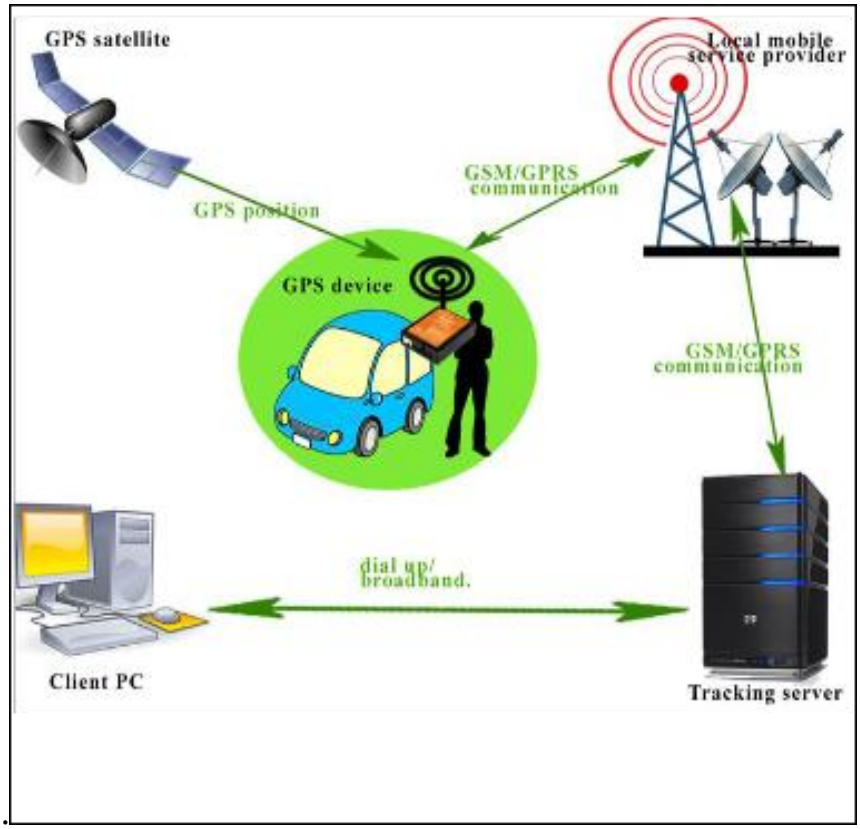
Mục đích phần này là kiểm soát vệ tinh đi đúng hướng theo quỹ đạo và thông tin thời gian chính xác. Có 5 trạm kiểm soát đặt rải rác trên trái đất. Bốn trạm kiểm soát hoạt động một cách tự động, và một trạm kiểm soát là trung tâm. Bốn trạm này nhận tín hiệu liên tục từ những vệ tinh và gửi các thông tin này đến trạm kiểm soát trung tâm. Tại trạm kiểm soát trung tâm, nó sẽ sửa lại dữ liệu cho đúng và kết hợp với hai ăng-ten khác để gửi lại thông tin cho các vệ tinh. Ngoài ra, còn một trạm kiểm soát trung tâm dự phòng và sáu trạm quan sát chuyên biệt.

1.1.3 Phần sử dụng

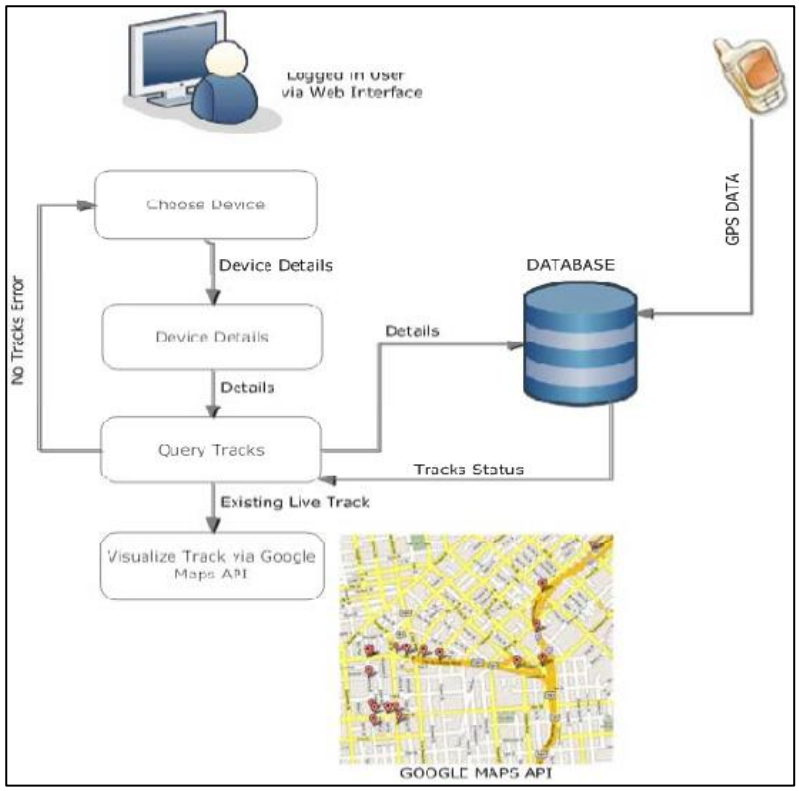
Phần sử dụng là thiết bị nhận tín hiệu vệ tinh GPS và người sử dụng thiết bị này.

1.2 Dữ liệu phương tiện vận tải

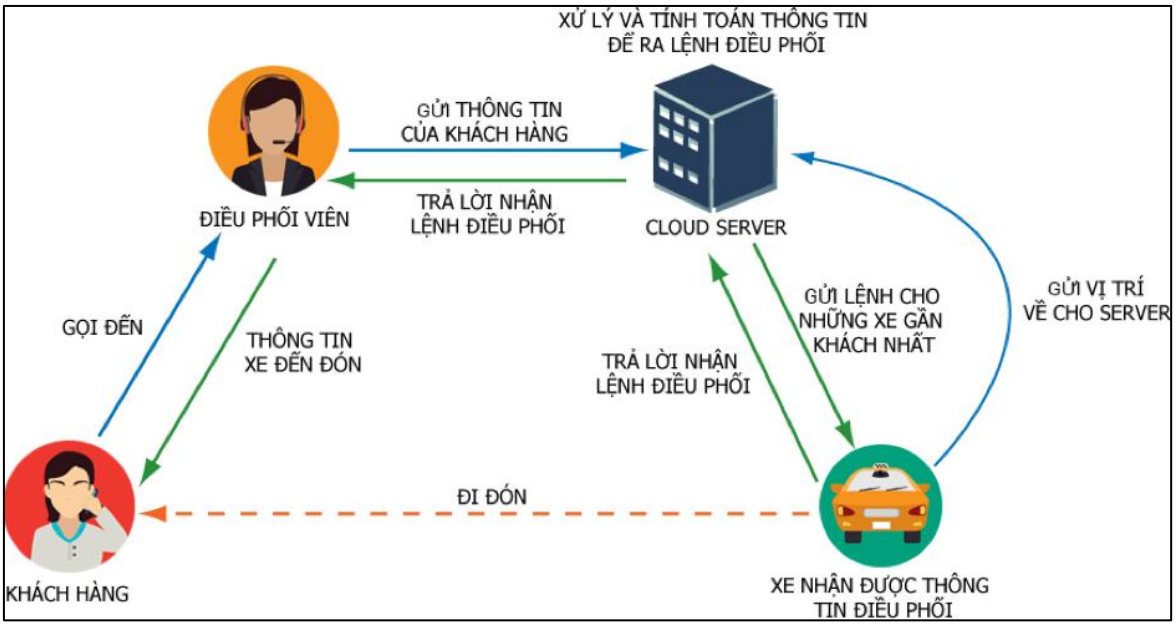
Dữ liệu phương tiện vận tải chính là một loại dữ liệu GPS, nó có thể có từ các thiết bị giám sát hành trình sử dụng những công nghệ có sẵn: Hệ thống định vị ô tô, GPSylon, Open GTS [10], dữ liệu từ các thiết bị di động thông minh dùng trong việc giải quyết các bài toán nghiệp vụ [9] (điển hình là các ứng dụng gọi xe xuất hiện ở Việt Nam gần đây)



Hình 1.2 Dữ liệu đến từ các thiết bị giám sát hành trình



Hình 1.3 Kiến trúc của hệ thống định vị sử dụng thiết bị di động thông minh



Hình 1.4 Sơ đồ hoạt động của một ứng dụng gọi xe taxi sử dụng thiết bị di động thông minh

Dữ liệu phương tiện vận tải có thể có từ các thiết bị giám sát hành trình sử dụng những công nghệ có sẵn: Hệ thống định vị ô tô, GPSylon, Open GTS [10], dữ liệu từ các thiết bị di động thông minh dùng trong việc giải quyết các bài toán nghiệp vụ [9] (điển hình là các ứng dụng gọi xe xuất hiện ở Việt Nam gần đây).

Nguồn dữ liệu được lấy từ chính bài toán kinh doanh của các công ty, cá nhân kinh doanh vận tải nên thường bao gồm thêm một số thông tin hữu ích có thể khai thác như: thông tin của các cảm biến (đóng mở cửa, điều hòa, xăng dầu, mắt thần trên taxi). Khái quát về dữ liệu của một phương tiện vận tải:

- Thời gian (tính bằng giây)
- Kinh độ
- Vĩ độ
- Cao độ
- Vận tốc (do thiết bị thu nhận từng giây, có thể được tính tương đối từ 4 thông tin đầu)
- Hướng di chuyển (do thiết bị thu nhận từng giây, có thể được tính tương đối từ 4 thông tin đầu)
- Trạng thái (do thiết bị thu nhận từng giây, do các dây cảm biến trên thiết bị hành trình gắn với những thành phần cụ thể trên xe)

1.3 Các ứng dụng của khai phá dữ liệu phương tiện vận tải

Nhìn chung, ứng dụng của khai phá dữ liệu phương tiện vận tải khá rộng, tuy nhiên có thể được chia làm những mục chính sau [1, 2]:

- Dịch vụ Hỗ trợ lập kế hoạch giao thông.
- Dịch vụ Quản lý và bảo trì cơ sở hạ tầng giao thông.
- Dịch vụ Giám sát và điều khiển giao thông.
- Dịch vụ Quản lý cơ sở dữ liệu tai nạn giao thông.
- Dịch vụ Quản lý nhu cầu giao thông.
- Dịch vụ Hỗ trợ giám sát việc chấp hành luật giao thông.
- Dịch vụ Hỗ trợ quản lý thông tin phương tiện vận tải.
- Dịch vụ Hỗ trợ quản lý thông tin lái xe.

Luận văn này tập trung vào mảng ứng dụng "Dịch vụ Giám sát và điều khiển giao thông" – là một nhu cầu bức thiết hiện nay để giải quyết các vấn đề về tắc đường, quy hoạch đô thị với các bài toán cụ thể:

- Phân vùng và phân cụm các cung đường di chuyển theo thời gian để tìm ra quy luật di chuyển của các phương tiện vận tải.
- Mô phỏng luồng di chuyển của các phương tiện vận tải theo vùng.
- Xếp hạng các khu vực đón, trả khách.
- Dự đoán luồng giao thông trong các vùng.

- Đưa ra gợi ý di chuyển cho tài xế dựa vào mật độ giao thông và kết quả xếp hạng của các vùng.

Kết luận: Chương 1 của luận văn trình bày tổng quan về dữ liệu GPS, gồm nguyên lý và các phần của dữ liệu GPS, đưa ra hai nguồn của dữ liệu phương tiện vận tải – một loại dữ liệu GPS là qua các thiết bị giám sát hành trình và các ứng dụng quản lý nghiệp vụ trên thiết bị di động thông minh, mô tả khái quát về dữ liệu vận tải. Đồng thời chương này cũng nêu ra được các ứng dụng của dữ liệu phương tiện vận tải, chỉ ra những ứng dụng của khai phá dữ liệu phương tiện vận tải mà luận văn tập trung.

Chương 2: Một số nghiên cứu về phân tích, mô phỏng tình trạng giao thông

Như đã đề cập trong chương 1, luận văn tập trung vào những bài toán cụ thể sau:

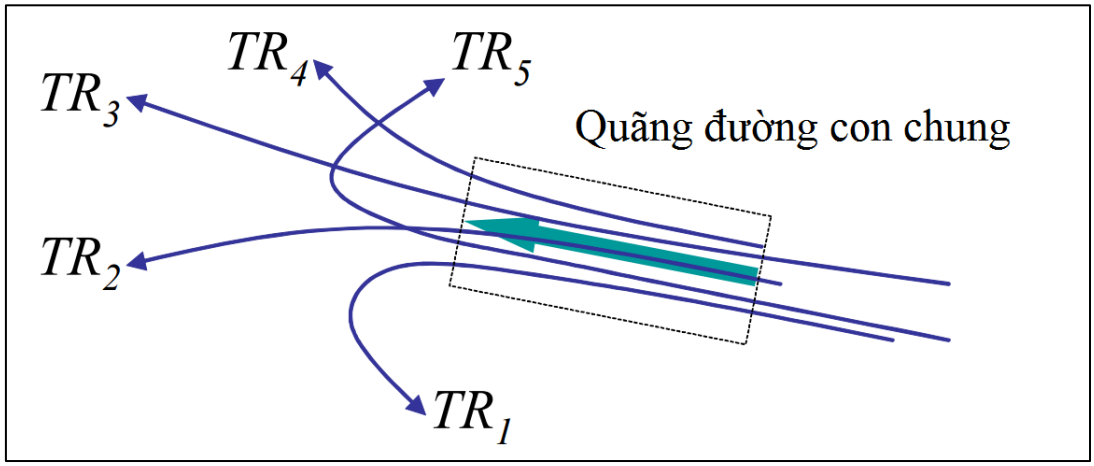
- **Phân vùng và phân cụm các cung đường di chuyển theo thời gian để tìm ra quy luật di chuyển của các phương tiện vận tải:** Cụ thể ở đây luận văn tiến hành phân tích dữ liệu của nhiều taxi trong cùng một ngày, trong một khoảng thời gian nhất định để tìm ra các cụm (các cung đường chung), loại bỏ những dữ liệu nhiễu, cụm không đặc trưng, phục vụ cho bài toán mô phỏng luồng di chuyển, tìm ra các đường đi chung, các đường đi tối ưu phục vụ cho bài toán gợi ý di chuyển. Phương pháp phân cụm thường chia thành [7]: không giám sát, giám sát, bán giám sát. Luận văn lựa chọn phương pháp không giám sát, cụ thể là mô hình và thuật toán Trajectory clustering của Jae-Gil Lee và cộng sự [6] sẽ trình bày bên dưới.
- **Mô phỏng luồng di chuyển của các phương tiện vận tải theo vùng:** Nhằm đạt mục tiêu khái quát hóa và tăng hiệu năng tính toán luận văn sử dụng tư tưởng chia vùng theo công trình của Naoto [8] và cách chia cung thời gian theo công trình của Xiaomeng Wang và cộng sự [15] và đề xuất cách biểu diễn mật độ theo vận tốc
- **Xếp hạng các khu vực đón, trả khách:** Luận văn thực hiện khái quát hóa khu vực đón, trả khách theo tư tưởng chia vùng trong công trình của Naoto [8] và cách chia cung thời gian trong công trình của Xiaomeng Wang và cộng sự [15]
- **Dự đoán luồng giao thông trong các vùng:** Luận văn thực hiện dự đoán vùng đến kế tiếp theo công trình của S´ebastien Gambs và cộng sự [11, 12] với cách gán nhãn dựa trên xếp hạng và mật độ, phục vụ cho bài toán gợi ý di chuyển tiếp theo
- **Đưa ra gợi ý di chuyển cho tài xế dựa vào mật độ giao thông và kết quả xếp hạng của các vùng:** Dựa trên bài toán dự đoán luồng giao thông và xếp hạng đón khách, luận văn thực hiện đưa ra các gợi ý di chuyển cho tài xế, sử dụng các cung đường đã phân cụm để gợi ý cung đường tốt nhất.

2.1 Thuật toán phân cụm TRACLUS

Phân cụm là cách chia các đối tượng dữ liệu thành các nhóm sao cho các đối tượng trong cùng một nhóm gần nhau hơn và các đối tượng của hai nhóm khác nhau thì khác nhau rất nhiều. Trong luận văn, bài toán phân cụm cho phép tìm hiểu các quy luật quãng đường của từng taxi. Các quy luật đường đi của taxi gồm có các đoạn đường được taxi dùng để di chuyển nhiều nhất, các cụm quãng đường sẽ được phân ra dựa trên khoảng cách thực tế.

Để giải quyết hai bài toán trên luận văn sử dụng công trình của Jae-Gil Lee và cộng sự [6], đó là thuật toán TRACLUS.

Để làm rõ thuật toán, giả sử có 5 quãng đường như trong Hình 2.1, có thể nhìn rõ rằng có một đặc điểm chung, biểu diễn bằng mũi tên trong hình chữ nhật. Tuy vậy, nếu nhóm những quãng đường này làm một, chúng ta không thể khám phá đặc điểm chung này khi mà chúng di chuyển đi các hướng khác nhau, vì vậy sẽ bị mất một số thông tin quý giá.



Hình 2.1 Mô hình quãng đường con chung

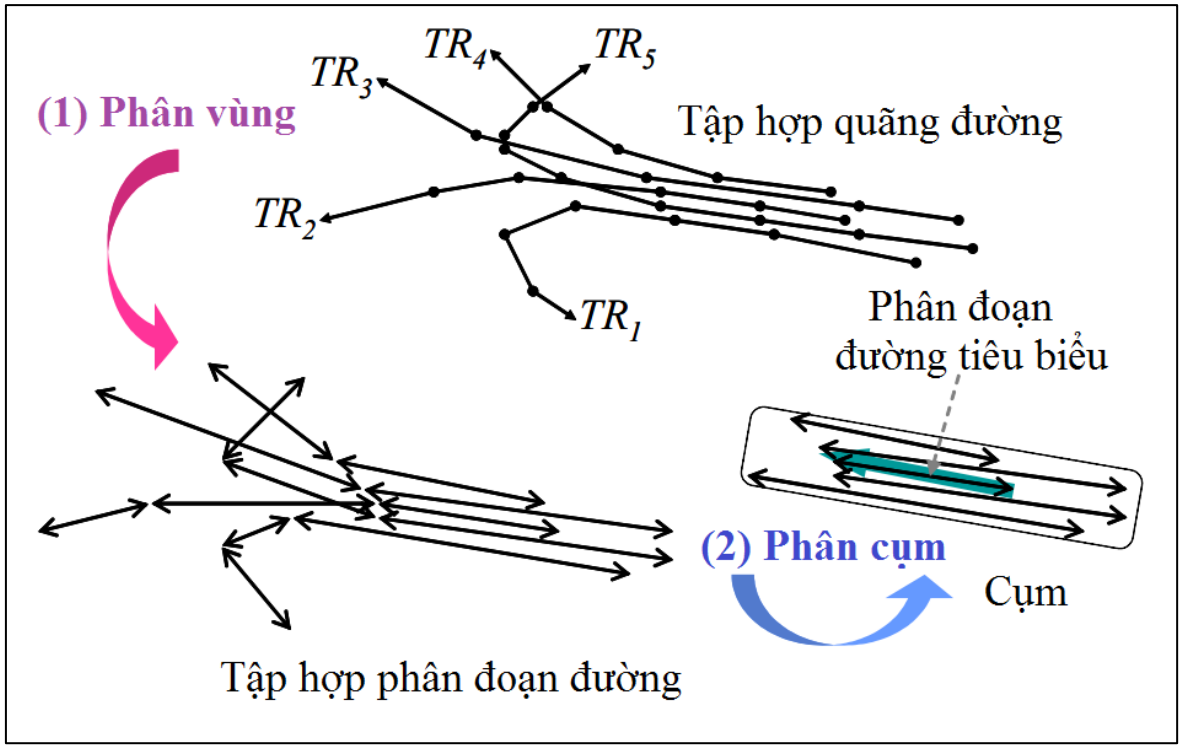
Giải pháp ở đây sẽ là phân chia các quãng đường thành tập hợp các phân đoạn đường và sau đó nhóm các phân đoạn đường. Công việc này nằm trong khuôn khổ phân vùng và cụm. Mục tiêu chính của việc phân vùng và cụm này là khám phá các quãng đường con (phân đoạn đường) chung từ bộ dữ liệu quãng đường đầu vào.

Việc khám phá các quãng đường con là rất hữu ích do chúng ta có những vùng quan tâm đặc biệt để phân tích. Trong trường hợp này, chúng ta tập trung vào những hành vi cụ thể trong khu vực đó.

Phương pháp phân vùng và cụm sẽ gồm 2 giai đoạn:

- Bước phân vùng: Mỗi quãng đường được tối ưu phân chia làm các phân đoạn đường. Các phân đoạn đường này sẽ là dữ liệu đầu vào cho bước tiếp theo.
- Bước phân cụm: các phân đoạn đường giống nhau được nhóm vào một cụm. Trong bài báo này, thuật toán phân cụm dựa trên mật độ được sử dụng.

Hình 2.2 miêu tả toàn bộ quá trình phân cụm quãng đường trong phương pháp phân vùng và cụm. Đầu tiên, mỗi quãng đường được chia ra làm các phân đoạn đường. Sau đó, các phân đoạn đường mà chúng ở gần nhau dựa trên tiêu chí khoảng cách được nhóm thành một cụm. Cuối cùng, đoạn đường tiêu biểu được tạo ra cho mỗi cụm. Thuật toán này được viết lại chi tiết như trong Thuật toán 1.



Hình 2.2 Ví dụ về phân vùng và cụm quãng đường

Thuật toán 1: TRACLUS (TRAjectory CLUStering)

Input: *Tập hợp quãng đường* $I = \{TR_1, \cdots, TR_{numtra}\}$

Output: (1) tập hợp các cụm $O = \{C_1, \cdots, C_{numclus}\}$

(2) tập hợp các đoạn đường tiêu biểu

Thuật toán:

/* **BƯỚC PHÂN VÙNG** */

01: **for each** (*TR* ∈ *I*) **do**

/* **Thuật toán 2** */

02: Thực hiện thuật toán phân vùng quãng đường; Nhận tập hợp *L* của các phân đoạn đường;

03: Tích lũy *L* vào trong một tập hợp *D*;

/* **BƯỚC PHÂN CỤM** */

/* **Thuật toán 3** */

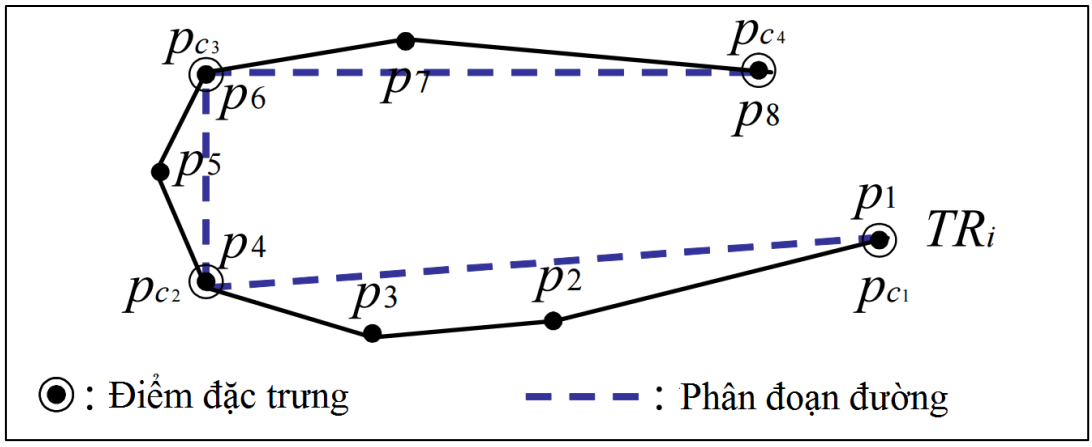
04: Thực hiện phân cụm phân đoạn đường cho *D*; kết quả gồm một tập hợp *O* gồm các cụm;

05: **for each** (*C* ∈ *O*) **do**

06: Thực hiện việc tạo *đoạn đường tiêu biểu*; kết quả gồm có *đoạn đường tiêu biểu*;

2.1.1 Phân vùng quãng đường

Chúng ta muốn tìm những điểm mà hành vi của các quãng đường thay đổi nhanh chóng, chúng ta gọi những điểm này là những điểm đặc trưng. Đối với mỗi $TR_i = p_1\ p_2\ p_3 \ldots p_{leni}$, chúng ta xác định một tập hợp các điểm đặc trưng $\{p_{c1}, p_{c2}, p_{c3}, \ldots, p_{cpari}\}$ $(c_1 < c_2 < \ldots < c_{pari})$. Mỗi điểm p_i tương ứng với một tọa độ gồm kinh độ và vĩ độ (X và Y trong tệp dữ liệu đầu vào). Sau đó TR_i được phân vùng tại mỗi điểm đặc trưng, và mỗi vùng được biểu diễn bởi phân đoạn đường. Hình 2.3 miêu tả một ví dụ về quãng đường và cách nó được phân đoạn.



Hình 2.3 Ví dụ về quãng đường và các phân đoạn

Việc phân chia tối ưu cần phải có hai tính chất sau: chính xác và súc tích. Tính chính xác có nghĩa rằng sự khác nhau giữa quãng đường và một tập hợp phân đoạn đường càng nhỏ càng tốt. Tính súc tích đồng nghĩa với số lượng phân đoạn càng ít càng tốt. Để thực hiện điều này chúng ta dùng thuật toán 2.

Thuật toán 2: Approximate Trajectory Partitioning

Input: $TR_i = p_1\ p_2\ p_3 \ldots . p_j \ldots p_{leni}$

Output: Tập hợp các điểm đặc trưng CP_i

Thuật toán:

01: Thêm $p_1$vào tập hợp CP_i; /* điểm bắt đầu */

02: *startIndex:*= 1, *length*:= 1;

03: **while** (*startIndex* + *length* ≤ len_i) **do**

04: *currIndex*:= *startIndex* + *length*;

05: $cost_{par}$:= MDL_{par}*(pstartIndex, pcurrIndex)*;

06: $cost_{nopar}$:= MDL_{nopar}*(pstartIndex, pcurrIndex)*;

/* kiểm tra nếu phân vùng ở điểm hiện tại làm MDL lớn hơn khi không phân vùng */

07: **if** ($cost_{par}$> $cost_{nopar}$) **then**

/* phân vùng điểm trước đo */

08: Thêm $p_{currIndex-1}$ vào trong CP_i;

09: *startIndex* := *currIndex* − 1, *length* := 1;

10: **else**

11: length := length + 1;

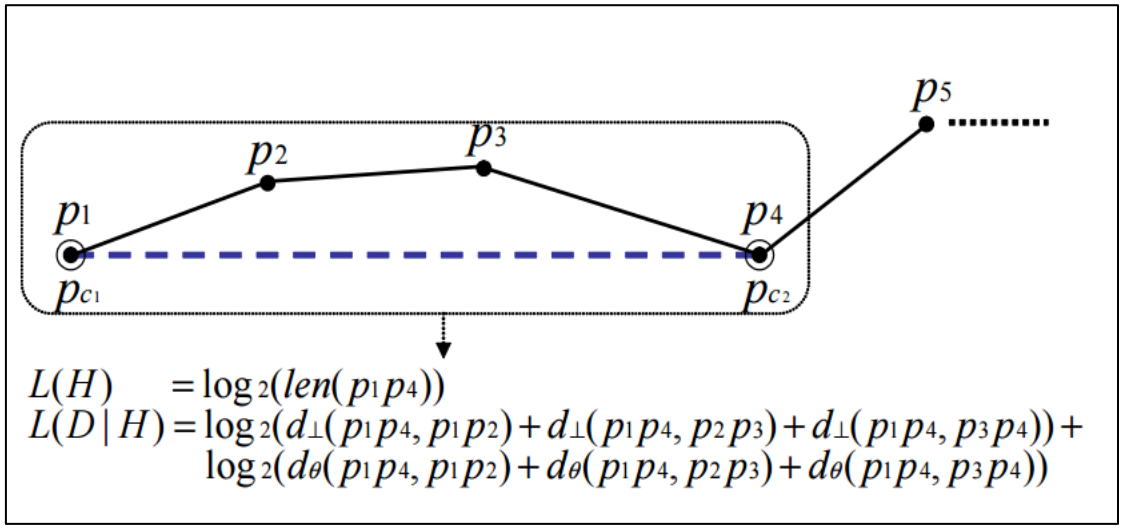
12: Thêm điểm p_{leni} vàoCP_i; /* điểm kết thúc */

Chiều dài tối thiểu mô tả (MDL - Minimum Description Length) được sử dụng để phân vùng đoạn đường. MDL sẽ được đo dựa trên L(H) (độ đo tính súc tích) và L(D|H) (độ đo tính chính xác). Công thức tính của L(H) và L(D|H) lần lượt như sau:

$$L(H) = \sum_{j=1}^{par_i - 1} \log_2(len(p_{c_j} p_{c_{j+1}}))$$

$$L(D|H) = \sum_{j=1}^{par_i - 1} \sum_{k=c_j}^{c_{j+1}-1} \{\log_2(d_\perp(p_{c_j} p_{c_{j+1}}, p_k p_{k+1})) + \log_2(d_\theta(p_{c_j} p_{c_{j+1}}, p_k p_{k+1}))\}$$

trong đó $d_\perp$ và d_θ lần lượt là khoảng cách vuông góc và khoảng cách góc giữa 2 phân đoạn đường $L_i = s_i e_i$ và $L_j = s_j e_j$.



Hình 2.4 Cách tính độ đo MDL

Dựa vào công thức trên, và ví dụ trong Hình 2.4, tính được L(H) và L(D|H) cho quãng đường {p_1 p_2 p_3 p_4 p_5 ...}.

2.1.2 Phân cụm

Trong thuật toán TRACLUS, thuật toán phân cụm DBSCAN được sử dụng. Đối với thuật toán DBSCAN, chúng ta cần xác định 2 tham số: ε (tương ứng với khoảng cách nhỏ nhất giữa 2 điểm để có thể gọi là điểm hàng xóm) và minPts (tương ứng với số lượng điểm hàng xóm).

Nε(L) được gọi là các hàng xóm của phân đoạn đường L ∈ D trong khoảng cách bán kính ε: $N_\varepsilon(L_i) = \{L_j \in D \mid dist(L_i, L_j) \leq \varepsilon\}$.

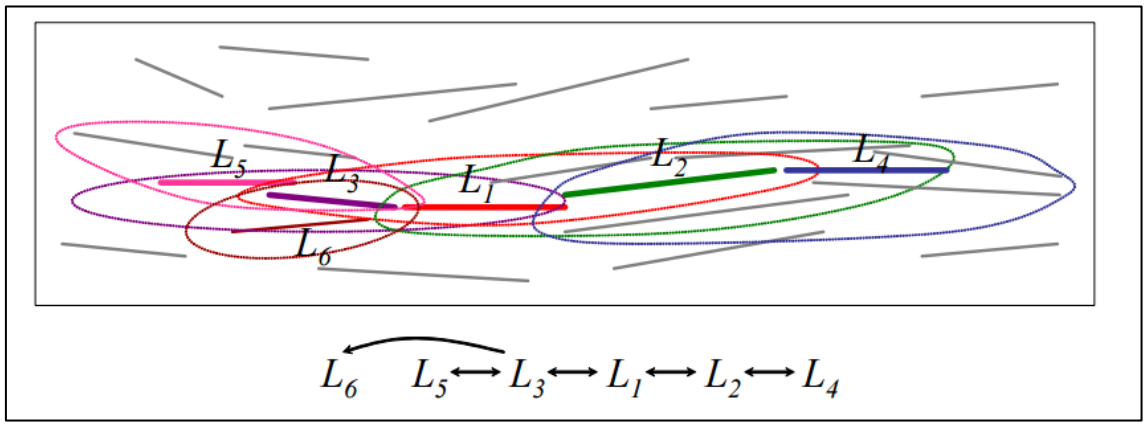
Phân đoạn đường $L_i \in D$ được gọi là phân đoạn đường với điều kiện ε và MinLns thỏa mãn nếu $|N_\varepsilon(L_i)| \geq$ MinLns và sẽ gọi là ngoại bên nếu không thỏa mãn điều kiện này.

Một phân đoạn đường $L_i \in D$ được coi là có khả năng truy cập mật độ trực tiếp (directly density reachable) từ một phân đoạn đường khác $L_j \in D$ với điều kiện ε và MinLns thỏa mãn nếu $L_i \in N\varepsilon(L_j)$ và $|N\varepsilon(L_j)| \geq$ MinLns.

Một phân đoạn đường $L_i \in D$ được gọi là có khả năng truy cập mật độ từ một phân đoạn đường khác $L_j \in D$ với điều kiện ε và MinLns thỏa mãn nếu có một chuỗi các đoạn đường L_j , L_{j-1}, $\cdots$, L_{i+1}, $L_i \in D$ sao cho L_k là mật độ truy cập trực tiếp từ L_{k+1} với điều kiện ε và MinLns thỏa mãn.

Một phân đoạn đường $L_i \in D$ được gọi là mật độ kết nối (density-connected) tới một phân đoạn đường khác $L_j \in D$ với điều kiện ε và MinLns thỏa mãn nếu có một phân đoạn đường $L_k \in D$ sao cho cả hai $L_i$và $L_j$là có khả năng truy cập mật độ từ L_k.

Chúng ta hãy nghiên cứu ví dụ trong Hình 2.5 sau được áp dụng DBSCAN cho bài toán phân cụm các đoạn đường. Ở đây minPts = 3 và ε là các hình eclipse, dựa vào định nghĩa trong DBSCAN chúng ta sẽ có:



Hình 2.5 Ví dụ về mật độ truy cập và mật độ kết nối

- L1, L2, L3, L4, và L5 là phân đoạn đường chính
- L2 và L3 có mật độ truy cập trực tiếp từ L1
- L6 có mật độ truy cập từ L1 nhưng ngược lại không đúng
- L1, L4 và L5 là mật độ kết nối.

Thuật toán phân cụm trong TRACLUS được viết lại như trong thuật toán 3:

Thuật toán 3: Phân cụm

Input: (1) Một tập hợp phân đoạn $D = \{L1, \ldots., L_{num_{ln}}\}$,
(2) Hai tham số ε and *MinLns*

Đầu ra: Một tập hợp cụm $O = \{C1, \ldots, C_{num_{clus}}\}$

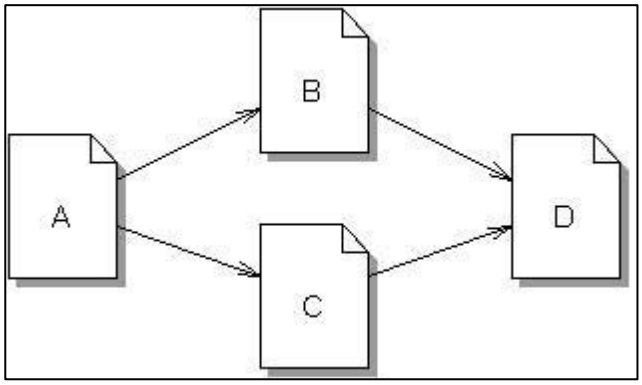
Thuật toán:

```
/* Bước1 */
01: clusterId = 0; /* khởi tạo id đầu tiên */
02: Để tất cả các đoạn đường trong D là chưa được phân loại;
03: for each (L ∈ D) do
04:     if (L chưa được phân loại) then
05:         Compute Nε(L);
06:         if (|Nε(L)| ≥ MinLns) then
07:             Gán clusterId cho∀X ∈ Nε(L);
08:             Thêm Nε(L) − {L} vào trong hàng đợi Q;
/* Step 2 */
09:             ExpandCluster(Q, clusterId, ε, M inLns);
10:            Tăng clusterId thêm 1; /* id mới */
11:         else
12:             Đặt L như ngoại biên;
/* Step 3 */
13: Chỉ định ∀L ∈ D cho cụm C_clusterId;
    /* kiểm tra số lượng quãng đường */
14: for each (C ∈ O) do
    /* một ngưỡng khác MinLns có thể sử dụng */
15:    if (|PTR(C)| < MinLns) then
16:         Xóa C khỏi tập hợp các cụm O;
    /* Step 2: tính tập mật độ kết nối */
17: ExpandCluster(Q, clusterId, ε, MinLns) {
18:     while (Q ≠ ∅) do
19:         Let M be the first line segment in Q;
20:         Compute Nε(M);
21:         if (|Nε(M)| ≥ MinLns) then
22:             for each (X ∈ Nε(M)) do
23:                 if (X chưa được phân loại hoặc ngoại biên) then
24:                     Gán clusterId cho X;
25:                  if (X chưa được phân loại) then
26:                      Thêm X vào hàng đợi Q;
27:         Bỏ M ra khỏi hàng đợi Q;
28: }
```

2.2 Mô hình giao thông dựa trên "PageRank"

2.2.1 Xếp hạng bằng duyệt web

Thuật toán PageRank là một trong những thuật toán xếp hạng trang web được sử dụng rộng rãi nhất, dựa trên giả thuyết rằng nếu một trang web có những liên kết quan trọng đến nó, thì liên kết của nó đến các trang khác cũng trở nên quan trọng. Do vậy PageRank tính toán các backlink(liên kết đến trang đó) và chia sẻ xếp hạng thông qua liên kết: Một trang có xếp hạng cao nếu tổng của các trang có liên kết đến nó cao [13].



Hình 2.6 Ví dụ về backlink

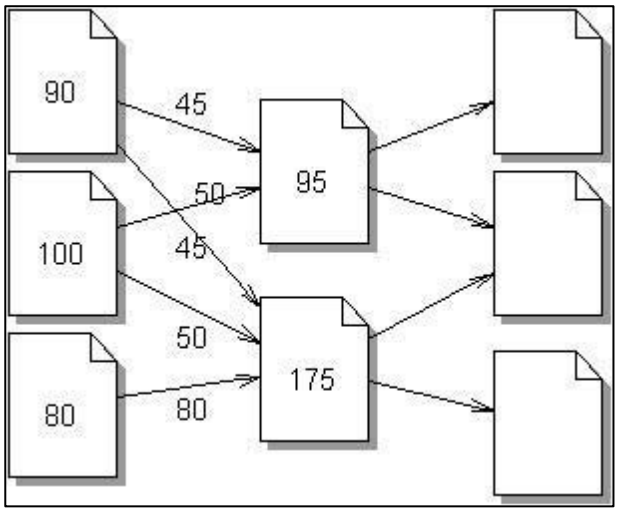
Theo như Hình 2.6 trang A là backlink của trang B, và trang C, trong khi trang B và D là backlink của trang D. Trong hình trên ta có 2 loại liên kết (link): in-link và out-link. Liên kết giữa trang A và trang B được coi là in-link đối với B và coi là out-link với A.Công thức cơ bản của PageRank được định nghĩa như sau:

$$PR(u) = \sum_{v \in B(u)} \frac{PR(v)}{N_v}$$

Trong công thức ta có những ký hiệu sau:

- u đại diện cho một trang web
- $B(u)$ là một tập hợp các trang có liên kết đến u
- PR(u) và PR(v) là số điểm xếp hạng của trang u và trang v
- N_v là số lượng out-link từ trang v

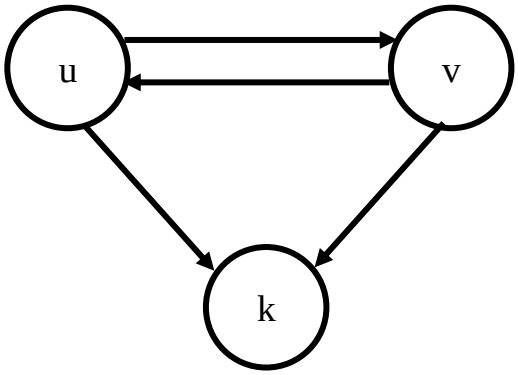
Theo công thức PR(u) là tổng của các xếp hạng PR(v) chia cho số lượng out-link. Quá trình lan truyền này được lặp lại cho đến khi thứ hạng của tất cả các trang web được hội tụ. Quá trình trên được gọi là duyệt web: mang xếp hạng từ một trang web v đến một trang liên kết u



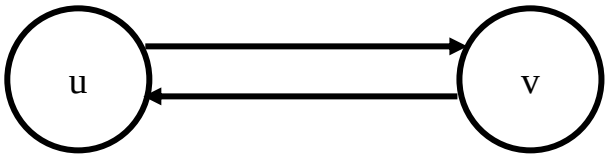
Hình 2.7 Ví dụ về phiên bản đơn giản của PageRank

2.2.2 Damping factor trong PageRank

Có một khái niệm quan trọng trong PageRank gọi là "damping factor" sử dụng trong quá trình chuyển thứ hạng. Khái niệm được sử dụng để tránh vấn đề đường cụt được mô tả ở hình 2.8 và hình 2.9 . Ở hình 2.8 có 3 trang web: u, v, k nhưng không có out-link từ k. Vì vậy tất cả xếp hạng đều dồn ở k và không ra ngoài. Ở hình 2.9, có 2 trang web u và v, nhưng hai trang này liên kết với nhau. Vì vậy xếp hạng chỉ chuyển giữa hai trang với nhau.



Hình 2.8 Không có outlink từ trang k



Hình 2.9 Chuyển xếp hạng giữa hai trang u và v

Để xử lý những vấn đề này trang web có thể nhảy đến một trang ngẫu nhiên khác với tỷ lệ α.

Khả năng nhảy này trong PageRank đặc trưng bởi hệ số "damping factor" (d). Hệ số này thường được đặt là 0.85. Công thức trở thành:

$$PR(u) = 1 - d + d \sum_{v \in B(u)} \frac{PR(v)}{N_v}$$

2.2.3 PageRank có trọng số

Định nghĩa trên của PageRank có một giả định là xếp hạng của một trang được chia đều cho tất cả những trang nó có liên kết. Ví dụ trang A có bốn liên kết in-link đến từ bốn trang B, C, D và E. Theo công thức PageRank [13] mỗi trang trong bốn trang trên đóng góp cho A xếp hạng như nhau. Tuy nhiên giả định này không đúng trong thực tế. Những trang quan trọng hơn hay phổ biến hơn thường có tỷ lệ chia sẻ xếp hạng cao hơn. Nói cách khác xếp hạng chuyển đến một trang web A từ các trang khác phụ thuộc vào độ phổ biến của các liên kết của nó (in-link và out-link) [14]

Độ phổ biến được tính từ in-link và out-link được ký hiệu là: $W^{in}_{(v,u)}$ và $W^{out}_{(v,u)}$

$W^{in}_{(v,u)}$ là trọng số của *link(v,u)* tính dựa trên số lượng in-link của trang u và số lượng in-link của tất cả những trang có liên kết từ trang v

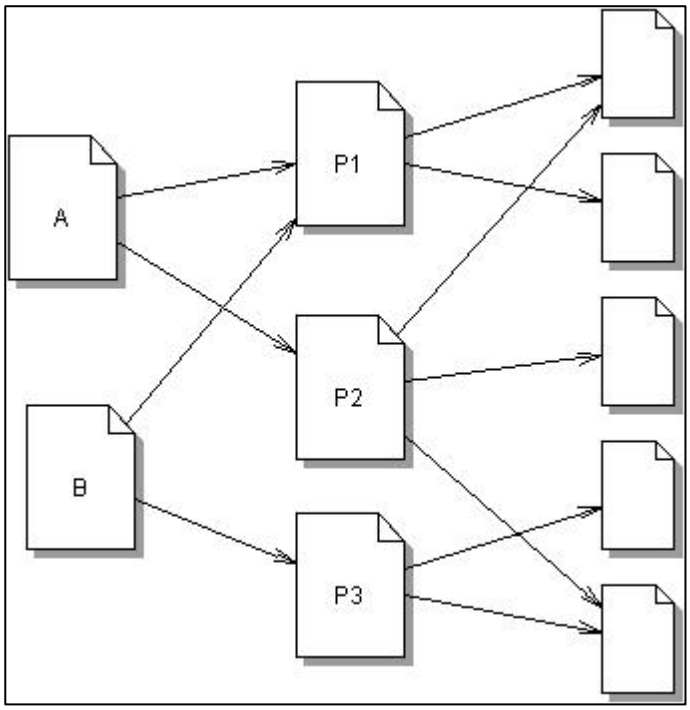
$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p}$$

Ở đây I_u và I_p đại diện cho số in-link của trang u và trang p, R(v) đại diện những trang mà trang v liên kết đến (những trang có in-link từ v)

$W^{out}_{(v,u)}$ là trọng số của *link(v,u)* tính dựa trên số lượng out-link của trang u và số lượng out-link của tất cả những trang có liên kết từ trang v

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p}$$

Ở đây O_u và O_p đại diện cho số out-link của trang u và trang p, R(v) đại diện những trang mà trang v liên kết đến (những trang có in-link từ v)



Hình 2.10 Liên kết của những trang web

Ở hình 2.10 là ví dụ cho liên kết của những trang web. Trang A có 2 trang liên kết đến: p1 và p2. Số lượng in-link và out-link của hai trang này là:

- $I_{p1} = 2$, $I_{p2} = 1$
- $O_{p1} = 2$ và $O_{p2} = 3$

Vì vậy

$$W^{in}_{(A,p1)} = \frac{I_{p1}}{I_{p1} + I_{p2}} = \frac{2}{3}$$

$$W^{out}_{(A,p1)} = \frac{O_{p1}}{O_{p1} + O_{p2}} = \frac{2}{5}$$

PageRank có trọng số được định nghĩa như sau:

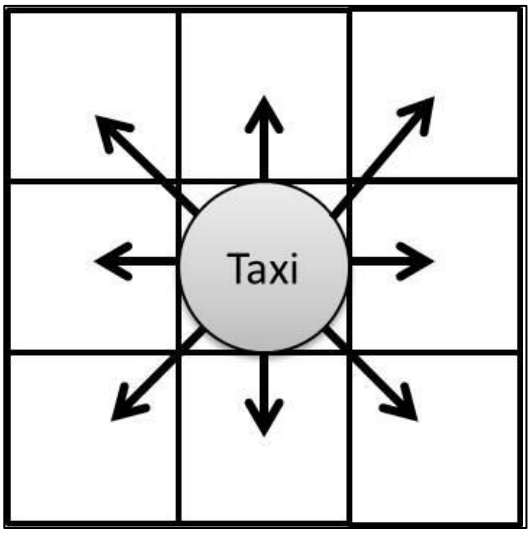
$$PR(u) = 1 - d + d \sum_{v \in B(u)} PR(v)\, W^{in}_{(v,u)} W^{out}_{(v,u)}$$

2.2.4 Xếp hạng bằng taxi

Luận văn áp dụng tư tưởng của PageRank có trọng số cho mô hình giao thông bằng cách thay quá trình duyệt web bằng quá trình di chuyển của taxi [8]. Có nghĩa là một chiếc taxi sẽ mang xếp hạng từ một vùng $M_{(i,j)}$ đến một vùng lân cận $M_{(i',j')}$ như Hình 2.11. Giống với tư tưởng của PageRank có trọng số, luồng

giao thông có xu hướng được chia sẻ xếp hạng cho những vùng quan trọng hơn (những địa điểm, đường phổ biến nhiều người biết) thay vì chia giá trị xếp hạng của một vùng đồng đều cho những vùng nó có liên kết.

Giả định rằng xếp hạng bởi taxi có thể tìm thấy một vùng có ảnh hưởng lớn đến luồng giao thông.



Hình 2.11 Xếp hạng bởi taxi

Trong trường hợp của các trang web, tất cả trang web có liên kết đều có thể là đối tượng cho quá trình lướt web – quá trình chuyển thứ hạng, trong khi đó taxi chỉ có thể chuyển đến các vùng lân cận. Vì vậy khả năng chuyển dịch từ vùng hiện tại $M_{(i,j)}$ đến vùng lân cận $M_{(i',j')}$ được định nghĩa bằng ma trận $P_{i,j}$. Có 8 vùng lân cận từ vùng $M_{(i,j)\ hiện}$ thời, và mỗi khả năng được ký hiệu bởi $p_{i',j'}$ (khả năng trở lại vùng hiện thời là 0).

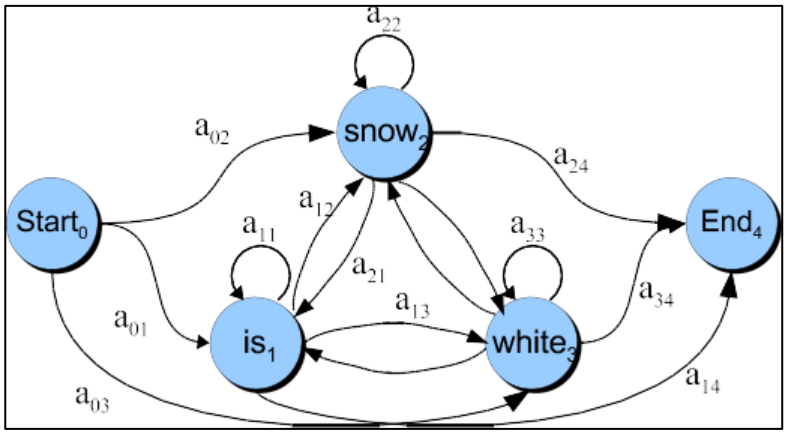
$$p_{i,j} = \begin{pmatrix} p_{i-1,j-1} & p_{i,j-1} & p_{i+1,j-1} \\ p_{i-1,j} & 0 & p_{i+1,j} \\ p_{i-1,j+1} & p_{i,j+1} & p_{i+1,j+1} \end{pmatrix}$$

2.3 Sử dụng xích Markov trong dự đoán điểm đến tiếp theo

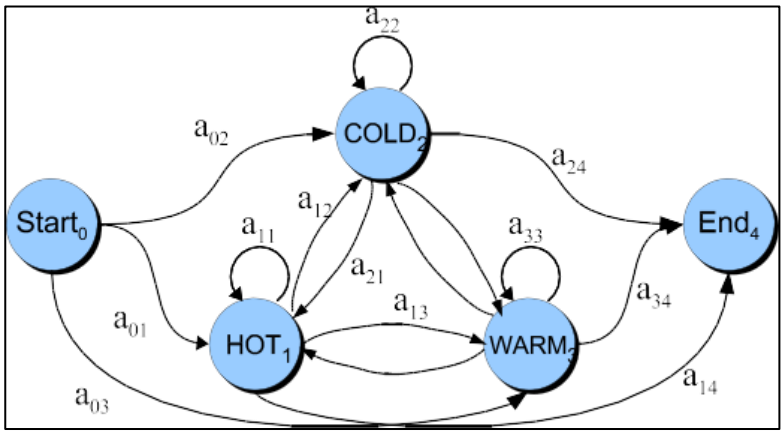
2.3.1 Xích Markov

Xích Markov là một trường hợp đặc biệt của automat hữu hạn có trọng số (weighted finite-state automaton). Ở đây automat hữu hạn có trọng số được định nghĩa là một mở rộng đơn giản của automat hữu hạn, trong đó mỗi cung liên quan đến một xác suất, cho biết khả năng đường đi đó được thực hiện. Xác suất trên tất cả các cung phải có tổng là 1. Với xích Markov chuỗi đầu vào xác định duy nhất trạng thái automat

sẽ đi qua. Xích Markov chỉ hữu ích trong việc gán xác suất cho các chuỗi rõ ràng vì nó không thể biểu diễn các vấn đề mơ hồ [3, 5].



Hình 2.12 Xích Markov biểu diễn chuỗi sự kiện thời tiết



Hình 2.13 Xích Markov biểu diễn xác suất một chuỗi từ

Hình 2.12 là một xích Markov biểu diễn một chuỗi sự kiện thời tiết, các trạng thái gồm có "HOT", "COLD", "RAINY". Hình 2.13 là một biểu diễn đơn giản khác của chuỗi Markov cho việc gán xác suất cho một chuỗi các từ. Một chuỗi Markov được xác định bằng:

$Q = q_1 q_2 \ldots q_N$	Một tập hợp trạng thái
$A = a_0 a_{02} \ldots a_{n1} \ldots a_{nn}$	Một ma trận xác suất chuyển đổi A, mỗi a*ij* đại diện cho một khả năng chuyển đổi từ trạng thái i sang trạng thái j. $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$
Q_0, q_{end}	Một trạng thái bắt đầu và kết thúc đặc biệt không liên quan đến các quan sát

Một xích Markov sử dụng một giả định quan trọng trong thứ tự của xích Markov bậc nhất: xác suất của một trạng thái cụ thể chỉ phụ thuộc vào trạng thái trước đó.

Thuộc tính Markov: $P(q_i|q_1 \dots q_{i-1}) = P(q_i|q_{i-1})$

Bởi vì mỗi a_{ij} biểu diễn một xác suất $p(q_j|q_i)$, luật xác suất yêu cầu giá trị của tất cả cung đi ra từ một trạng thái phải có tổng là 1:

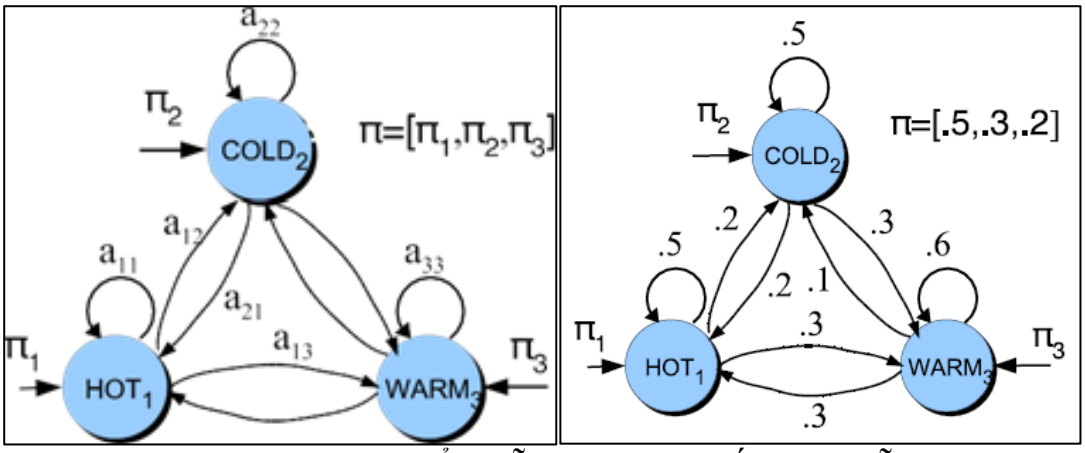
$$\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$$

Một cách biểu diễn thay thế đôi khi được sử dụng cho xích Markov mà không phụ thuộc vào trạng thái bắt đầu hay kết thúc mà thay vào đó nó biểu diễn sự phân bố của các trạng thái bắt đầu và trạng thái được chấp nhận:

$\pi = \pi_1, \pi_2, \dots, \pi_N,$	Phân bố xác suất ban đầu trên các trạng thái π_i là khả năng xích Markov sẽ bắt đầu ở trạng thái i. Một vài trạng thái j có thể có π_j=0, nghĩa là chúng không thể là trạng thái bắt đầu. Cũng theo phân phối xác suất $\sum_{i=1}^{n} \pi_i = 1$
QA = $\{q_x, q_y \dots\}$	Một tập hợp $QA \subset Q$ của những trạng thái được chấp nhận hợp lệ

Vì vậy khả năng trạng thái 1 là trạng thái bắt đầu có thể được biểu diễn như là a_{01} hoặc π_1. Vì mỗi π_i biểu diễn xác suất $p(q_i|START)$, tất cả xác suất π có tổng là 1:

$$\sum_{i=1}^{n} \pi_i = 1$$



Hình 2.14 Xích Markov biểu diễn theo phân bố cho chuỗi sự kiện thời tiết

Một cách biểu diễn thay thế đôi khi được sử dụng cho xích Markov mà không phụ thuộc vào trạng thái bắt đầu hay kết thúc mà thay vào đó nó biểu diễn sự phân bố của các trạng thái bắt đầu và trạng thái được chấp nhận

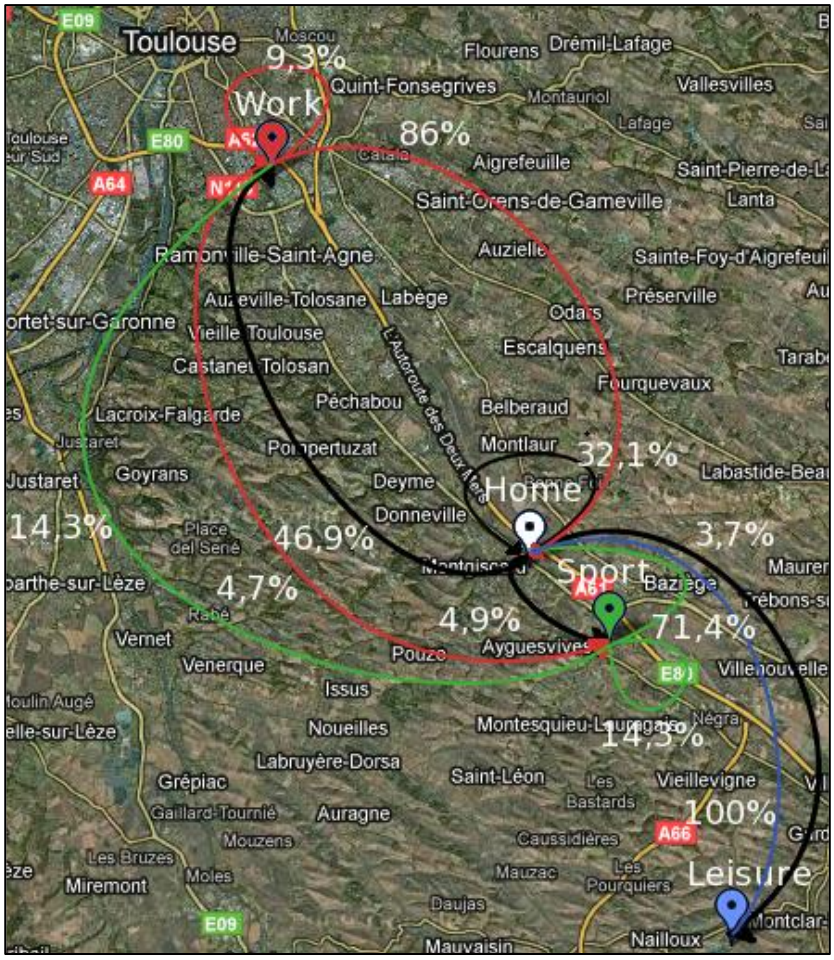
2.3.2 Xích Markov di động (Mobility Markov Chain - MMC)

Xích Markov di động (tên tiếng Anh là Mobility Markov Chain, từ bây giờ sẽ ký hiệu là MMC) mô hình hóa hành vi di chuyển của một người như là một quá trình ngẫu nhiên rời rạc. Trong đó xác suất di chuyển đến một trạng thái (Ở đây là một địa điểm) chỉ phụ thuộc vào trạng thái trước đó (địa điểm trước đó) và phân bố xác suất của quá trình chuyển đổi giữa các trạng thái [11, 12]. Chính xác hơn một MMC bao gồm:

- Một tập hợp trạng thái P = {p_1,…,p_k}, ở đây mỗi trạng thái tương ứng với một địa điểm có tần suất cao (Xếp hạng theo thứ tự giảm dần của tầm quan trọng). Những địa điểm này thường có một ý nghĩa và do đó các nhãn như "Nhà" hoặc "Nơi làm việc" thường có thể được gắn vào chúng. Ngữ nghĩa của một số trạng thái đôi khi có thể được suy ra từ cấu trúc của MMC. Trong luận văn này chúng ta không dán các nhãn ngữ nghĩa cho các địa điểm, thay vào đó chúng ta gán ID của vùng cho các địa điểm này.
- Một tập hợp các chuyển tiếp, như là $t_{i,j}$, đại diện cho việc chuyển từ trạng thái p_i sang trạng thái p_j. Một chuyển đổi từ một trạng thái sang chính nó có thể xảy ra nếu như người đó di chuyển từ một trạng thái sang một địa điểm không thường xuyên rồi quay lại trạng thái đó. Ví dụ một cá nhân có thể rời khỏi nhà mình để đến hiệu thuốc, sau đó trở về nhà (Ở đây hiệu thuốc là địa điểm không thường xuyên nên không được đưa vào các trạng thái)

Một xích Markov di động có thể được biểu diễn như là một đồ thị hoặc một ma trận chuyển dịch. Ở biểu diễn theo đồ thị, những node đại diện cho những trạng thái, còn những mũi tên mô tả các chuyển tiếp giữa những trạng thái. Trong biểu diễn bằng ma trận chuyển dịch, hàng đại diện cho các trạng thái nguồn (địa điểm nguồn) và cột đại diện cho trạng thái đích (địa điểm đích). Giá trị của mỗi ô là xác suất của chuyển tiếp tại hàng và cột.

Xích Markov di động tiêu chuẩn là không nhớ theo nghĩa là dự đoán điểm tương lai chỉ phụ thuộc vào điểm hiện tại. Tuy nhiên, điều này có hạn chế là xích markov di động "quên" điểm đã từng đến trước đó trước khi đến trạng thái hiện tại, điều này có tác động tiêu cực đến độ chính xác của dự đoán. Để giải quyết vấn đề này, S´ebastien Gambs và các cộng sự đề xuất khái niệm n-MMC [12] – là một MMC mà trạng thái tiếp theo không chỉ dựa vào một trạng thái trước đó mà dựa vào chuỗi n trạng thái trước đó.



Hình 2.15 Ví dụ của n-MMC với n = 1

2.3.3 Sử dụng n-MMC để dự đoán điểm đến tiếp theo

Trong [12] tác giả bài báo sử dụng thuật toán DJ-Cluster để tìm ra những địa điểm mà một người hay đến, tuy nhiên, trong khuôn khổ luận văn này, chúng ta sử dụng các ô (vùng) đã chia ở bước trước như là những địa điểm trong việc dự đoán điểm đến tiếp theo.

Để dự đoán điểm đến tiếp theo dựa trên n vị trí cuối cùng, ta sử dụng ma trận chuyển dịch có thay đổi, mà trong ma trận này hàng đại diện cho n điểm đến cuối cùng – thay đổi so với ma trận chuyển dịch ở nguyên bản là hàng đại diện địa điểm cuối, cột đại diện cho điểm đích. Để minh họa việc dự đoán điểm đến tiếp theo, ở đây sử dụng bảng 1 và hình 2.16 lần lượt cho ma trận chuyển dịch và biểu đồ của 2-MMC. 2-MMC bao gồm 4 trạng thái khác nhau: "Home"(H), "Work"(W) "Leisure"(L) và "Other"(O). Mục tiêu là đoán điểm đến tiếp theo dựa trên 2 điểm phía trước (ở đây n = 2). Vì vậy, hàng của ma trận chuyển dịch ký hiệu tất cả những cặp có thể kết hợp của các địa điểm (HH, HW, HO, WH, WW, WO, OH, OW, OO,…) trong khi đó một cột đại diện cho địa điểm tiếp theo trong n-MMC. Ví dụ, nếu như địa điểm lúc trước là H và địa điểm hiện giờ là W, dự đoán địa điểm tiếp theo sẽ là Home (H) và sự chuyển dịch sẽ chuyển từ trạng thái HW sang WH, bởi vì chúng ta cập nhật vị trí trước đó cho W và vị trí hiện thời cho H.

Source/Dest	H	W	L	O
H W	1,00	0,00	0,00	0,00
H L	1,00	0,00	0,00	0,00
H O	0,64	0,34	0,00	0,00
W H	0,00	0,84	0,08	0,08
L H	0,00	0,50	0,00	0,50
O H	0,00	1,00	0,00	0,00
O W	1,00	0,00	0,00	0,00

Bảng 2.1 Ma trận chuyển dịch

Thuật toán *4: Tạo một n-MMC*

Input: *D: tập hợp các bản ghi dữ liệu di chuyển*
n: số lượng các địa điểm lúc trước
MinPts: số lượng nhỏ nhất của bản ghi dữ liệu trong một cụm
ε: bán kính lớn nhất của một cụm
d_{mer}: khoảng cách các cụm có thể hợp nhất

Output: *n-MMC* được tính

/*Tìm ra danh sách các cụm*/

01: Tiền xử lý dữ liệu trong tập *D* bằng cách xóa những bản ghi dữ liệu có sự di chuyển và dữ liệu dư thừa, được tập *D'*

02: Chạy thuật toán phân cụm trên *D'* để tìm ra những cụm tiêu biểu

03: Hợp nhất các cụm có cùng một điểm chung

04: Hợp nhất các cụm nằm trong khoảng cách d_{mer}

05: Thu được *danh sách POIs* là danh sách của tất cả các cụm được khởi tạo

/*Gán nhãn cho các địa điểm, khởi tạo nhãn của cột và hàng trong ma trận chuyển dịch*/

06: **for each** *cụm C* in *danh sách POIs* **do**

07: Tính *khoảng thời gian*, *bán kính* và *mật độ* của *C*

08: **end for**

09: Sắp xếp các cụm trong *danh sách POIs* theo giảm dần về mật độ

10: **for each** *cụm* C_i trong *danh sách POIs* **do**

11: Tạo các trạng thái p_i tương ứng trong xích Markov di động

12: **end for**

13: **for each** *bản ghi dữ liệu m* trong *D'* **do**

14: **if** khoảng cách giữa *bản ghi dữ liệu m* và tâm *cụm* C_i nhỏ hơn *bán kính*$_i$ **then**

15: Cập nhật *n -1* địa điểm lúc trước (lần lượt theo thời gian) và địa điểm hiện tại theo *Ci*

15: Dán nhãn *bản ghi dữ liệu m* với *n – 1* địa điểm trước đó và C_i

16: **else**

17: Dán nhãn *bản ghi dữ liệu m* với giá trị "unknown"

18: **end if**

19: **end for**

/*Tính xác suất chuyển dịch*/

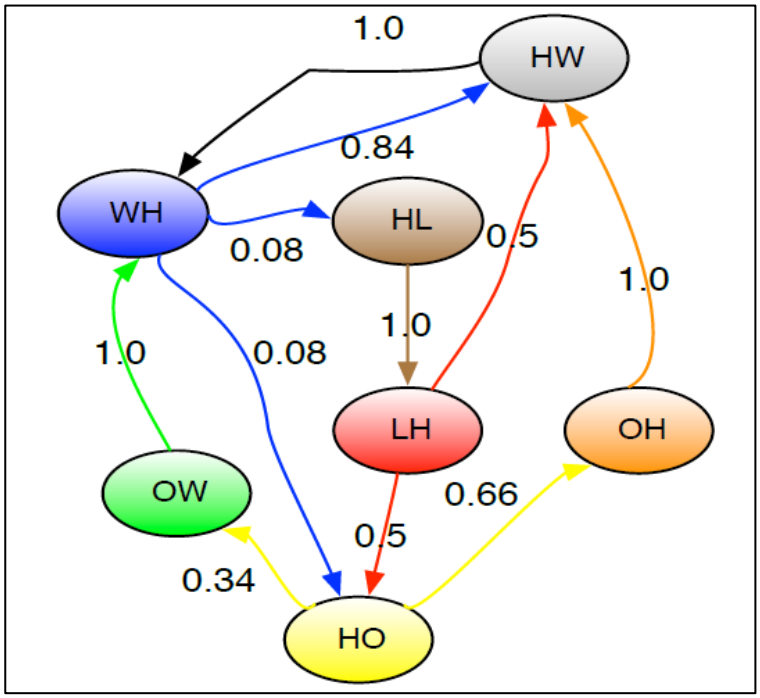
20: Xóa tất cả bản ghi dữ liệu gán nhãn "unknown"

21: Đưa tất cả bản ghi dữ liệu có cùng nhãn về một thể hiện duy nhất

22: Tính tất cả khả năng chuyển dịch giữa mỗi cặp trạng thái trong xích Markov

23: **return** *n-MMC* đã được tính

Thuật toán dự đoán điểm đến tiếp theo đòi hỏi n điểm đã đến trước đó và n-MMC làm đầu vào. Thuật toán làm việc theo cách đơn giản sau đây: Ví dụ đầu vào là ma trận chuyển dịch ở bảng 1 và hai địa điểm trước đó là HO. Thuật toán tìm hàng tương ứng với n địa điểm trước đó và tìm khả năng chuyển dịch lớn nhất có thể xảy ra. Trong ví dụ trên, vì những địa điểm trước đó là HO, địa điểm được dự đoán là H với khả năng là 64%.



Hình 2.16 Đồ thị biểu diễn 2-MMC

Thuật toán 5: Dự đoán sử dụng *n-MMC*
Input: *n-MMC*: Xích Markov di động cho n vị trí quá khứ *M*: Ma trận chuyển dịch của *n-MMC* *n địa điểm lúc trước* **Output**: *Điểm đến tiếp theo được dự báo*
01: Tìm *hàng r* trong *ma trận M* tương ứng với *n* địa điểm đến trước đó 02: Tìm cột tương ứng với khả năng lớn nhất của chuyển dịch p_{max} cho dòng r 03: **return** địa điểm tương ứng với cột với p_{max}

Kết luận: Chương 2 của luận văn trình bày các phương pháp, kỹ thuật được nghiên cứu và áp dụng cho bài toán phân tích, mô phỏng tình trạng giao thông trong luận văn gồm: Thuật toán phân cụm TRACLUS, cách mô phỏng tình trạng giao thông dựa trên thuật toán PageRank sử dụng quá trình di chuyển của taxi để xếp hạng, dự đoán điểm đến tiếp theo sử dụng xích Markov di động n. Chương này cũng chỉ rõ cách sử dụng các kỹ thuật, phương pháp trên để giải quyết các bài toán cụ thể được đặt ra trong luận văn.

Chương 3: Xây dựng hệ thống phân tích, mô phỏng tình trạng giao thông

Với cơ sở dữ liệu được cung cấp là nguồn thu thập từ thiết bị giám sát hành trình gắn trên xe taxi và từ ứng dụng gọi xe taxi, ta tiến hành xây dựng hệ thống qua các bước tổng quan như sau:

- B1: Chia dữ liệu ra thành các tập bản ghi theo ngày (mỗi ngày là một tập bản ghi), chia phân biệt ngày thường và ngày cuối tuần.
- B2: Tiến hành chạy thuật toán phân cụm trên từng tập bản ghi theo ngày ta được các cụm của cung đường di chuyển theo ngày (1), tiến hành chạy thuật toán phân cụm trên từng khung thời gian ta được các cụm cung đường di chuyển theo khung thời gian (2).
- B3: Chia vùng bản đồ Hà Nội thành các ô (vùng) ta được tọa độ, giới hạn của các ô (vùng) (3).
- B4: Dựa trên tọa độ của các ô (vùng) (3) và các cụm cung đường di chuyển theo khung thời gian, biểu diễn luồng di chuyển của các phương tiện vận tải theo thời gian.
- B5: Dựa vào thuật toán PageRank, với các cách tính điểm ban đầu dựa vào: Số lượng xe; số lượng khách lên xe, xuống xe; vận tốc; ta tính các xếp hạng khác nhau cho các vùng dựa vào PageRank, thu được xếp hạng của các ô (vùng) (4).
- B6: Dựa trên vùng và mật độ của vùng hiện tại/ vùng và xếp hạng của vùng hiện tại cùng với mô hình n-MMC [12], chọn các điểm đến tiếp theo là các vùng lân cận, ta xác định vùng đến tiếp theo, được vùng có thể lựa chọn và vùng có xác suất đến nhiều nhất thời điểm tiếp theo (5).
- B7: Dựa trên (5) đưa ra các lựa chọn tốt nhất cho tài xế, dựa trên (1) gợi ý cho tài xế cách di chuyển theo các cung đường khác nhau dựa trên kết nối giữa các vùng

3.1 Các đề xuất

3.1.1 Đề xuất phân vùng bản đồ Hà Nội

Để khái quát hóa các dữ liệu vận tải trong một khu vực, ta tiến hành chia bản đồ Hà Nội thành các ô (vùng), số ô này có thể được cài đặt theo các thông số:

- - Kinh độ, vĩ độ của điểm phía trên góc trái (điểm bắt đầu)

- Chiều dài, chiều rộng của mỗi ô
- Số lượng các ô theo chiều ngang
- Số lượng các ô theo chiều dọc

3.1.2 Cách tính xếp hạng cho PageRank có trọng số

Dựa trên kết quả nghiên cứu của Bin Jiang và các cộng sự [4] ta thấy rằng: dữ liệu giao thông và di chuyển phù hợp với mô hình PageRank có trọng số do đặc tính của giao thông là các khu vực gần khu vực phát triển, giao thông thuận lợi có xu hướng phát triển (tương tự với tắc đường) nên ta chọn mô hình PageRank có trọng số để biểu diễn dữ liệu giao thông và tính xếp hạng cho các vùng

Dựa trên mô hình PageRank có trọng số [14] ta thực hiện thuật toán PageRank có trọng số cho các mục đích khác nhau với các in-link, out-link là các luồng di chuyển của taxi:

- Số lượng xe: Ta lấy giá trị khởi tạo là số xe trong mỗi vùng khi bắt đầu chạy thuật toán
- Số lượng khách lên xe, xuống xe: Lấy giá trị khởi tạo là số khách lên xe; xuống xe
- Vận tốc: Lấy giá trị khởi tạo là vận tốc trung bình toàn ngày chia cho vận tốc trung bình của vùng, phần này cần xử lý để tránh các vùng có vận tốc trung bình là 0

3.1.3 Sử dụng mô hình n-MMC với các nhãn về xếp hạng

Dựa trên kết quả nghiên cứu của S´ebastien Gambs và các cộng sự [11, 12] và đặc tính của dữ liệu giao thông, ta nhận thấy:

- Các luồng di chuyển giao thông là có quy luật, dựa vào địa điểm lúc trước của một người (một nhóm người) ta có thể dự đoán được điểm tiếp theo
- Dữ liệu giao thông có tính lan truyền (một vùng tắc đường có thể khiến các vùng tiếp theo của luồng di chuyển bị tắc)

Ta tiến hành gán nhãn các địa điểm của một người (một nhóm người) dựa trên cả vận tốc di chuyển (tắc – thấp – trung bình - cao) hoặc xếp hạng của địa điểm (vùng) đó (thấp – trung bình – cao), cụ thể từ Bảng 2.1 ta tạo thành Bảng chi tiết hơn như sau:

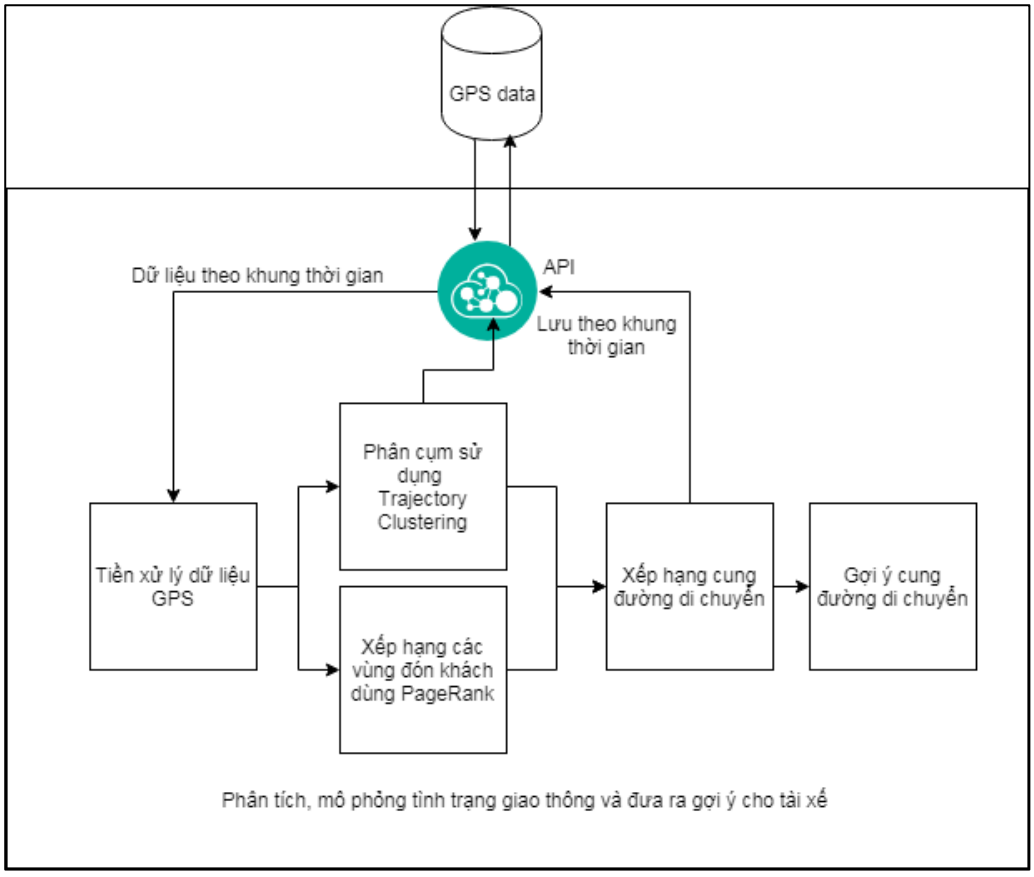
Source/Dest	H thấp	W cao	L thấp	O thấp
H thấp W thấp	1,00	0,00	0,00	0,00
H cao L thấp	1,00	0,00	0,00	0,00
H trung bình O tắc	0,64	0,34	0,00	0,00
W cao H cao	0,00	0,84	0,08	0,08
L trung bình H trung bình	0,00	0,50	0,00	0,50
O cao H thấp	0,00	1,00	0,00	0,00
O thấp W cao	1,00	0,00	0,00	0,00

Bảng 3.1 Bảng ma trận chuyển dịch có thêm nhãn về tốc độ di chuyển

Từ cơ sở các địa điểm đích, ta tính điểm cho mỗi lựa chọn và đưa ra lời khuyên cho tài xế.

3.2 Tổng quan hệ thống

Hệ thống được thiết kế như sau



Hình 3.1 Hệ thống mô phỏng và đưa ra gợi ý giao thông

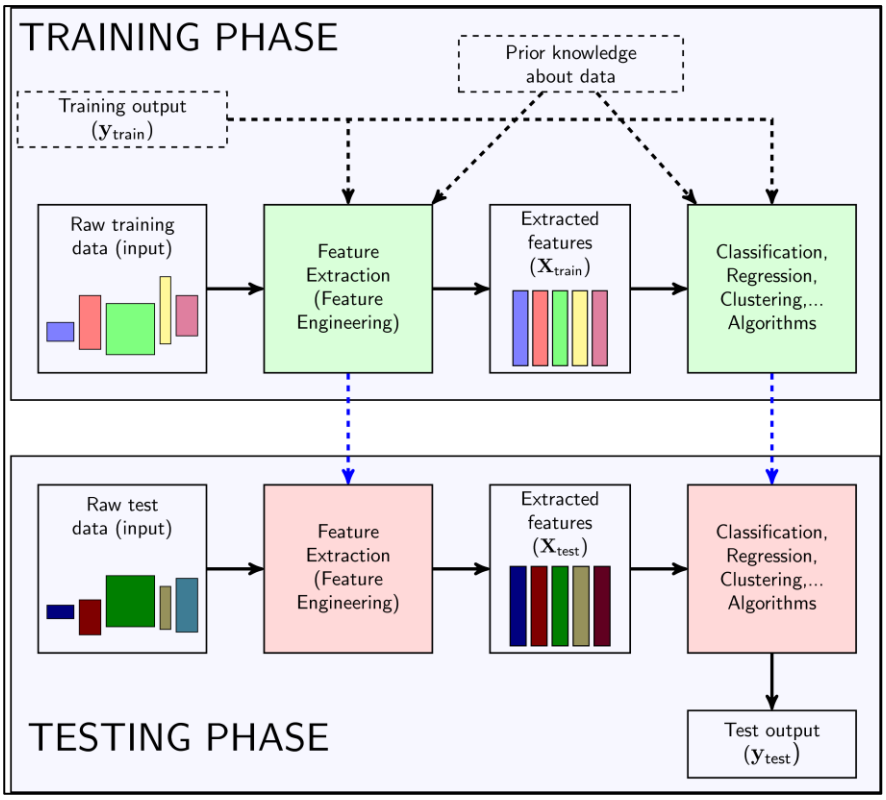
Với các thành phần:

- **GPS data:** Cơ sở dữ liệu của hệ thống, ở hệ thống trong luận văn cơ sở dữ liệu này lưu trữ:

  - Dữ liệu về các bản tin GPS của từng phương tiện (mỗi phương tiện phân biệt bằng id của phương tiện)
  - Dữ liệu về các cung di chuyển đã phân cụm bằng thuật toán TraClus
  - Dữ liệu về ma trận chuyển dịch qua tập huấn
- **Tiền xử lý dữ liệu GPS:** Module xử lý các dữ liệu nhiễu (kinh độ, vĩ độ, vận tốc không hợp lý)
- **Phân cụm sử dụng TrajectoryClustering:** Module phân cụm sử dụng thuật toán TrajectoryClustering và lưu trữ dữ liệu đã phân cụm
- **Xếp hạng các vùng đón khách bằng PageRank:** Module sử dụng thuật toán PageRank để xếp hạng các vùng theo các tiêu chí khác nhau
- **Xếp hạng và gợi ý cung đường di chuyển:** Hai module sử dụng mô hình n-MMC để tập huấn và gợi ý các cung đường di chuyển dựa trên dự đoán về luồng di chuyển, vận tốc

Phần lựa chọn các công nghệ để xây dựng hệ thống sẽ được trình bày trong mục 4.2

Mô hình xử lý dữ liệu cho phần dự đoán được mô tả như hình 3.2:



Hình 3.2 Mô hình chung cho các bài toán dự đoán

Có hai pha lớn là Training phase và Testing phase. Với bài toán dự đoán điểm đến tiếp theo sử dụng n-MMC, chúng ta có cặp dữ liệu (input, output)

Training Phase:

- **raw training input.** Raw input là tất cả các thông tin ta biết về dữ liệu. Với bài toán trong luận văn thì chính là thông tin về dữ liệu GPS của phương tiện vận tải
- **(optional) output của training set.** Trong luận văn phần này chính là ma trận xác suất chuyển dịch dự đoán phương tiện đến tiếp theo với thông tin về vận tốc trung bình trong vùng đó
- **(optional) Prior knowledge about data:** Là giả thiết về dữ liệu đang có, ở đây luận văn đưa ra giả thiết vận tốc trung bình của phương tiện ở trong vùng là trung bình cộng vận tốc của các bản ghi

Main Algorithms: Luận văn sử dụng thuật toán và mô hình n-MMC

Testing Phase: Với raw input mới, luận văn sử dụng dữ liệu thu được từ Training phase qua main algorithms để dự đoán output.

Kết luận: Chương 3 của luận văn trình bày mô hình về hệ thống bài toán phân tích và mô phỏng tình trạng giao thông dựa vào khai phá dữ liệu vận tải và mô hình tập huấn, đánh giá dữ liệu cho bài toán dự đoán – gợi ý di chuyển cho phương tiện vận tải. Chương này cũng đưa ra quy trình thực hiện giải quyết các bài toán trong luận văn, các đề xuất để kết nối, bổ sung cho các kỹ thuật, giải pháp trình bày trong chương 2 nhằm giải quyết các bài toán trong luận văn nêu ra ở chương 1

Chương 4: Thử nghiệm và đánh giá

4.1 Tổng quan về dữ liệu sử dụng trong đề tài

4.1.1 Định dạng dữ liệu

Dữ liệu định vị của phương tiện vận tải được thiết bị định vị ghi lại và gửi về máy chủ theo một khoảng thời gian cố định. Nếu một phương tiện bật máy (ở trạng thái bật chìa khóa điện), dữ liệu sẽ được gửi lên 15 giây một lần, ngược lại, ở trạng thái tắt máy, dữ liệu sẽ được gửi 30 giây một lần.

Như đã trình bày trong chương 1, dữ liệu định vị tuy có cách biểu diễn khác nhau với những thiết bị khác nhau, tuy nhiên dữ liệu cơ bản nhất của phương tiện vận tải gồm những thông tin như sau:

- Thời gian (tính bằng giây)
- Kinh độ
- Vĩ độ
- Vận tốc (do thiết bị thu nhận từng giây, có thể được tính tương đối từ 4 thông tin đầu)
- Hướng di chuyển (do thiết bị thu nhận từng giây, có thể được tính tương đối từ 4 thông tin đầu)
- Trạng thái (do thiết bị thu nhận từng giây, do các dây cảm biến trên thiết bị hành trình gắn với những thành phần cụ thể trên xe)

Dữ liệu sử dụng trong luận văn là dữ liệu từ các nguồn như sau:

Dữ liệu thiết bị giám sát hành trình của Công ty TNHH Phát triển Công nghệ Điện tử Bình Anh với phương tiện là xe taxi và dữ liệu từ ứng dụng đặt xe, điều phối taxi do chính tác giả luận văn xây dựng.

4.1.2 Dữ liệu từ thiết bị giám sát hành trình

Dữ liệu đầu vào từ thiết bị giám sát hành trình của công ty trách nhiệm hữu hạn Phát triển Công nghệ Điện tử Bình Anh được lưu trong file text, với định dạng như sau:

Đường dẫn đến file text: \<năm: 4 chữ số>\<tháng: 2 chữ số>\<ngày: 2 chữ số>

1 dòng dữ liệu có những thông tin như sau (cách nhau bởi dấu phẩy):

@00:00:17,105.862778,20.992922,0,0,131112,0,0,km(0),vbgt(0),mt()

Trong đó:

@: bắt đầu dòng tin

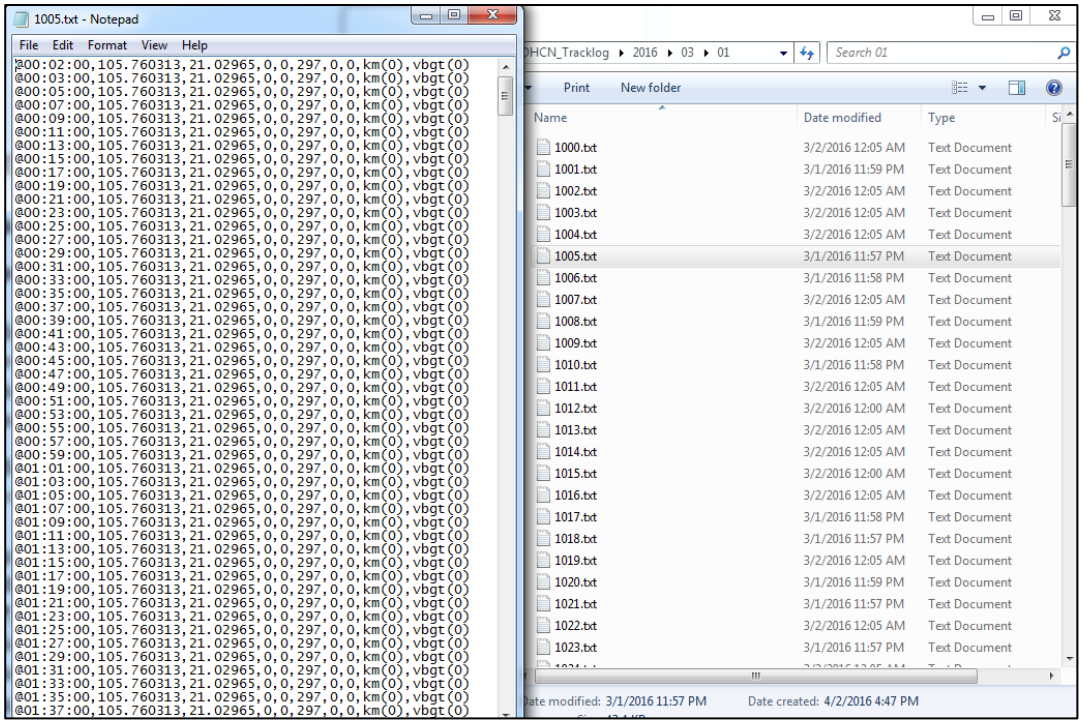
00:00:17: thời gian trong ngày: giờ: phút: giây

105.862778: Longitude: kinh độ

20.992922: Latitude: vĩ độ

Số 0 ở vị trí thứ 6 là status, sẽ thể hiện trạng thái có khách hay không như sau:

- CÓ KHÁCH = Status & 3 > 0 (phép AND bit)
- KHÔNG KHÁCH = Status & 3 = 0 (phép AND bit)



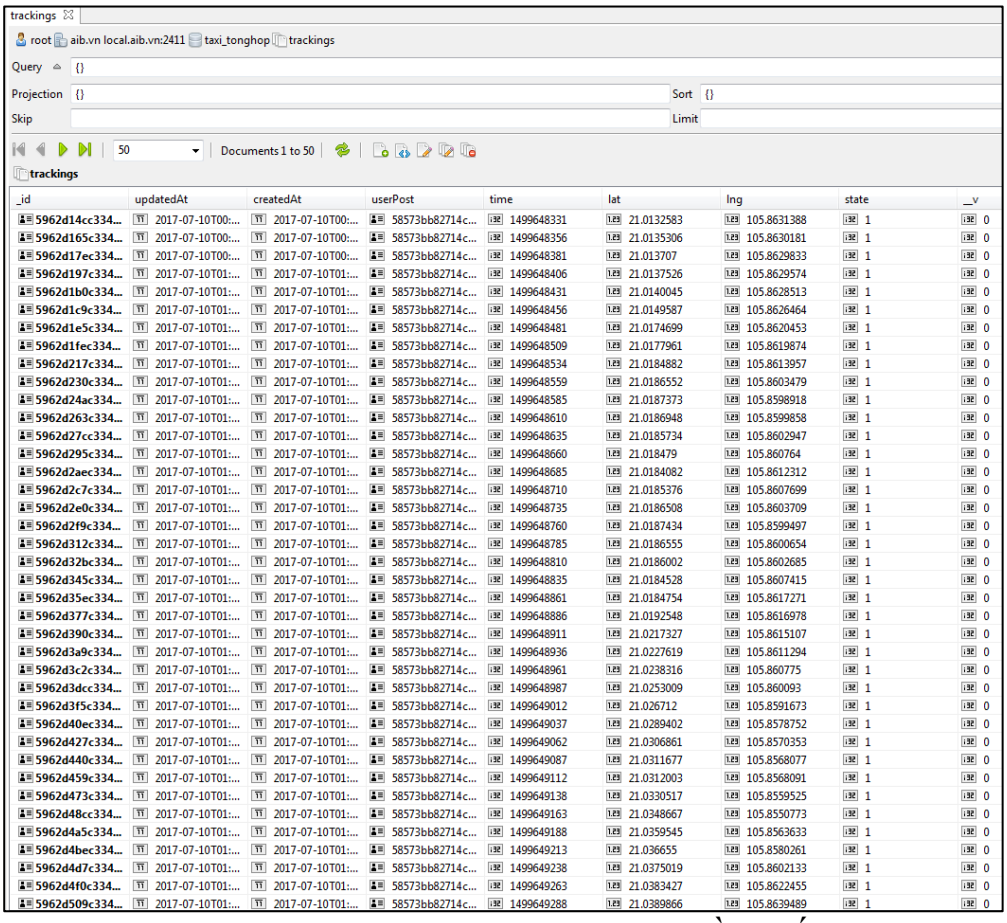
Hình 4.1Dữ liệu gps từ thiết bị giám sát hành trình của công ty Bình Anh

Dữ liệu tư thiết bị giám sát hành trình của công ty TNHH Phát triển Công nghệ Điện tử Bình Anh gồm 30 ngày, với số xe là 100 xe, tổng dung lượng là 1.33 GB.

4.1.3 Dữ liệu từ ứng dụng đặt taxi, điều phối taxi

Dữ liệu đầu vào từ ứng dụng đặt taxi, điều phối taxi được lưu trong CSDL MongoDB và định dạng như sau:

```
{
    "userPost": "58573bb02714c9029a615c5c",
    "time": 1487138188,
    "lat": 21.0056755,
    "lng": 105.8010069,
    "state": 1,
}
```



Hình 4.2 Dữ liệu từ ứng dụng điều phối taxi

Dữ liệu từ ứng dụng đặt xe taxi gồm 23 triệu bản ghi, chiếm dung lượng 3GB, tuy nhiên dữ liệu từ ứng dụng khá rời rạc và nhiều nhiễu

4.1.4 Dữ liệu xử lý trong hệ thống

Sau khi tiền xử lý dữ liệu từ các nguồn dữ liệu, chúng ta thu được dữ liệu đầu vào để chạy thuật toán phân cụm như sau: dữ liệu có 4 cột lần lượt là: vĩ độ (Y), kinh độ (X), ID (ID tương ứng với mỗi taxi), trạng thái khách hàng gồm có 3 trạng thái: 1- không có khách, 2 - trên đường đón khách, và 3- có khách

Vĩ độ (Y)	Kinh độ (X)	ID	Trạng thái khách hàng
21.0300596	105.7889164	0	3
21.0300596	105.7889164	0	3
21.0301935	105.7859652	0	3
21.0301178	105.7896338	0	1
21.0287439	105.7889675	0	1
21.0296401	105.7913306	0	1
21.0671696	105.8348092	0	1
21.0671696	105.8348092	1	1

Bảng 4.1 Dữ liệu đầu vào cho thuật toán phân cụm

Đầu ra sau khi phân cụm trong thuật toán TRACLUS . Dữ liệu sẽ gồm điểm xuất phát (vĩ độ, kinh độ), điểm đích (vĩ độ, kinh độ), ID và ID cụm.

Điểm xuất phát		Điểm đích		ID	ID cụm
Vĩ độ (Y)	Kinh độ (X)	Vĩ độ (Y)	Kinh độ (X)		
21.04617	105.790172	21.046049	105.781655	0	2
21.038296	105.791987	21.032248	105.790092	0	1
21.030695	105.784669	21.030111	105.788194	0	5
21.030111	105.788194	21.03984	105.790379	0	1

Bảng 4.2 Dữ liệu sau khi phân cụm

4.2 Lựa chọn công nghệ

Để xây dựng API phân tích và lấy dữ liệu online, ta sử dụng ngôn ngữ Nodejs để viết api cho truy vấn dữ liệu, python để phân tích, mô hình hoá dữ liệu, và cơ sở dữ liệu là Mongodb

4.2.1 Ngôn ngữ Nodejs

Node.js là một phần mềm mã nguồn mở được viết dựa trên ngôn ngữ JavaScript cho phép lập trình viên có thể xây dựng các ứng dụng chạy trên máy chủ. Ban đầu, Node.js được phát triển bởi Ryan Dahl. Phiên bản đầu tiên của Node.js được cho ra mắt vào năm 2009.

Node.js có thể chạy được trên nhiều nền tảng khác nhau như Windows, Linux hay Mac OS. Node.js được phát triển sử dụng V8 Engine là bộ thư viện JavaScript được Google phát triển để viết trình duyệt web Chrome.

Bản thân Node.js không phải là một ngôn ngữ lập trình mới, thay vào đó Node.js là một nền tảng mã nguồn mở (hay phần mềm mã nguồn mở) được viết dựa trên ngôn ngữ JavaScript.

Node.js có thể được dùng để tạo các ứng dụng chạy trên môi trường máy chủ như các ứng dụng web. Tuy nhiên Node.js không chỉ giới hạn ở việc tạo các website mà nó còn có thể được dùng để phát triển các công cụ chạy trên máy tính cá nhân.

Trong khi JavaScript thường được dùng trên trình duyệt thì Node.js lại được sử dụng để phát triển ứng dụng chạy trên máy chủ server. Node.js cũng có thể được chạy như một ứng dụng độc lập trên máy tính cá nhân (mà không cần phải thông qua môi trường của trình duyệt). Nói chính xác hơn thì không thể chạy Node.js sử dụng môi trường trình duyệt.

Về bản chất Node.js là một phần mềm mở rộng được phát triển trên nền tảng ngôn ngữ JavaScript. Vì vậy cú pháp của Node.js giống với cú pháp của JavaScript.

Ưu điểm của Node.js

- *JSON APIs:* NodeJS được điều khiển bởi REST/JSON APIs, với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tối ưu cho các dịch vụ Webs làm bằng JSON.
- *Ứng dụng trên 1 trang:* Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh, các ứng dụng sử dụng Node.js sẽ không cần tải lại trang, có thể gồm rất nhiều request từ người dùng cần sự hoạt động nhanh.
- *Shelling tools unix:* NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất.
- *Streamming Data (Luồng dữ liệu):* Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
- *Ứng dụng Web thực:* Node.js có thể sử dụng để xây dựng 1 ứng dụng chat, feed ... Facebook, Twitter.

4.2.2 Ngôn ngữ python

Python là một ngôn ngữ lập trình phổ biến. Được tạo ra bởi Guido van Rossum vào năm 1991.

Ngày nay, Python được sử dụng trong nhiều mục đích, trong luận văn ngôn ngữ python được sử dụng với mục đích phục vụ các tính toán khoa học

Hiện nay, với khả năng xử lý các phép toán phức tạp của mình, Python đang được sử dụng nhiều trong việc phát triển Trí Tuệ Nhân Tạo và các nghiên cứu trong lĩnh vực Machine Learning.

4.2.3 Cơ sở dữ liệu Mongo

MongoDB (bắt nguồn từ “humongous”) là một hệ cơ sở dữ liệu NoSQL mã nguồn mở.

Thay cho việc lưu trữ dữ liệu vào các bảng có quan hệ với nhau như truyền thống, MongoDB lưu các dữ liệu cấu trúc dưới dạng giống với JSON(JavaScript Object Notation) và gọi tên là BSON. Dự án được bắt đầu triển khai vào tháng 10 năm 2007 bởi 10gen trong khi công ty này đang xây dựng một nền tảng như là dịch vụ (Platform as a Service) giống như Google App Engine. Phải đến năm 2009, dự án này được tách độc lập. Hệ thống có thể chạy trên Windows, Linux, OS X và Solaris. Nó được một số tổ chức sử dụng trong thực tế như:

- Caigslist : Công ty làm việc trong lịch vực môi giới quảng cáo trên các website khác (giống adMicro của Việt Nam). MongoDB giúp cho công ty này quản lý hàng tỉ các bản ghi quảng cáo thuận tiện và nhanh chóng.
- Foursquare là một mạng xã hội gắn các thông tin địa lý. Công ty này cần lưu dữ liệu của rất rất nhiều vị trí của các địa điểm như quán cafe, nhà hàng, điểm giải trí, lịch sử, … và ghi lại những nơi mà người sử dụng đã đi qua.
- CERN : Trung tâm nghiên cứu năng lượng nguyên tử của Châu Âu, sử dụng MongoDB để lưu trữ lại các kết quả, dữ liệu thí nghiệm của mình. Đây là một lượng dữ liệu khổng lồ sẽ dùng để sử dụng trong tương lai.
- MTV Networks, Disney Interactive Media Group, bit.ly, The New York Times, The Guardian, SourceForge, Barclays, …

4.2.3.1 Ưu điểm của MongoDB

- Dễ học, có một số nét khá giống với CSDL quan hệ – Quản lý bằng command line hoặc bằng GUI như RockMongo hoặc phpMoAdmin
- Linh động, không cần phải định nghĩa cấu trúc dữ liệu trước khi tiến hành lưu trữ, điểm này rất hữu ích khi ta cần làm việc với các dạng dữ liệu không có cấu trúc.
- Khả năng mở rộng tốt (distributed horizontally), khả năng cân bằng tải cao, tích hợp các công nghệ quản lý dữ liệu vẫn tốt khi kích thước và thông lượng trao đổi dữ liệu tăng.
- Miễn phí

4.2.3.2 Kiến trúc của MongoDB

Một MongoDB Server sẽ chứa nhiều database. Mỗi database lại chứa một hoặc nhiều colection. Đây là một tập các documnents, về mặt logic thì chúng gần tương tự như các table trong CSDL quan hệ. Tuy nhiên, điểm hay ở đây là ta không cần phải định nghĩa trước cấu trúc của dữ liệu trước khi thao tác thêm, sửa dữ liệu… Một document là một đơn vị dữ liệu – một bản ghi (không lớn hơn 16MB). Mỗi chúng lại chứa một tập các trước hoặc các cặp key – value. Key là một chuỗi ký tự, dùng để truy xuất giá trị dạng : string, integer, double, … Dưới đây là một ví dụ về MongoDB document

{

_id : ObjectId("4db31fa0ba3aba54146d851a"),

username : "joegunchy",

email : "joe@mysite.org",

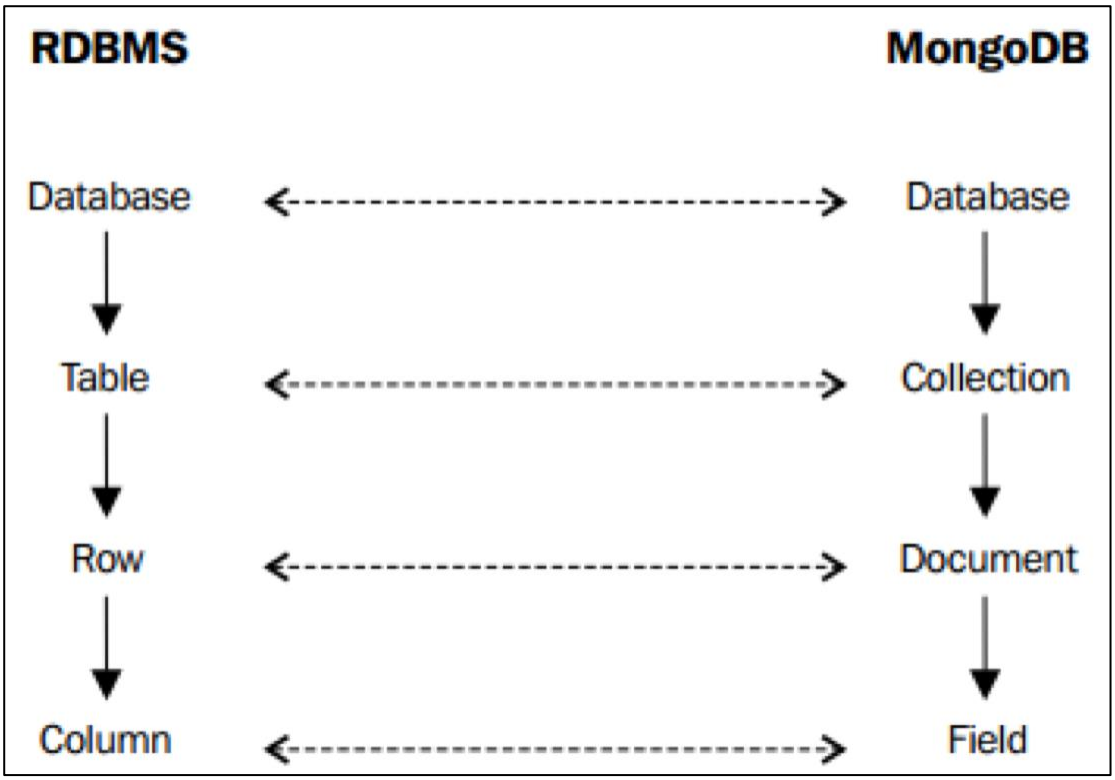
age : 26,

is_admin : true,

created : "Sun Apr 24 2011 01:52:58 GMT+0700 (BDST)"

}

Cấu trúc có vẻ khá giống JSON, tuy nhiên, khi lưu trữ document này ra database, MongoDB sẽ serialize dữ liệu thành một dạng mã hóa nhị phân đặc biệt – BSON. Ưu điểm của BSON là hiệu quả hơn các dạng format trung gian như XML hay JSON cả hệ tiêu thụ bộ nhớ lẫn hiệu năng xử lý. BSON hỗ trợ toàn bộ dạng dữ liệu mà JSON hỗ trợ (string, integer, double, Boolean, array, object, null) và thêm một số dạng dữ liệu đặc biệt như regular expression, object ID, date, binary, code.



Hình 4.3 So sánh giữa RDBMS và MongoDB

4.3 Kết quả thu được

4.3.1 Môi trường thử nghiệm

Các thuật toán và mô hình hệ thống được xây dựng và thử nghiệm trên các máy tính có cấu hình như sau:

Máy server

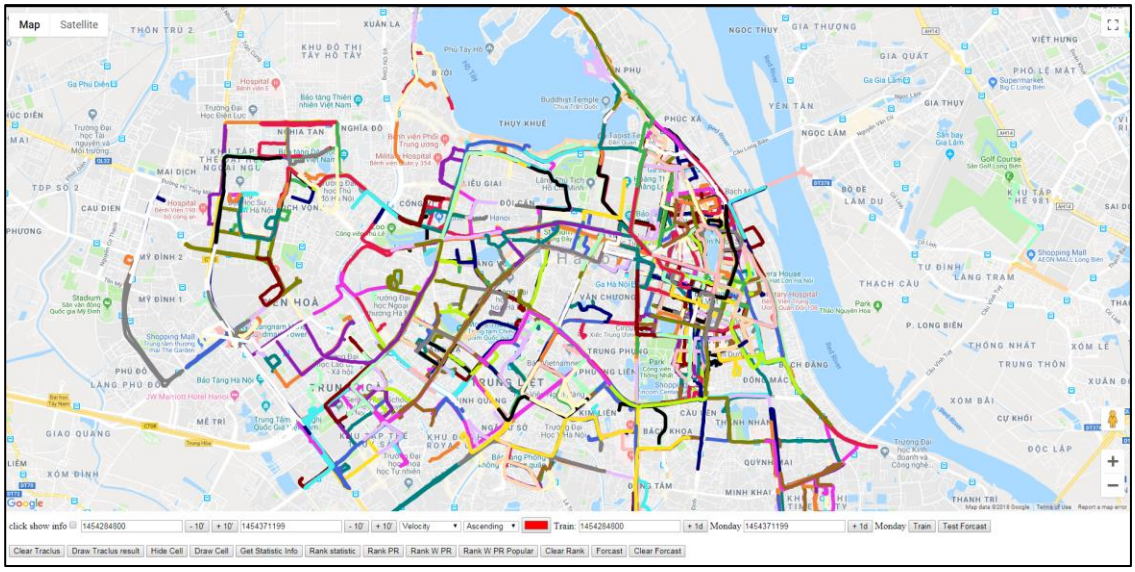
- •CPU: Intel(R) Xeon(R) CPU E3-1230 v5 @ 3.40GHz
- RAM: 8 GB
- GPU: Intel HD Graphic
- Hệ điều hành Centos 7

Máy client

- CPU: Intel® Core ™ i5 CPU M520
- RAM: 8 GB
- GPU: ATI mobility Radeaon HD 5730
- Hệ điều hành Win7 Ultimate

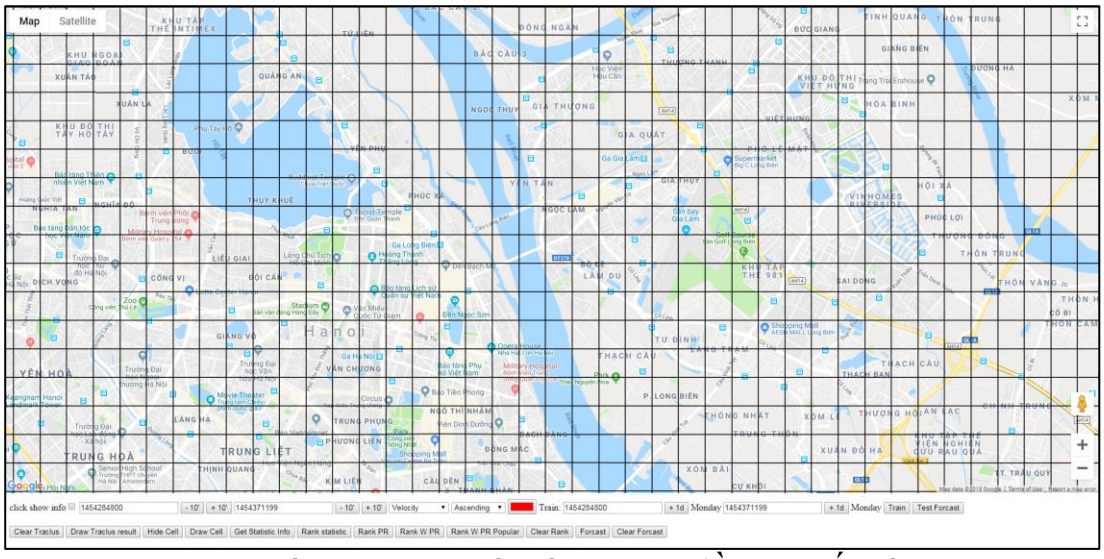
4.3.2 Kết quả thử nghiệm

Các quãng đường mà xe đi qua được phân chia thành các cụm quãng đường nhờ vào thuật toán TRACLUS. Các cụm này sẽ được biểu diễn bởi các màu khác nhau trên Hình 4.4. Chúng ta có thể thấy các quãng đường có chung đặc tính (đặc điểm địa lý) sẽ được gom chung vào cùng một cụm. Mỗi cụm được biểu diễn bằng một màu ngẫu nhiên khác nhau. Một cụm thỏa mãn những yếu tố như sau:Các đoạn đường con chung có sự gần nhau về địa lý, số lượng đường con trong cụm tối thiểu là 3. Với các thông số này cho phép phát hiện hành vi cũng như quy luật di chuyển của taxi.



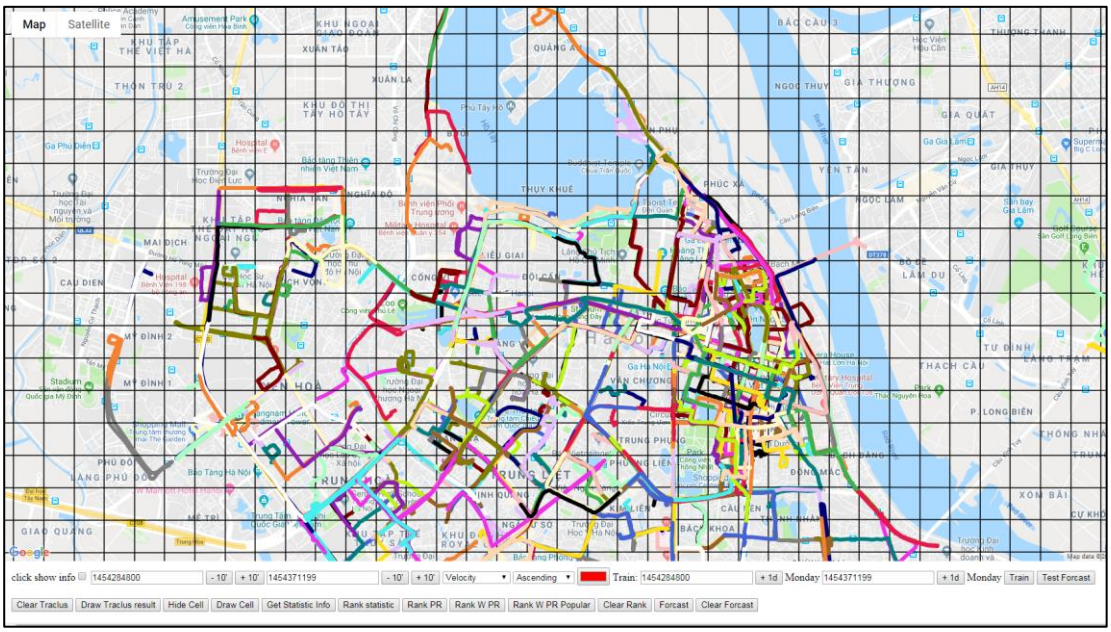
Hình 4.4 Kết quả thuật toán TRACLUS trên dữ liệu mẫu

Để thực hiện đề xuất ở mục 3.1.1 ta tiến hành chia vùng (ô) cho bản đồ Hà Nội, với điểm bắt đầu là: (20.9333, 105.75) và điểm kết thúc là (21.1333, 105.95), các điểm bắt đầu và điểm kết thúc được lấy theo dữ liệu về bản đồ địa chính và theo nhu cầu của bài toán đặt ra, thu được chiều dài gồm 50 vùng (ô), chiều rộng gồm 50 vùng (ô), tổng thể là 2500 vùng (ô), mỗi vùng (ô) có chiều dài và chiều rộng xấp xỉ 500m



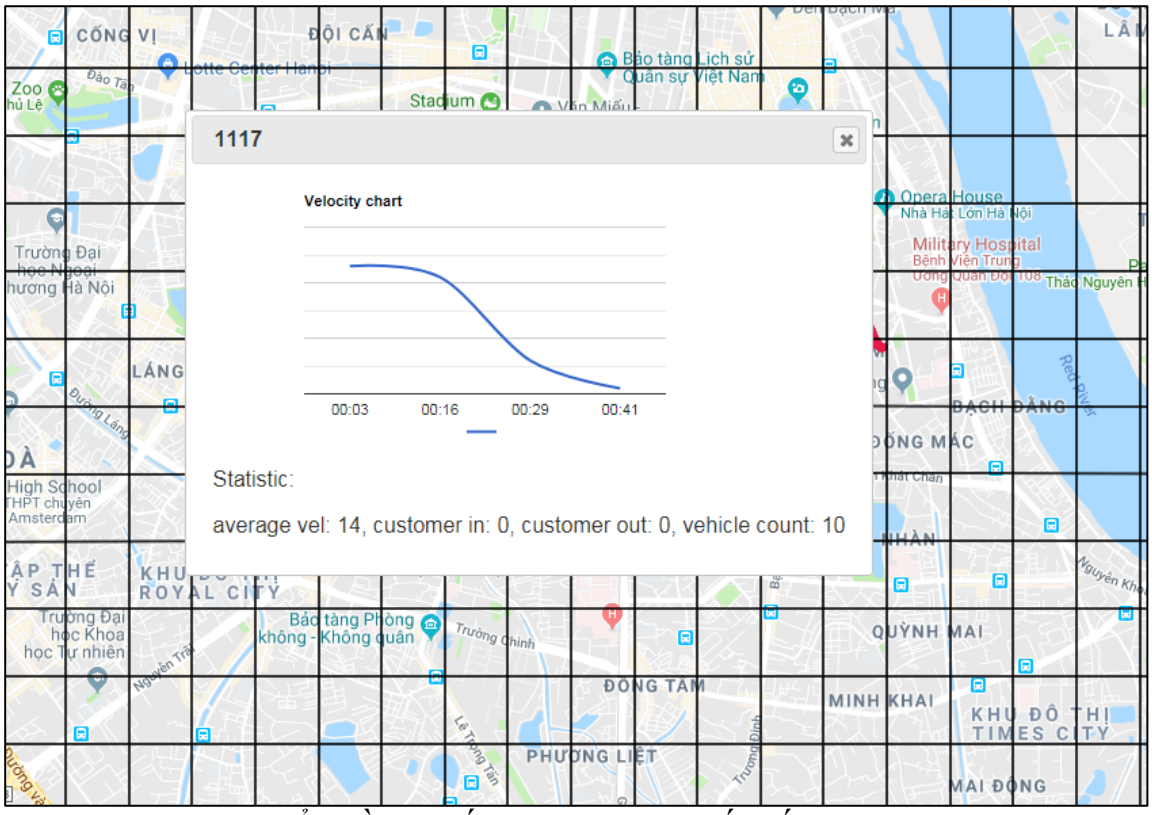
Hình 4.5 Chia ô (vùng) bản đồ theo cấu hình

Cách chia vùng (ô) này có thể kết hợp với biểu diễn dữ liệu phân cụm trong hình 4.4 để cho thấy một số thông tin về các cụm (mật độ, độ quan trọng)



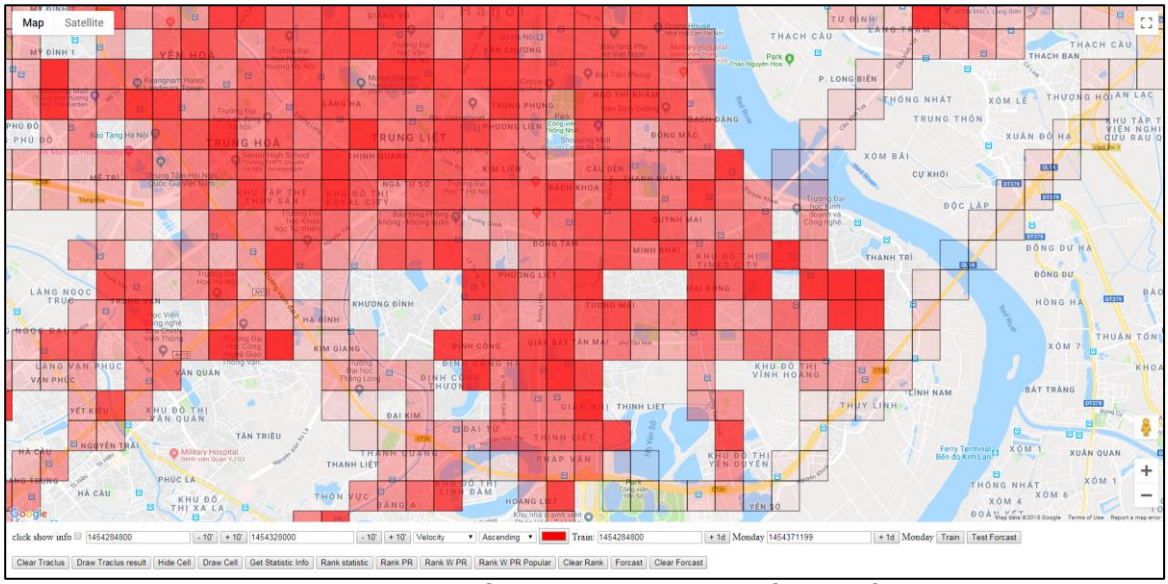
Hình 4.6 Hiển thị các tuyến di chuyển trên bản đồ chia ô (vùng)

Để có thêm thông tin về độ quan trọng của các vùng (ô) luận văn thực hiện thống kê và vẽ biểu đồ về vận tốc trên các vùng (ô) như hình 4.7



Hình 4.7 Biểu đồ vận tốc và các thông số thống kê của một ô (vùng)

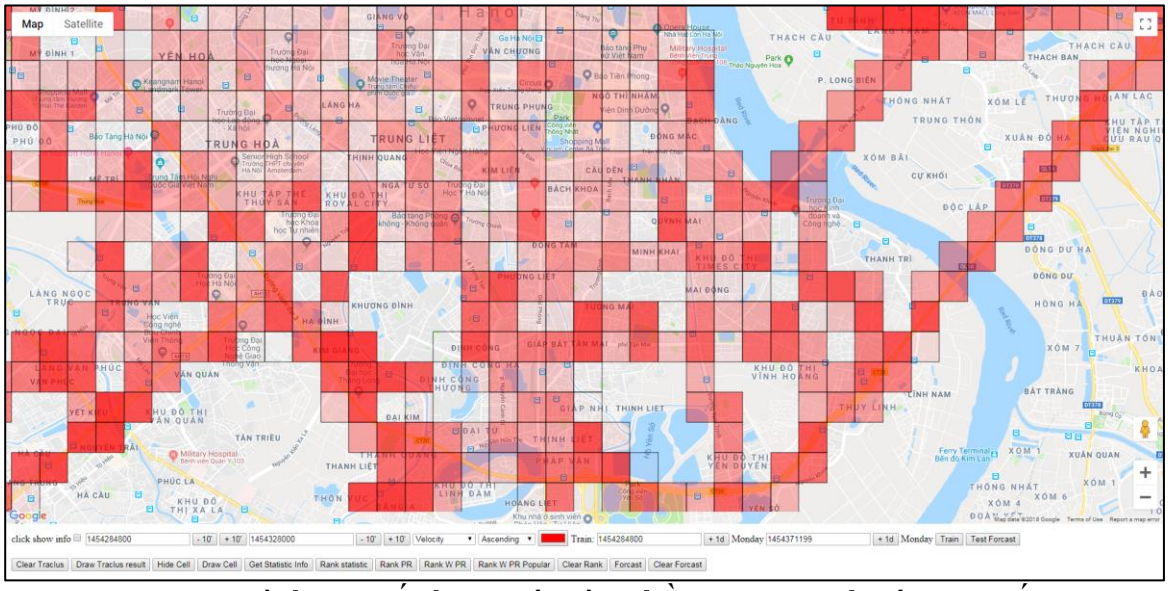
Để có thông tin tổng quan, luận văn tiến hành xếp hạng các vùng (ô) bằng phương pháp thống kê, ở hình 4.8 là xếp hạng các vùng (ô) theo vận tốc, với các màu đỏ đậm hơn là các vùng (ô) có vận tốc di chuyển cao hơn.



Hình 4.8 Xếp hạng các vùng bằng thống kê

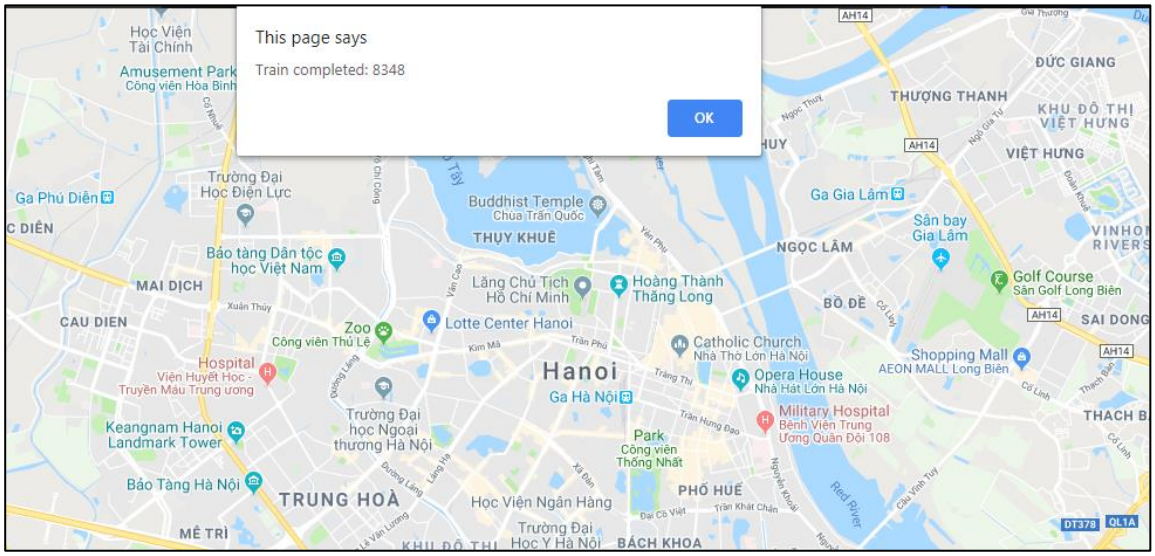
Thực hiện xếp hạng bằng PageRank có trọng số cho vận tốc di chuyển cùng thời gian với hình 4.8, ta nhận thấy các vùng (ô) có màu đỏ đậm trong thuật toán PageRank cho ta các vùng đỏ liền mạch hơn (do tính chất lan truyền) và tập trung

vào các đoạn đường cao tốc, những vùng đỏ đậm liên tiếp có thể gợi ý cho tài xế di chuyển trên cung đường có vận tốc cao (giúp tránh tắc đường, tiết kiệm nhiên liệu)

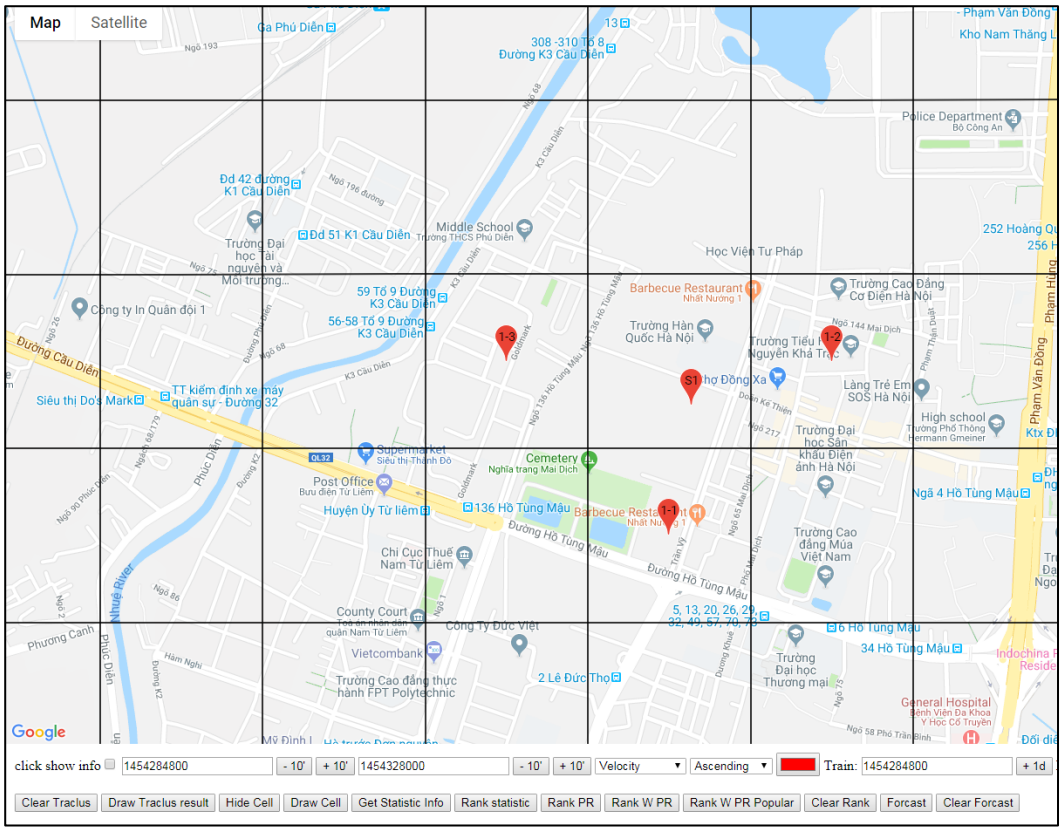


Hình 4.9 Xếp hạng các vùng bằng PageRank có trọng số

Ngoài ra luận văn tiến hành training dữ liệu dựa trên các ngày trong tuần trong hai tuần bằng mô hình n-MMC (kết quả trong hình 4.10) từ đó đưa ra dự đoán cho tài xế tại một thời điểm để lựa chọn cung đường tốt nhất trong hình 4.11



Hình 4.10 Traing tập dữ liệu mẫu theo từng ngày



Hình 4.11 Gợi ý các vùng có thể di chuyển

4.4 Tính chính xác của dữ liệu dự đoán

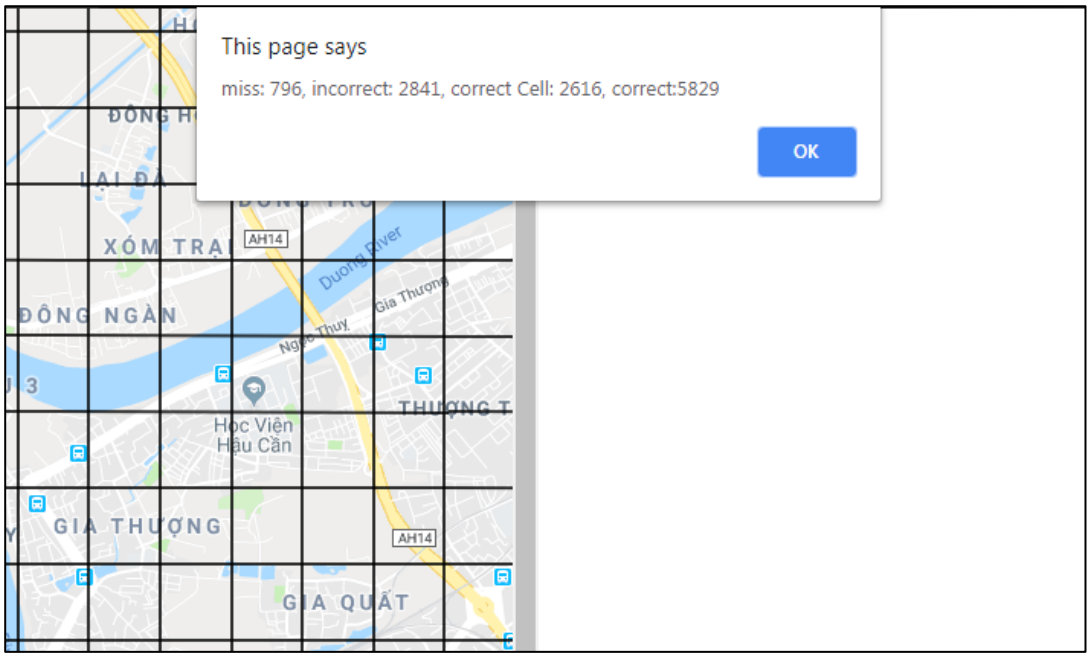
Sử dụng mô hình chung cho các bài toán dự đoán như ở hình 3.2 trên hai nguồn dữ liệu ở mục 4.1 luận văn tiến hành dự đoán điểm đến và mật độ điểm đến tiếp theo dựa trên tập dữ liệu ta thu được kết quả dự đoán chính xác về điểm đến từ 70% - 85% với dữ liệu từ thiết bị giám sát hành trình, và 50 – 73% với dữ liệu từ ứng dụng đặt xe taxi, và chính xác về cả điểm đến và mật độ điểm đến từ 45% - 60% với cả hai bộ dữ liệu.

Với các tham số như trong hình 4.12:

- **Miss:** Các điểm để dự đoán không nằm trong tập dữ liệu huấn luyện
- **Incorrect:** Dự đoán sai cả về nhãn và mật độ của điểm đích
- **Correct cell:** Dự đoán đúng về điểm đến, nhưng sai về mật độ của điểm đích
- **Correct:** Đúng cả về điểm đến và mật độ

Với cách tính như sau:

- Độ chính xác về cả điểm đến và mật độ = correct/tổng
- Độ chính xác về cả điểm đến = (correct + correct cell)/tổng



Hình 4.12 Kiểm tra tính chính xác của dữ liệu dự đoán

Kết luận: Trong chương 4 của luận văn tác giả đã trình bày quá trình thử nghiệm bao gồm: môi trường thử nghiệm, kết quả thử nghiệm. Kết quả thử nghiệm được thực hiện trên hai bộ dữ liệu về taxi từ thiết bị giám sát hành trình và ứng dụng đặt xe taxi, trình bày tổng quan về các kết quả thu được, đưa ra cách đánh giá và đánh giá độ chính xác của mô hình dự báo

KẾT LUẬN

Những vấn đề đã được giải quyết trong luận văn

Luận văn đã tiến hành nghiên cứu giải quyết các bài toán trong Giám sát và điều khiển giao thông. Bài toán này được đánh giá có độ phức tạp cao và có ứng dụng thực tiễn lớn. Phương pháp giải quyết của luận văn tập trung vào phân cụm các cung đường di chuyển, xếp hạng các vùng giao thông, dự đoán lưu lượng và điểm đến, trên cơ sở đó gợi ý cung đường di chuyển cho người tham gia giao thông.

Dựa trên các nghiên cứu đã có, luận văn đề xuất một số cách áp dụng, kết hợp các nghiên cứu để giải các bài toán thực tiễn. Luận văn đã xây dựng mô hình nhằm giải quyết các bài toán đặt ra và thử nghiệm trên máy tính cá nhân.

Luận văn cũng đã tiến hành xây dựng giao diện trực quan để hiển thị kết quả của các bài toán đặt ra. Luận văn được chạy trên hai bộ dữ liệu thực tế từ hai nguồn dữ liệu khác nhau và đã có một số kết quả nhất định.

Định hướng nghiên cứu trong tương lai

Tiến hành khắc phục tình trạng thiếu chính xác do dữ liệu thưa, đặc biệt là dữ liệu từ các ứng dụng di động. Tiến hành xây dựng hệ thống gợi ý theo hướng tiếp cận học tăng cường.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. Nguyễn Văn Tăng (2017) "Phát triển dịch vụ ứng dụng công nghệ GPS trong quản lý, giám sát, điều phối và tối ưu hóa kế hoạch sử dụng phương tiện", Bộ công thương - Chương trình quốc gia phát triển công nghệ cao đến năm 2020

[2]. Viện Khoa học và Công nghệ Giao thông (2016) "Dự thảo về tiêu chuẩn quốc gia cho kiến trúc hệ thống giao thông thông minh its", Bộ Khoa học và Công nghệ

Tiếng Anh

[3]. A. A. Markov (2006) "Classical Text in Translation An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains", Science in Context 19(4), pp. 591–600

[4]. Bin Jiang (2008) "Ranking Spaces for Predicting Human Movement in an Urban Environment", Journal International Journal of Geographical Information Science Volume 23 Issue 7, July 2009 pp. 823-837

[5]. Daniel Jurafsky & James H. Martin (2006) "Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition", Chapter 6

[6]. Jae-Gil Lee, Jiawei Han, Kyu-Young Whang (2007) "Trajectory clustering: a partition-and-group framework", Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07). ACM, New York, NY, USA, pp. 593-604.

[7]. Jiang Bian, Dayong Tian, Yuanyan Tang, Dacheng Tao (2018), "A survey on trajectory clustering analysis"

[8]. Naoto Mukai (2013) "PageRank-based Traffic Simulation Using Taxi Probe Data", Procedia Computer Science, 2013. 22: pp. 1156-1163.

[9]. Raj Kishen Moloo, Varun Kumar Digumber (2011) "Low-Cost Mobile GPS Tracking Solution", 2011 International Conference on Business Computing and Global Informatization

[10]. Sameer Darekar, Atul Chikane, Rutujit Diwate, Amol Deshmukh, Prof. Archana Shinde (2012) “Tracking System using GPS and GSM: Practical Approach”, IJSER journal

[11]. S´ebastien Gambs, Marc-Olivier Killijian, Miguel N´u˜nez del Prado Cortez (2011) “Show Me How You Move and I Will Tell You Who You Are”, transactions on data privacy 4 (2011) pp. 103–126

[12]. S´ebastien Gambs, Marc-Olivier Killijian, Miguel N´u˜nez del Prado Cortez (2012) “Next Place Prediction using Mobility Markov Chains” K.4 COMPUTERS AND SOCIETY MPM '12 Proceedings of the First Workshop on Measurement, Privacy, and Mobility

[13]. Sergey Brin, Lawrence Page (1998) “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Computer Networks and ISDN Systems. 30 pp. 107–117

[14]. Wenpu Xing, Ali Ghorbani (2004) “Weighted PageRank Algorith Proceedings of the Second Annual Conference on Communication Networks and Services Researchm”

[15]. Xiaomeng Wang, Ling Peng, Tianhe Chi, Mengzhu Li, Xiaojing Yao, Jing Shao (2015) “A Hidden Markov Model for Urban-Scale Traffic Estimation Using Floating Car Data”, PLoS ONE 10(12).