

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**Nguyễn Văn Nhân**

**DỰ ĐOÁN VỊ TRÍ DI CHUYỂN CỦA NGƯỜI DÙNG  
DỰA TRÊN DỊCH VỤ ĐỊNH VỊ GPS**

**LUẬN VĂN THẠC SĨ TOÁN - TIN HỌC  
Ngành Cơ sở Toán học cho Tin học**

**Tp. Hồ Chí Minh, Năm 2016**



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**Nguyễn Văn Nhân**

**DỰ ĐOÁN VỊ TRÍ DI CHUYỂN CỦA NGƯỜI DÙNG  
DỰA TRÊN DỊCH VỤ ĐỊNH VỊ GPS**

**Chuyên ngành: Cơ sở Toán học cho Tin học**

**Mã số chuyên ngành: 60 46 05**

**LUẬN VĂN THẠC SĨ TOÁN TIN - HỌC**

**NGƯỜI HƯỚNG DẪN KHOA HỌC:  
TS. Nguyễn Thanh Bình**

**Tp. Hồ Chí Minh, Năm 2016**



## **LỜI CẢM ƠN**

Em xin bày tỏ lòng biết ơn sâu sắc đến thầy - TS. Nguyễn Thanh Bình bởi sự động viên, chỉ bảo, hướng dẫn tận tình của thầy nên em mới có thể hoàn thành luận văn này.

Em cũng xin gửi lời cảm ơn đến các thầy cô trong khoa Toán-Tin học trường Đại học Khoa học Tự nhiên TP.HCM đã tận tình dạy dỗ, chỉ bảo kiến thức quý báu giúp em hoàn thành khóa học và làm nền tảng cho nghiên cứu của em. Em cũng gửi lời cảm ơn đến các bạn, các anh chị cao học K23 đã giúp đỡ và luôn đồng hành cùng em trong suốt thời gian học tập và thực hiện luận văn.

Vì thời gian làm luận văn có hạn và trình độ còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp ý kiến của các thầy cũng như là của quý độc giả để luận văn này hoàn thiện hơn nữa.

TPHCM, ngày 30 tháng 6 năm 2016  
Học viên thực hiện:

Nguyễn Văn Nhân

# MỤC LỤC

LỜI CẢM ƠN .....	i
MỤC LỤC .....	ii
DANH MỤC CÁC BẢNG .....	iv
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ .....	vi
LỜI NÓI ĐẦU .....	1
1 GPS là gì .....	1
2 Ứng dụng của dịch vụ định vị GPS .....	3
3 Mục tiêu của luận văn .....	5
4 Sự phát triển của đề tài .....	7
5 Bố cục luận văn .....	9
Chương 1 – THU THẬP DỮ LIỆU .....	11
1.1 Thiết kế cơ sở dữ liệu .....	11
1.2 Phân cụm dữ liệu địa điểm .....	14
1.3 Chuẩn hóa thời gian .....	17
Chương 2 – TẠO BỘ DỮ LIỆU HUÂN LUYỆN .....	20
2.1 Địa điểm hiện tại .....	21
2.2 Thời gian .....	21
2.3 Histogram địa điểm .....	22
2.4 Khoảng cách giữa địa điểm .....	24
2.5 Gán nhãn dữ liệu .....	25
Chương 3 – MỘT SỐ MÔ HÌNH DỰ ĐOÁN CƠ BẢN .....	27

3.1.1 Định nghĩa chuỗi Markov .....	27
3.1.2 Huấn luyện bài toán với mô hình chuỗi Markov .....	29
3.2.1 Mô hình máy hỗ trợ vec-tơ SVM .....	30
3.2.2 Ứng dụng của SVM .....	34
3.2.3 Huấn luyện SVM .....	34
3.3.1 Cây quyết định .....	35
3.3.2 Giải thuật xây dựng cây quyết định .....	38
3.3.3 Xén tia cây quyết định .....	39
 Chương 4- THỰC NGHIỆM VÀ KẾT QUẢ .....	40
4.1 Bỏ thuộc tính địa điểm hiện tại .....	41
4.2 Bỏ thuộc tính thời gian .....	43
4.3 Bỏ thuộc tính histogram địa điểm .....	44
4.4 Tổng kết .....	46
 Chương 5 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	47
5.1 Ưu điểm .....	47
5.2 Khuyết điểm .....	47
5.3 Hướng phát triển .....	48
 CÁC TÀI LIỆU THAM KHẢO .....	49
PHỤ LỤC .....	51

## **DANH MỤC CÁC BẢNG, BIỂU ĐỒ**

Bảng 1.1: Bảng người dùng (user) trong cơ sở dữ liệu.....	12
Bảng 1.2: Bảng venue trong cơ sở dữ liệu.....	12
Bảng 1.3: Bảng check-in trong cơ sở dữ liệu.....	13
Bảng 1.4: Bảng dữ liệu người dùng được trích xuất trên cơ sở dữ liệu.....	14
Bảng 1.5: Bảng dữ liệu 1.4 sau khi được phân cụm .....	16
Bảng 1.6: Bảng dữ liệu 1.5 sau khi được chuẩn hoá theo thời gian .....	19
Bảng 2.1: Bảng ký hiệu trong luận văn.....	20
Bảng 2.2: Bảng giá trị thời gian chia theo trong tuần và cuối tuần .....	21
Bảng 2.3: Bảng giá trị thời gian theo từng ngày .....	22
Bảng 2.4: Bảng giá trị giờ trong ngày.....	22
Bảng 2.5 Ma trận khoảng cách của tập địa điểm .....	24
Bảng 3.1 Một số hàm nhân thường dùng.....	34
Bảng 4.0 Tỉ lệ chính xác dự đoán của từng mô hình .....	41
Bảng 4.1 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính địa điểm hiện tại.....	42
Bảng 4.2 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính thời gian.....	43
Bảng 4.3 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính histogram địa điểm.....	45
Biểu đồ 4.0 Biểu đồ so sánh kết quả dự đoán của các mô hình.....	41
Biểu đồ 4.1 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính địa điểm hiện tại.....	43

Biểu đồ 4.2 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính thời gian .....	44
Biểu đồ 4.3 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính histogram địa điểm .....	45

## **DANH MỤC CÁC HÌNH VẼ**

Hình 0.1. Hệ thống định vị toàn cầu GPS .....	1
Hình 0.2. Vị trí giao nhau của ba vòng tròn.....	2
Hình 0.3 Hình ảnh người dùng Foursquare check-in.....	3
Hình 0.4 Foody- ứng dụng tìm kiếm địa điểm ăn uống.....	4
Hình 0.5 Thiết bị đeo tay (wearable) trả lời giúp cho người dùng .....	6
Hình 0.6. Sự di chuyển hằng ngày của một người dùng.....	7
Hình 0.7 Một phần mô hình Markov được xây dựng bởi Ashbrook .....	8
Hình 1.1 Liên kết giữa các bảng trong cơ sở dữ liệu .....	13
Hình 1.2 Kết quả phân cụm dữ liệu .....	17
Hình 2.1 Số lượng check-in người dùng.....	22
Hình 2.2 Histogram địa điểm của người dùng.....	22
Hình 2.3 Sơ đồ di chuyển của một người dùng trong ngày .....	25
Hình 2.3 Sơ đồ di chuyển của một người dùng trong ngày .....	28
Hình 3.2 Phân lớp dữ liệu với trường hợp đơn giản .....	31
Hình 3.3 Phân lớp dữ liệu trong trường hợp phức tạp .....	31
Hình 3.4 Ánh xạ dữ liệu từ không gian gốc sang không gian đặc trưng .....	31
Hình 3.5 Siêu phẳng với lề cực đại .....	33
Hình 3.6 Bộ dữ liệu cần phân lớp .....	35
Hình 3.7 Cây quyết định phân loại dữ liệu .....	36

# LỜI NÓI ĐẦU

Bài toán “*Dự Đoán Vị Trí Di Chuyển Của Người Dùng Dựa Trên Dịch Vụ Định Vị GPS*” đã được nhiều tác giả quan tâm gần đây. Đầu năm 2002, Daniel và Thad tại viện tính toán Georgia ở Atlanta [1] đã đưa ra một số mô hình để khai thác dữ liệu di chuyển người dùng. Từ khi thiết bị di động thông minh (smartphone Android, Windows Phone, iOS) được tích hợp đầu thu tín hiệu GPS, những công nghệ nổi tiếng như Google, Facebook đã tìm cách lưu lại lịch sử di chuyển của người dùng, thông qua đó, họ sẽ biết được những nơi người dùng thường đi đến nhiều nhất và sẽ kết hợp với những nhà đầu tư tập trung những dịch vụ quảng cáo, khuyến mãi, ăn uống... tại những địa điểm này.

Ngoài tính thương mại, chúng tôi mong muốn đem lại cho người dùng những tiện ích tuyệt vời khác (trình bày ở phần mục tiêu luận văn), đây cũng là động lực và mục tiêu cho chúng tôi thực hiện đề tài luận văn thạc sĩ này. Tiếp theo chúng tôi xin trình bày sơ lược về dịch vụ định vị GPS.

## 1. GPS là gì?

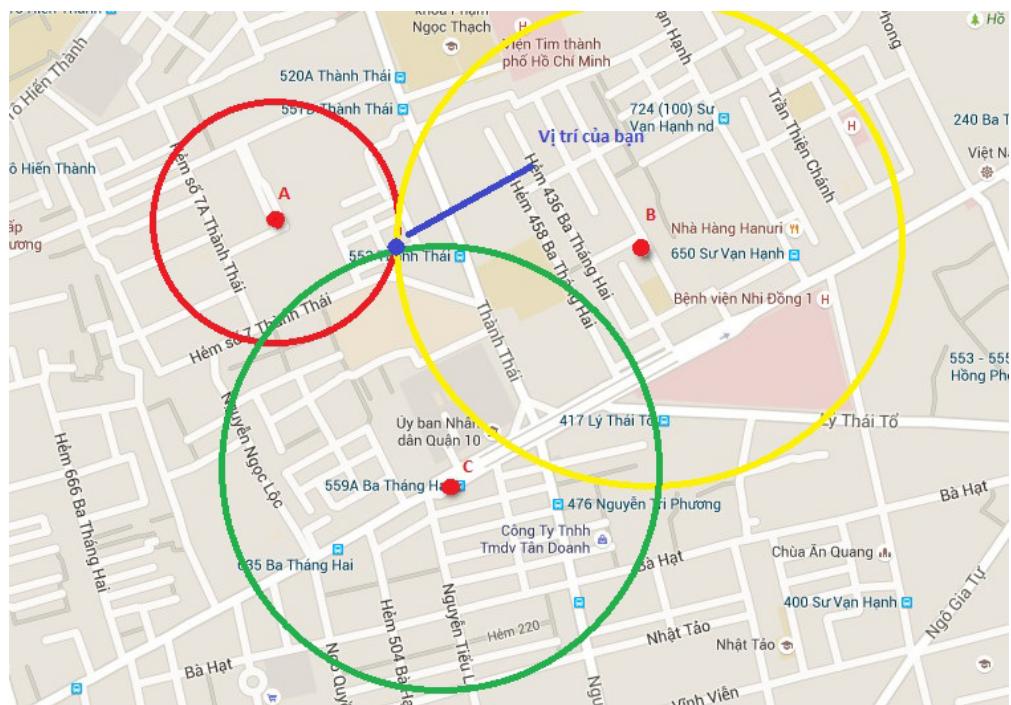


Hình 0.1. Hệ thống định vị toàn cầu GPS [24]

GPS là viết tắt của “global positioning system” (hệ thống định vị toàn cầu), thực chất là một mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng. Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào.

GPS cho phép mọi người trên thế giới sử dụng một số chức năng của GPS miễn phí. Nên bạn có thể sử dụng định vị trên các thiết bị thu GPS để xác định vị trí của mình một cách chính xác và hoàn toàn miễn phí.

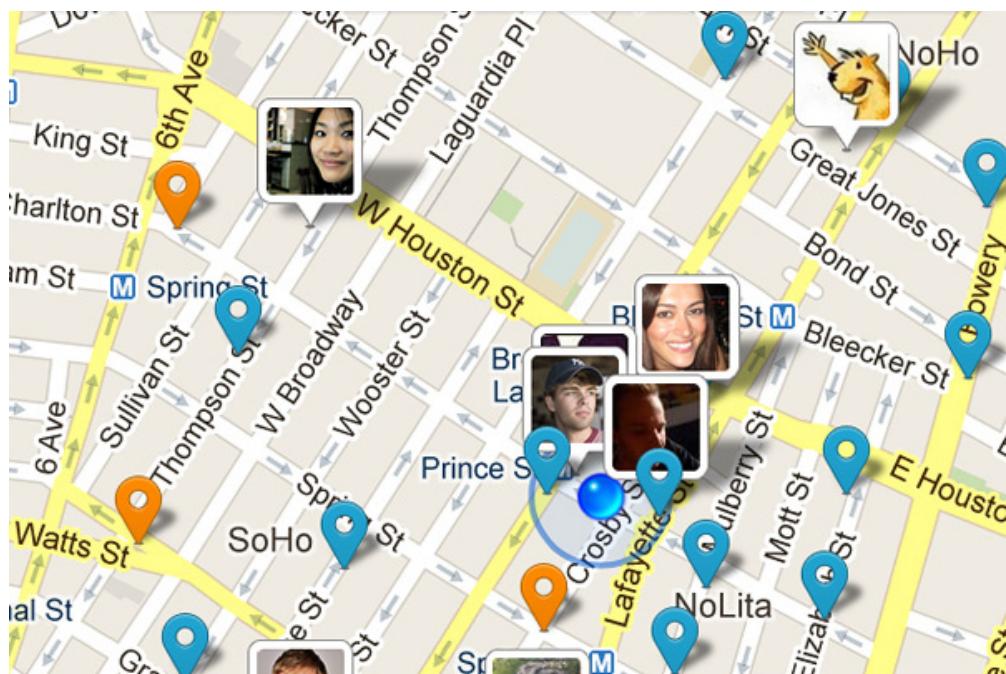
Các vệ tinh GPS bay hai vòng trong một ngày theo một quỹ đạo đã được tính toán chính xác và liên tục phát các tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận các tín hiệu này và giải mã bằng các phép tính lượng giác, qua đó sẽ tính toán và hiển thị được vị trí của người dùng.



Hình 0.2. Vị trí giao nhau của ba vòng tròn chính là vị trí của ban [24]

## 2. Ứng dụng của dịch vụ định vị GPS

Với sự hỗ trợ từ dịch vụ GPS, những mạng xã hội dựa trên địa điểm đang phát triển rất mạnh tại thời điểm hiện tại. Foursquare [18] là ứng dụng đánh dấu địa điểm được phổ biến nhiều nhất trên cộng đồng mạng xã hội. Khi dùng Foursquare bạn có thể tùy ý đánh dấu những địa điểm mình đặt chân đến kèm theo hình ảnh, mô tả đặc trưng cho điểm đó đến với cộng đồng Foursquare (hình 0.3). Cũng tương tự với chức năng đó, người dùng có thể tìm kiếm và tham khảo trước những địa điểm mình sắp đến bằng cách xem hình ảnh, những dòng mô tả và bình luận đánh giá từ những người dùng Foursquare.



Hình 0.3 Hình ảnh người dùng Foursquare check-in. [25]

Ngoài Foursquare, Facebook [19] và Twitter [20] cũng là hai dịch vụ mạng xã hội dựa trên địa điểm phổ biến nhất hiện tại. Hai mạng xã hội nổi tiếng này cho phép người dùng chia sẻ thông tin những địa điểm đã đi qua cho tất cả bạn bè của họ. Nếu bạn muốn tìm kiếm địa điểm ăn uống và những nhà hàng đang có chương trình khuyến mãi, giảm giá tại Việt Nam thì Foody [21] là ứng dụng thích hợp

dành cho bạn (hình 0.4). Foody sẽ giúp bạn tìm những món ăn bạn thích, những địa điểm ăn uống ngon nhưng giá rẻ được đông đảo người dùng lựa chọn.



Hình 0.4 Foody- ứng dụng tìm kiếm địa điểm ăn uống. [26]

Ngoài Foody, còn có địa điểm ăn uống [23] giúp bạn tìm kiếm những địa điểm du lịch tuyệt vời, bạn sẽ được cộng đồng mạng tư vấn, chia sẻ những nơi đẹp nhất trên khắp mọi miền đất nước, sẽ là một ứng dụng không thể bỏ qua đối với những người đam mê du lịch và thích thưởng thức những món ăn nổi tiếng của từng vùng miền.

Những tính năng chính của mạng xã hội kể trên là:

- Cập nhật liên tục các địa điểm check-in của bạn bè.
- Tìm kiếm địa điểm tuyệt vời ngay gần khu mình đang sống.
- Nhận các phiếu giảm giá đặc biệt.
- Ghi lại và chia sẻ những địa điểm đã đi qua.

Những mạng xã hội kể trên đem lại cho họ những khoảng lợi nhuận lớn từ những tính năng quảng cáo hiệu quả cho nhà hàng, quán ăn, khách sạn... Với sự giúp đỡ của mạng xã hội, các địa điểm này ngày càng thu hút nhiều người dùng đến đây nhiều hơn, tạo ra những hiệu ứng tích cực trong việc quảng bá du lịch, hình ảnh con người và đất nước đến với bạn bè quốc tế. Còn đối với người dùng,

lưu lại vị trí di chuyển sẽ có lợi ích gì cho họ? Câu hỏi này cũng sẽ trả lời cho mục tiêu đề tài luận văn của chúng tôi.

### 3. Mục tiêu của luận văn

Người dùng có một danh sách các công việc cần làm tại một địa điểm nhất định, chúng tôi xin nêu ra một số ngữ cảnh sau:

- Nếu người dùng đang lái xe, họ cần có một ứng dụng nhắc nhở phải ghé qua cửa hàng, siêu thị mua đồ dùng khi họ gần đến điểm này. Điều này là cần thiết trong trường hợp họ bực bội hoặc mất tập trung mà quên ghé ngang để mua sắm. Thay vào đó, ta có thể nhắc nhở họ một vài dặm trước khi đến cửa hàng sẽ có hiệu quả hơn.
- Giả sử mô hình di chuyển của anh A cho thấy rằng anh ấy sẽ ăn trưa và uống cà phê tại một nhà hàng vào thứ năm. Nếu chị B cũng làm việc ở khu vực này vào ngày thứ năm gần giờ ăn trưa, chị B muốn có một ứng dụng thông báo rằng anh A ăn trưa gần đó để chị B gọi điện cho anh A, gặp gỡ cùng đi ăn chung và bàn một số công việc liên quan của họ.
- Hãy tưởng tượng rằng người dùng tham gia một lớp học vào lúc 4:00-05:30 chiều mỗi ngày. Khi người dùng bắt đầu vào lớp học, họ mong muốn có một ứng dụng tự động chuyển điện thoại của họ sang chế độ rung để tránh làm ảnh hưởng đến tiết học. Nếu ai đó gọi họ trong giờ học, thiết bị đeo tay (Wearable) có thể tự động trả lời giúp cho họ, để lại tin nhắn với người gọi rằng họ đang trong giờ học và sẽ ra khỏi lớp lúc 05:30. Khi người dùng đi ra khỏi lớp, ứng dụng giúp họ tự động kích hoạt điện thoại sang chế độ chuông và thông báo rằng đã có người gọi đến cho họ. (hình 0.5).

Mục tiêu của luận văn là tạo ra cho người dùng một ứng dụng thực tiễn chạy trên hệ điều hành android của điện thoại di động, giúp họ thực hiện được những tiện ích mà chúng tôi đã đề cập ở một số ngữ cảnh trên. Để thực hiện được điều này, chúng ta phải biết được địa điểm đến tiếp theo mà họ sẽ ghé qua trong thời gian tới (30 phút, 1 giờ, 2 giờ...). Vì thế, chúng tôi thực hiện bài toán khai thác dữ liệu di chuyển của người dùng để dự đoán điểm đến tiếp theo của họ.

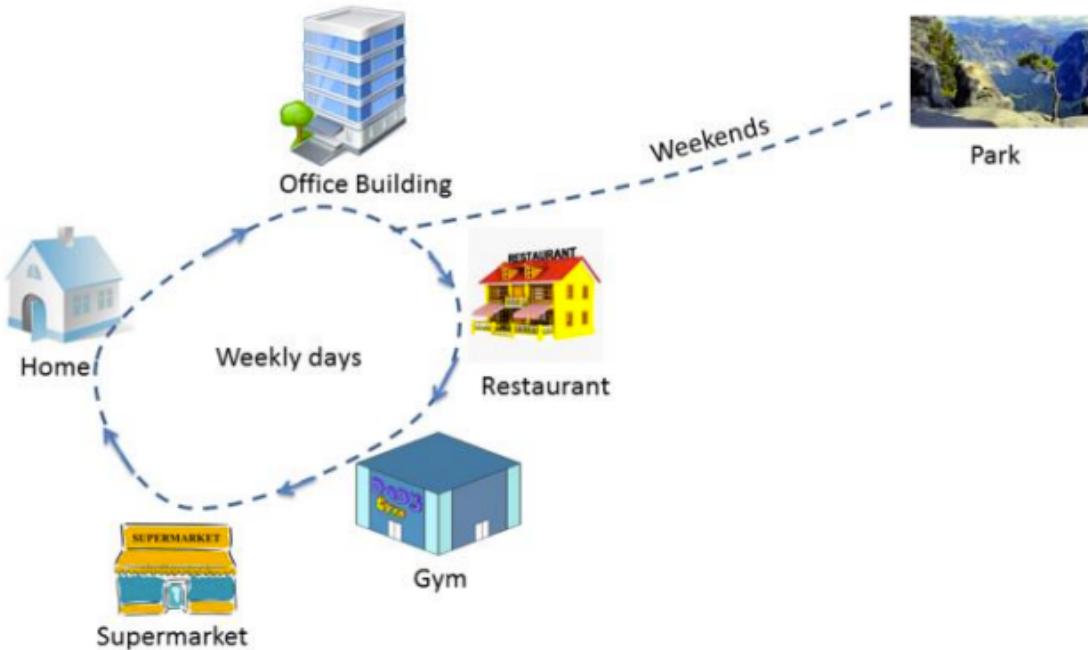


*Hình 0.5 Thiết bị đeo tay (wearable) trả lời giúp cho người dùng khi họ đang bận [27].*

Sự di chuyển của một con người liên quan đến một số tính chất và đặc điểm sau: nghề nghiệp, độ tuổi, giới tính, các mối quan hệ xã hội... Sự di chuyển này thường mang tính chu kỳ, ít thay đổi địa điểm liên tục (hình 0.6).

Chẳng hạn, với một người dùng thông thường thì quy trình di chuyển của họ như sau: trong những ngày làm việc thì buổi sáng đi đến chỗ làm, buổi tối về nhà, cuối tuần thứ 7, chủ nhật thường đi đến những nơi mua sắm, vui chơi, giải trí... nếu chúng ta khai thác được dữ liệu di chuyển này thì có thể dự đoán được

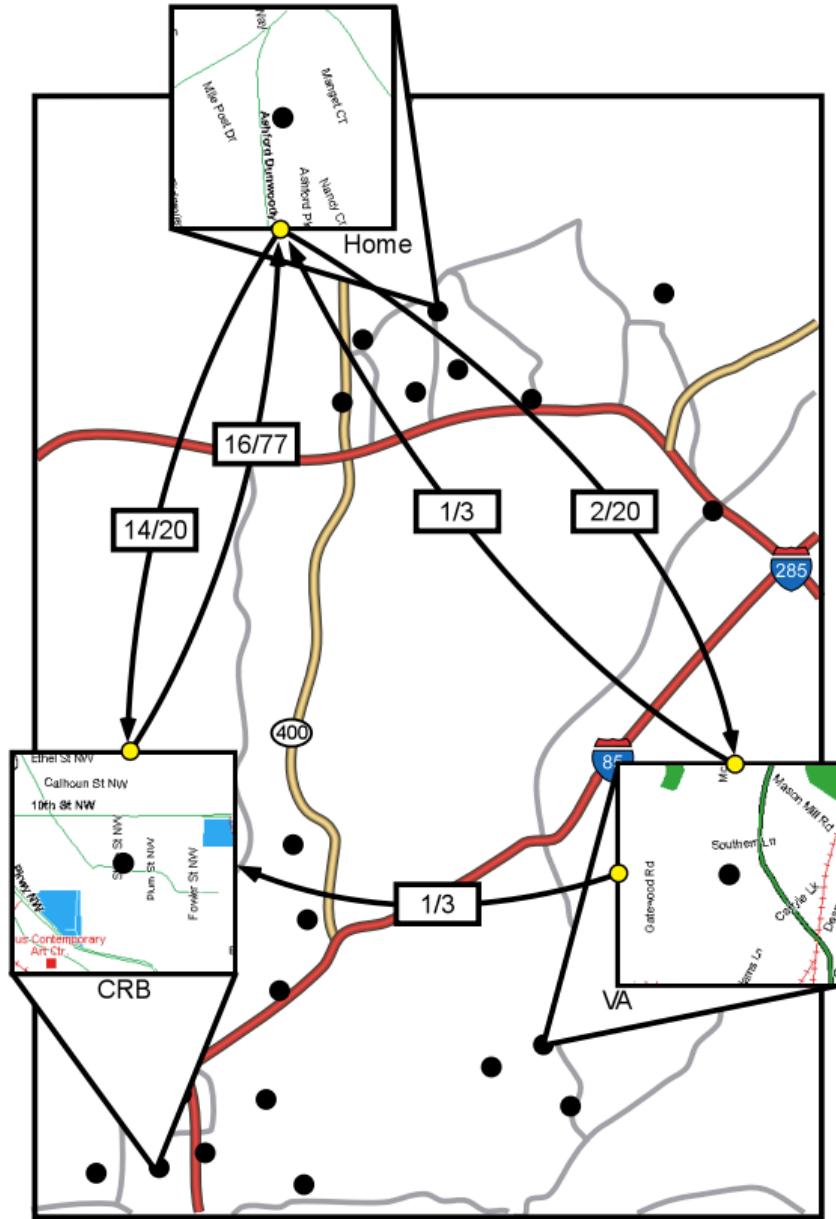
điểm đi đến tiếp theo với điều kiện biết trước được thời gian cùng địa điểm hiện tại của người dùng.



Hình 0.6. Sự di chuyển hàng ngày của một người dùng

#### 4. Sự phát triển của đề tài

Hiện tại có khá nhiều nhóm tác giả đang nghiên cứu về đề tài này. Cao và She [3] đưa ra thuật toán xác định các địa điểm có ý nghĩa đối với người dùng, gom cụm các điểm này thành tập cụm dữ liệu địa điểm và tiến hành huấn luyện với mô hình xích Markov tìm hiểu thêm mối quan hệ xã hội trong ảnh hưởng đến hướng di chuyển tiếp theo của người dùng. Trong các bài báo [2] và [4], các tác giả đã sử dụng những thông tin khác từ người dùng như phương tiện di chuyển (di bộ, xe máy, ô tô, các hoạt động khác như ăn trưa, mua sắm...) hay các điểm truy cập mạng AP (Access Point) làm chuỗi quan sát, chuỗi địa điểm di chuyển làm trạng thái, sau đó tiến hành dự đoán với mô hình Markov ẩn.



*Hình 0.7 Một phần mô hình Markov được xây dựng bởi Ashbrook tại 3 địa điểm Home, CRB (Centennial Research Building) và VA (Dept. of Veterans Affairs) ở thành phố Atlanta, nước Mỹ.*

Năm 2007, các tác giả Akoush và Sameh sử dụng bộ dữ liệu Reality Mining Project tại MIT của dự án Nokia Challenge [6] để tiến hành dự đoán, tác giả sử dụng phương pháp học Bayes kết hợp mạng Neural Networks nhưng kết

quả đạt được không cao, các bài báo [7] và [8] sử dụng mạng hàm cơ sở bán kính RBF (*Radial Basis Neural Network*) một cải tiến của mạng ANN để dự đoán và kết quả đạt được là cao hơn so với sử dụng mạng chuẩn ANN.

Năm 2012, Các tác giả Noulas, Scellato và Lathia [10] đến từ bộ môn Khoa học máy tính, Đại học Cambridge, UK đã đưa ra phương pháp dự đoán mới, họ không chỉ sử dụng địa điểm check-in trong quá khứ, thời gian, không gian mà còn xem xét đến các yếu tố xã hội (quan hệ bạn bè, người thân, các địa điểm bạn bè của họ đã ghé thăm...). Tiếp theo, tác giả tính *rank* (hạng) cho mỗi tính năng cho địa điểm hiện tại, địa điểm nào có *rank* cao nhất sẽ là kết quả của dự đoán.

Các bài báo [9], [11], [12] tiếp cận bài toán theo một hướng khác, họ dùng Conditional Random Fields và kỹ thuật khai phá dữ liệu, khai thác các hoạt động hằng ngày, các mối quan hệ xã hội, dữ liệu lịch sử cuộc gọi... để dự đoán, họ không chỉ dự đoán địa điểm mà còn dự đoán các hoạt động xảy ra tiếp theo của người dùng.

## 5. **Bố cục luận văn**

Để giải quyết bài toán, một trong những bước rất quan trọng của đề tài chính là bước thu thập dữ liệu. Trong luận văn này, chúng tôi thực hiện thu thập dữ liệu di chuyển của người dùng thông qua ứng dụng *Movement Predictor* (xem thêm ở phần phụ lục). Sau khi thu thập dữ liệu xong, chúng tôi bắt đầu chuẩn hóa lại dữ liệu, tiếp theo là tìm kiếm mô hình dự đoán thích hợp nhất.

Để tìm được mô hình thích hợp, chúng tôi thực hiện khảo sát những mô hình cơ bản sau: chuỗi Markov, SVM (Support Vector Machine), cây quyết định (Decision Tree). Để tiện theo dõi, chúng tôi chia ra thành 5 chương, nội dung các chương cụ thể như sau:

**Chương 1:** Trình bày cách thu thập dữ liệu thông qua ứng dụng *Movement Predictor*, thiết kế bảng và mối quan hệ giữa các bảng này trong cơ sở dữ liệu. Chương này cũng trình bày về những thuật toán tiền xử lý dữ liệu bao gồm: phân cụm dữ liệu địa điểm từ những địa điểm trên cơ sở dữ liệu thành những địa điểm quan trọng và có ý nghĩa đối với người dùng, chuẩn hóa bộ dữ liệu theo thời gian.

**Chương 2:** Trình bày cách trích xuất những đặc điểm, tính năng trong dữ liệu di chuyển ở quá khứ của người dùng, từ đó tạo ra bộ dữ liệu vec-tor đặc trưng, tiến hành gán nhãn điểm đến tiếp theo cho từng vec-tor. Tạo ra bộ dữ liệu dùng cho việc kiểm tra.

**Chương 3:** Giới thiệu sơ lược về những mô hình máy học cơ bản như: chuỗi Markov, máy học SVM và cây quyết định. Đây sẽ là nền tảng kiến thức cho việc xây dựng các mô hình để dự đoán vị trí di chuyển của người dùng.

**Chương 4:** Trình bày các phương pháp thực nghiệm, kết quả dự đoán, so sánh hiệu quả của các mô hình.

**Chương 5:** Trình bày những ưu khuyết điểm, hướng phát triển và kết luận của luận văn.

Phần phụ lục ở cuối cùng của luận văn trình bày một số tính năng của ứng dụng *Movement Predictor*, đặc biệt là tính năng dự đoán vị trí di chuyển tiếp theo của người dùng.

# **Chương 1– THU THẬP DỮ LIỆU**

Chương này trình bày cách thu thập dữ liệu di chuyển của người dùng thông qua ứng dụng *Movement Predictor*, thiết kế các bảng thông tin địa điểm, check-in và người dùng trên cơ sở dữ liệu. Sau cùng là những thuật toán chuẩn hoá dữ liệu.

## **1.1. Thiết kế cơ sở dữ liệu**

*Movement Predictor* là một ứng dụng chạy trên hệ điều hành android thích hợp cho các thiết bị điện thoại di động, dùng để thu thập thông tin di chuyển của người dùng. Sau khi cài đặt, ứng dụng sẽ yêu cầu bạn bật GPS để có thể thu thập dữ liệu địa điểm và thông tin di chuyển của bạn. Người dùng có thể check-in tại những địa điểm có sẵn trong danh sách hiển thị của ứng dụng, được chúng tôi liên kết với SDK (Software Development Kit) của *Foursquare*. Ứng dụng tương thích với những thiết bị android có hệ điều hành phiên bản từ 4.0 trở lên và phải có tích hợp sẵn GPS.

SDK của *Foursquare* là một bộ công cụ dùng cho nhà phát triển muốn lấy tất cả danh sách địa điểm nổi tiếng theo tổng quát hay một danh mục cụ thể chẳng hạn như: ăn sáng, ăn trưa, ăn tối, coffee... gần một nơi xác định. Một số tính năng khác của SDK dùng để lấy dữ liệu từ người dùng *FourSquare*, chúng tôi đính kèm trong phần phụ lục cuối cùng của luận văn này.

### **a. Bảng thông tin người dùng:**

Thông tin của một người dùng được lưu trữ trên hệ thống máy chủ bao gồm: họ tên, giới tính, địa chỉ email và những định danh *id* mạng xã hội mà họ tham gia, thông qua *id* này, chúng ta có thể khai thác thêm những địa điểm check-in khác và thông tin nghề nghiệp, bạn bè, người thân của họ. Thông tin người dùng được thiết kế theo bảng 1.1.

Trường (field)	Kiểu dữ liệu	Ý nghĩa
Id (Khóa chính)	Integer	Id của người dùng
firstname	Text	Họ
lastname	Text	Tên
gender	Tiny int	Giới tính
email	Text	Địa chỉ email
socialid	Text	Id facebook người dùng
usertype	Text	Loại người dùng

Bảng 1.1: Bảng người dùng (user) trong cơ sở dữ liệu.

### b. Bảng thông tin địa điểm:

Thông tin một địa điểm bao gồm: Tên địa điểm, địa chỉ, cặp tọa độ GPS (lat, lng), thuộc thành phố, quốc gia nào. Thông tin địa điểm được lưu trên cơ sở dữ liệu như sau:

Trường (field)	Kiểu dữ liệu	Ý nghĩa
Id (Khóa chính)	Integer	Id của địa điểm check-in
name	Text	Tên địa điểm
address	Text	Địa chỉ
lat	Text	Kinh độ
lng	Text	Vĩ độ
cc	Text	Country code (Mã quốc gia)
city	Text	Thành phố
country	Text	Quốc gia
category	Text	(Nhà hàng, khách sạn, sport...)
venuetype*	Tiny Integer	Loại địa điểm

Bảng 1.2: Bảng venue trong cơ sở dữ liệu.

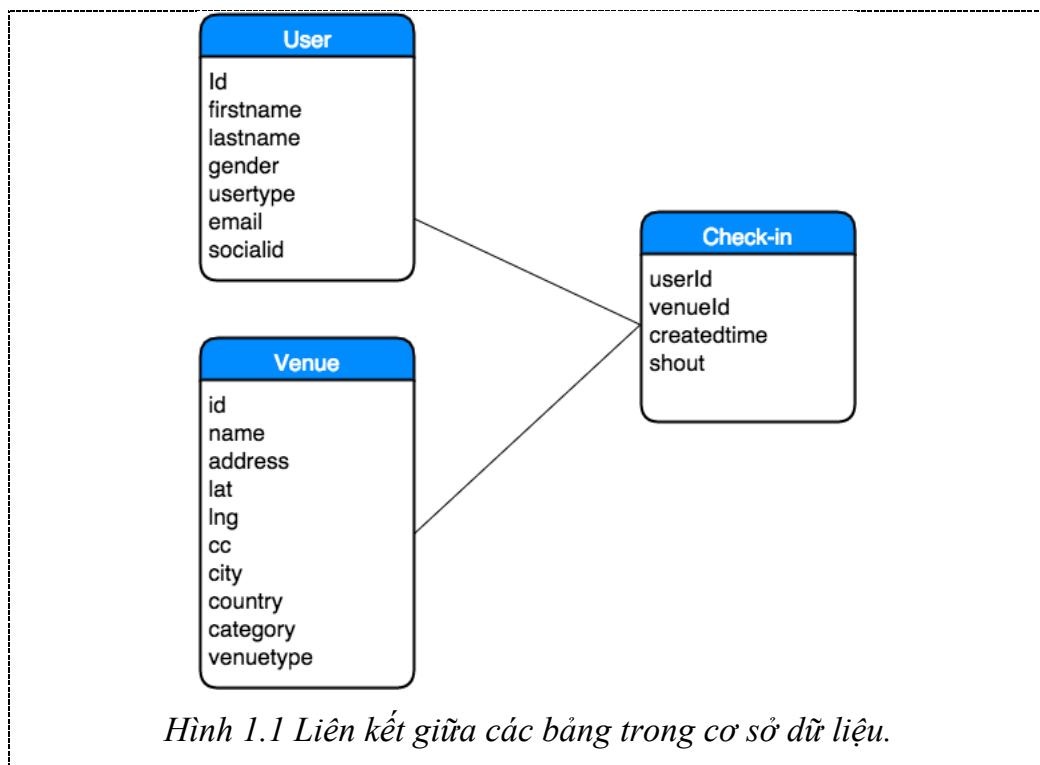
*venuetype*\* có thể nhận một trong các giá trị số sau: 0, 1, 2 và 3 tương ứng với địa điểm này thuộc danh mục check-in của Facebook, Twitter, Foursquare và người dùng tự tạo ra.

### c. Bảng thông tin check-in:

Thông tin check-in bao gồm: *id* của người dùng, *id* của địa điểm, thời gian check-in vào thời điểm nào và một vài dòng ghi chú của người dùng về điểm check-in này.

Trường (field)	Kiểu dữ liệu	Ý nghĩa
userId (khóa chính)	Integer	Id của người dùng
venueId (khóa chính)	Integer	Id của địa điểm
createdTime	Datetime	Thời gian check-in
shout	Text	Thông tin check-in người dùng.

Bảng 1.3: Bảng check-in trong cơ sở dữ liệu.



Một người dùng có thể check-in tại nhiều địa điểm khác nhau, một địa điểm có thể được nhiều người dùng check-in. *Hình 1.1* thể hiện mối quan hệ giữa bảng người dùng và bảng địa điểm được kết nối thông qua bảng check-in.

## 1.2. Phân cụm dữ liệu địa điểm

Sau khi người dùng cài đặt ứng dụng *Movement Predictor* và bật tín hiệu GPS, ứng dụng sẽ tự động gửi dữ liệu di chuyển lên trên hệ thống máy chủ, mẫu thông tin được trích xuất từ cơ sở dữ liệu như sau:

Name	Address	Lat	Lng	Time
Home	Số 2B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781407	106.661411	08:05:00
2C	2C, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781405	106.661410	08:11:03
Tạp hoá	Số 23B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781410	106.661413	08:14:17
Work	Số 68, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.7648894	106.7056227	08:50:45
Số 57	Số 57, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.764537	106.705908	09:15:37
School	Số 227, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762532	106.682656	10:30:07
Căn tin SP	Số 225, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762439	106.682833	11:35:23
Alo Trà	Số 235, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.761896	106.683168	12:10:56

Bảng 1.4: Bảng dữ liệu người dùng được trích xuất trên cơ sở dữ liệu.

Chúng ta thấy rằng những địa điểm check-in như *School*, *Căn tin SP*, *Alo Trà* rất gần nhau, suy ra ta có thể chọn một địa điểm làm đại diện cho những điểm này. Dữ liệu check-in của một người dùng là liên tục nên cần phải tìm những địa điểm có ý nghĩa đối với người dùng và chỉ tiến hành dự đoán trên những tập địa điểm này, sẽ là khó khăn và vô nghĩa nếu thực hiện dự đoán chính xác địa điểm mà người dùng đang di chuyển, thay vào đó, dự đoán họ sẽ đến khu vực nào trong thời gian tiếp theo, để tìm ra được những địa điểm có ý nghĩa, chúng ta dùng thuật toán phân cụm địa điểm.

Từ một tập dữ liệu ban đầu, chúng tôi gom nhóm các địa điểm ở gần nhau thành một nhóm và chọn một địa điểm đặc trưng cho nhóm, địa điểm này được xem là địa điểm có ý nghĩa đối với người dùng. Thuật toán này được xây dựng trên nền tảng của thuật toán *K-means* có sẵn.

### Thuật toán phân cụm địa điểm

#### **Input:**

$M$ : tập tất cả các địa điểm người dùng check-in chưa qua xử lý.

$K$ : số địa điểm ý nghĩa cho một người dùng.

#### **Output:**

$L$ : tập tất cả các địa điểm sau khi phân cụm.

#### **Thuật toán:**

**Bước 1:** Chọn ngẫu nhiên  $K$  địa điểm trong tập  $L$  cho  $K$  cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm.

**Bước 2:** Tính khoảng cách giữa các địa điểm đến  $K$  tâm (dùng khoảng cách Euclidean).

**Bước 3:** Nhóm các địa điểm vào nhóm gần nhất.

**Bước 4:** Xác định lại tâm mới cho các nhóm.

**Bước 5:** Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng.

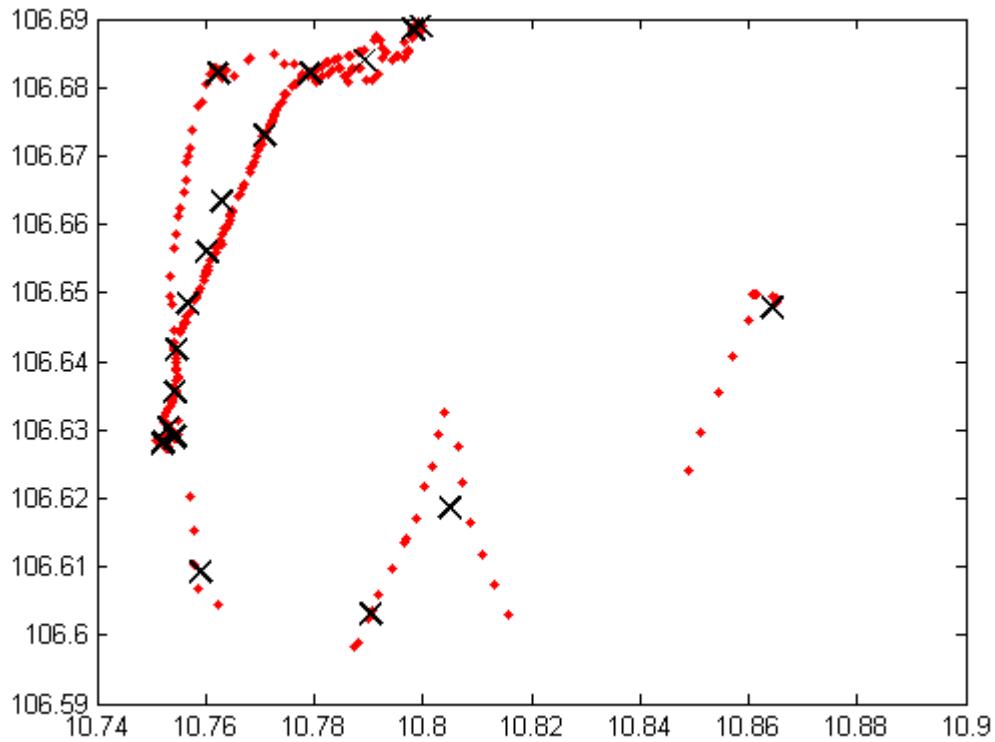
Thông thường, chọn  $K$  có giá trị là 15 đến 20 đại diện cho những địa điểm có ý nghĩa của người dùng,  $I$  có giá trị 1 km là bán kính lớn nhất để xác định một điểm có thuộc vào tâm của một cụm hay không. Bảng dữ liệu 1.4 sau khi thực hiện phân cụm sẽ cho kết quả như sau:

Name	Address	Lat	Lng	Time
Home	Số 2B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781407	106.661411	08:05:00
Home	2C, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781405	106.661410	08:11:03
Home	Số 23B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781410	106.661413	08:14:17
Work	Số 68, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.7648894	106.7056227	08:50:45
Work	Số 57, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.764537	106.705908	09:15:37
School	Số 227, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762532	106.682656	10:30:07
School	Số 225, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762439	106.682833	11:35:23
School	Số 235, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.761896	106.683168	12:10:56

Bảng 1.5: Bảng dữ liệu 1.4 sau khi được phân cụm.

Hình 2.2 là hình ảnh phân cụm cho một người dùng trong hệ thống, có 4 khu vực lớn người dùng hay check-in. Những địa điểm có mật độ check-in nhiều nhất với toạ độ lần lượt là: (10.75, 10.63), (10.8, 106.68) và (10.8, 106.69). Có thể

đoán ra rằng đây là những địa điểm rất có ý nghĩa đối với người dùng, có thể là nhà, nơi làm việc và những nơi ăn uống.



*Hình 1.2 Kết quả phân cụm dữ liệu (trục ngang: vĩ độ, trục đứng: kinh độ, màu đỏ: tập điểm check-in, màu đen: kết quả phân cụm).*

Sau khi phân cụm dữ liệu, chúng tôi tiến hành lưu trữ lại các thông tin đại diện của các cụm (*id*, tên địa điểm, địa chỉ, lat, lng...) và chuẩn hóa lại dữ liệu ban đầu theo tập địa điểm phân cụm này.

### **1.3. Chuẩn hóa thời gian:**

Bảng dữ liệu 1.4 cho thấy rằng, có một số khoảng thời gian khá lâu, chúng ta không thấy người dùng check-in. Có một số nguyên nhân như sau: tín hiệu GPS yếu, mạng kết nối không có hoặc người dùng quên bật GPS... Đối với dữ liệu bị khuyết, thời gian check-in không đồng đều sẽ gây khó khăn cho bước huấn luyện dữ liệu và ảnh hưởng đến kết quả dự đoán.

Vì thế chúng ta cần chuẩn hoá dữ liệu nhằm bổ sung thông tin dữ liệu khuyết và làm đều thời gian check-in của người dùng. Từ bộ dữ liệu (sau khi đã thực hiện phân cụm) ban đầu, chuẩn hóa thời gian check-in  $t$  theo một khoảng nhất định, chúng tôi chọn là  $t = 5$  phút, theo nghĩa là người dùng sẽ lưu lại ở một địa điểm có ý nghĩa tối thiểu 5 phút, như vậy ta sẽ có số khoảng thời gian là:

$$T = \frac{24 \times 60}{5} = 288$$

Chuẩn hóa bộ dữ liệu theo 288 khoảng thời gian cho từng ngày di chuyển của người dùng. Đối với những khoảng thời gian  $t$  mà người dùng không check-in (lý do đã nêu ở trên) thì thực hiện lấy địa điểm liền kề trước khoảng thời gian này  $t - 1$  mà người dùng đã check-in. Nếu tại khoảng thời gian  $t - 1$  mà người dùng  $u$  vẫn chưa thực hiện check-in, chúng tôi sẽ chọn địa điểm phổ biến nhất mà người dùng check-in nhiều nhất.

### Thuật toán chuẩn hóa thời gian

#### **Input:**

$L$ : tập tất cả các địa điểm người dùng đã được phân cụm địa điểm.

#### **Output:**

$S$ : tập dữ liệu đã được chuẩn hóa theo thời gian.

#### **Thuật toán:**

**Bước 1:** Tìm ngày bắt đầu, ngày kết thúc check-in tương ứng  $minVal$ ,  $maxVal$ .

#### **Bước 2:** Repeat

1. Duyệt tất cả địa điểm check-in có thời gian nằm trong khoảng  $[minVal, maxVal]$ .
2. Duyệt  $k$  khoảng thời gian từ 1 đến  $T$  cho từng ngày di chuyển.
3. Tìm  $l \in L$  sao cho  $l_i \equiv k$  nhất.
4. Lưu lại giá trị  $l$  vào tập hợp  $S$ .

Chuẩn hoá dữ liệu có thể chọn những khoảng thời gian khác nhau, chẳng hạn khoảng thời gian  $t$  có thể nhận những giá trị sau: 10, 15, 30, 60 (phút). Tuỳ theo giá trị  $t$ , vec-tơ dữ liệu sau khi chuẩn hoá sẽ có kích thước khác nhau.

Name	Address	Lat	Lng	Time
Home	Số 2B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781407	106.661411	08:05:00
Home	2C, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781405	106.661410	08:10:00
Home	Số 23B, An Lạc A, Quận Bình Tân, Hồ Chí Minh	10.781410	106.661413	08:15:00
<p><i>Điền khuyết dữ liệu - check-in được chọn như sau:</i>  <i>08:15-08:30:Home; 08:30-08:50:Work</i></p>				
Work	Số 68, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.7648894	106.7056227	08:50:00
Work	Số 57, Hoàng Diệu, Quận 4, Hồ Chí Minh	10.764537	106.705908	09:15:00
School	Số 227, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762532	106.682656	10:30:00
<p><i>Điền khuyết dữ liệu - check-in được chọn là School</i></p>				
School	Số 225, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.762439	106.682833	11:35:00
<p><i>Điền khuyết dữ liệu - check-in được chọn là School</i></p>				
School	Số 235, Nguyễn Văn Cừ, Quận 5, Hồ Chí Minh	10.761896	106.683168	12:10:00

Bảng 1.6: Bảng dữ liệu 1.5 sau khi được chuẩn hoá theo thời gian.

## Chương 2 – TẠO BỘ DỮ LIỆU HUẤN LUYỆN

Sau khi chuẩn hoá bộ dữ liệu, chúng ta cần trích xuất những đặc trưng di chuyển trong quá khứ của người dùng, xem xét tính năng nào ảnh hưởng đến sự di chuyển của người dùng, từ đó tạo ra bộ dữ liệu huấn luyện. Trong luận văn này, chúng tôi sử dụng những ký hiệu dưới đây:

Ký hiệu	Ý nghĩa
$lat$	Kinh độ (latitude)
$lng$	Vĩ độ (longitude)
$u$	Một người dùng trong hệ thống.
$l$	Một địa điểm trong hệ thống. $l = (lat, lng)$ .
$t$	Thời gian, lấy giá trị trong tập hợp $T = \{0,1,2,\dots,23\}$
$lc\_id$	location current id: Địa điểm hiện tại
$t\_day$	time day: đặc trưng ngày trong tuần
$h\_range$	hour range: đặc trưng giờ trong ngày
$ln\_id$	location next id: Địa điểm tiếp theo
$v$	Một vec-tơ trong bộ huấn luyện, gồm các giá trị đặc trưng và nhãn.
$U = \{u_1, u_2, \dots, u_m\}$	Tập tất cả người dùng trong hệ thống.
$L = \{l_1, l_2, \dots, l_n\}$	Tập tất cả địa điểm check-in của một người dùng $u$ .
$h_1, h_2, \dots, h_n$	Histogram của $n$ địa điểm $l_1, l_2, \dots, l_n$
$r_1, r_2, \dots, r_n$	Tỉ lệ khoảng cách (ratio) giữa đến và đi của người dùng tại $n$ địa điểm $l_1, l_2, \dots, l_n$

Bảng 2.1: Bảng ký hiệu trong luận văn.

## 2.1. Địa điểm hiện tại

Có thể nói địa điểm hiện tại của người dùng ảnh hưởng nhiều nhất đến điểm di chuyển tiếp theo của họ. Nếu biết được địa điểm hiện tại, quá khứ di chuyển từ địa điểm này đến những địa điểm khác, thì khả năng dự đoán chính xác sẽ cao. Cụ thể, xét tại thời điểm hiện tại  $t$ , người dùng  $u$  đang ở  $l_1$ . Trong quá khứ di chuyển tại  $l_1$ , họ thường đi đến  $l_2$  hoặc  $l_3$ . Vậy khả năng điểm đến tiếp theo sẽ là 2 địa điểm này. Chúng tôi ký hiệu  $lc\_id$  để thể hiện cho địa điểm hiện tại,  $lc\_id$  lấy giá trị trong tập hợp  $L$ .

## 2.2. Thời gian

Thời gian cũng là một đặc trưng quan trọng trong di chuyển của người dùng, xét tại một địa điểm  $l$  vào những khoảng thời gian khác nhau sẽ cho ra những kết quả dự đoán khác nhau. Đối với một người dùng thông thường, trong tuần làm việc họ ít thay đổi quy trình di chuyển, cuối tuần họ sẽ di chuyển nhiều hơn. Chúng tôi ký hiệu  $t\_day$  để thể hiện cho đặc trưng này:

$t\_day$	Ngày trong tuần
$t\_day=1$	Trong tuần làm việc từ thứ 2 đến thứ 6
$t\_day=0$	Cuối tuần thứ 7 và chủ nhật

Bảng 2.2: Bảng giá trị thời gian chia theo trong tuần và cuối tuần.

Chúng ta cũng có thể mở rộng khoảng thời gian dự đoán ra làm 7 ngày trong tuần. Lúc đó  $t\_day$  sẽ nhận những giá trị sau:

$t\_day$	Ngày trong tuần
$t\_day=1$	Thứ hai
$t\_day=2$	Thứ ba
$t\_day=3$	Thứ tư

$t\_day=4$	Thứ năm
$t\_day=5$	Thứ sáu
$t\_day=6$	Thứ bảy
$t\_day=7$	Chủ nhật

Bảng 2.3: Bảng giá trị thời gian theo từng ngày.

Xét từng ngày trong tuần, vào những giờ khác nhau, người dùng cũng sẽ di chuyển tới những địa điểm khác nhau. Để đặc trưng cho tính năng này, chúng tôi ký hiệu  $h\_range$ . Những giá trị của  $h\_range$  như sau:

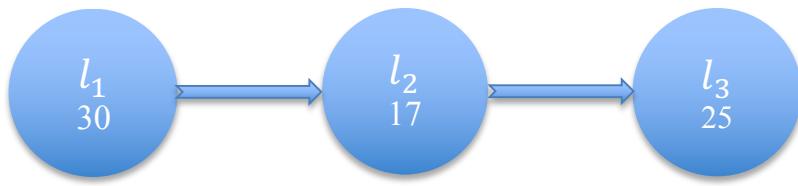
$h\_range$	Giờ trong ngày
$h\_range = 1$	Từ 0 đến 6 giờ
$h\_range = 2$	Từ 6 đến 12 giờ
$h\_range = 3$	Từ 12 đến 18 giờ
$h\_range = 4$	Từ 18 đến 24 giờ

Bảng 2.4: Bảng giá trị giờ trong ngày.

Tương tự như  $t\_day$ , chúng ta cũng có thể chia nhỏ giá trị của  $h\_range$  ra làm nhiều khoảng, chẳng hạn theo khoảng cách là 3 giờ ( $h\_range$  nhận 8 giá trị từ 1 đến 8), và mịn hơn nữa là 1 giờ ( $h\_range$  nhận 24 giá trị từ 1 đến 24).

### 2.3. Histogram địa điểm

Histogram của một địa điểm cho ta biết được tần số xuất hiện của người dùng ở đâu là nhiều nhất trong khoảng thời gian gần đây. Đặc trưng này cũng giúp chúng ta xác định được điểm nào có ảnh hưởng nhiều nhất, thói quen di chuyển của họ và dự đoán điểm đến chính xác hơn. Giả sử người dùng  $u$  đang ở vị trí  $l$  tại thời điểm  $t$ . Trong khoảng thời gian 6 giờ gần đây, họ di chuyển như sau:



Hình 2.1 Số lượng check-in người dùng trong khoảng 6 giờ gần nhất.

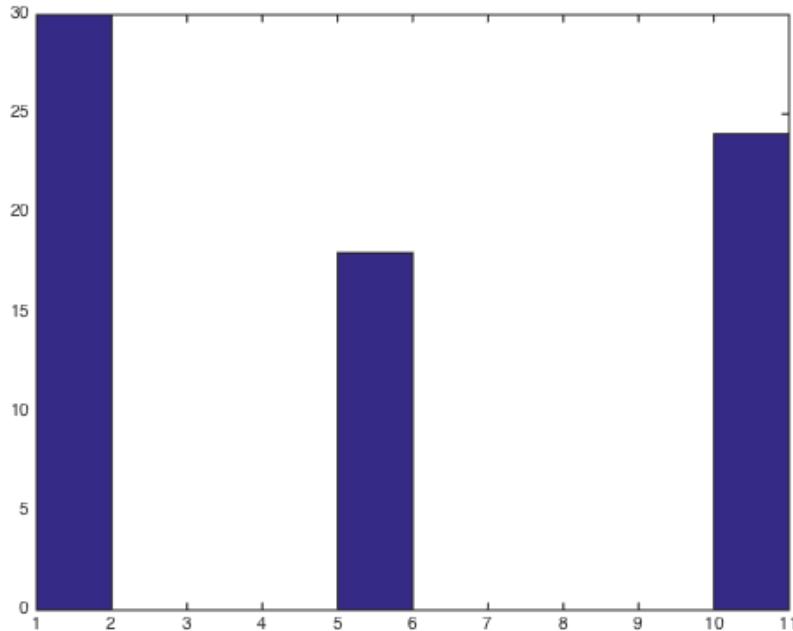
Histogram cho  $l_1, l_2, l_3$  lần lượt là:

$$h_1 = \frac{30}{(30+17+25)} = 0.42$$

$$h_2 = \frac{17}{(30+17+25)} = 0.23$$

$$h_3 = \frac{25}{(30+17+25)} = 0.35$$

Những địa điểm  $l_4, \dots, l_n$  có histogram bằng 0. Dưới đây là hình ảnh histogram của một người dùng trong hệ thống:



Hình 2.2 Histogram địa điểm của người dùng trong khoảng 6 giờ gần đây.

## 2.4. Khoảng cách giữa địa điểm

Khoảng cách giữa các địa điểm giúp chúng ta xác định được điểm nào gần với địa điểm hiện tại của người dùng nhất. Đặc trưng này sẽ quan trọng khi họ đến một nơi hoàn toàn mới, không thuộc vào tập  $n$  địa điểm phân cụm  $\{l_1, l_2, \dots, l_n\}$ . Những địa điểm nào có khoảng cách gần với vị trí hiện tại sẽ là điểm có khả năng nhất. Dưới đây là ma trận khoảng cách của  $n = 8$  địa điểm của một người dùng trong hệ thống.  $d_{ij} = (i, j)$  là khoảng cách giữa 2 địa điểm  $i$  và  $j$ .

0	9	7	1	8	6	5	6
9	0	3	8	1	6	5	13
7	3	0	5	2	5	3	11
1	8	5	0	7	5	4	7
8	1	2	7	0	5	4	12
6	6	5	5	5	0	2	8
5	5	3	4	4	2	0	8
6	13	11	7	12	8	8	6

Bảng 2.5 Ma trận khoảng cách của tập địa điểm.

Giả sử sơ đồ di chuyển của một người dùng trong ngày như hình 2.3.

Chúng tôi ký hiệu  $r_i$  thể hiện cho đặc trưng tỉ lệ khoảng cách đến và đi tại địa điểm  $l_i$ ,  $r_i$  cho từng địa điểm ở sơ đồ trên được tính như sau:

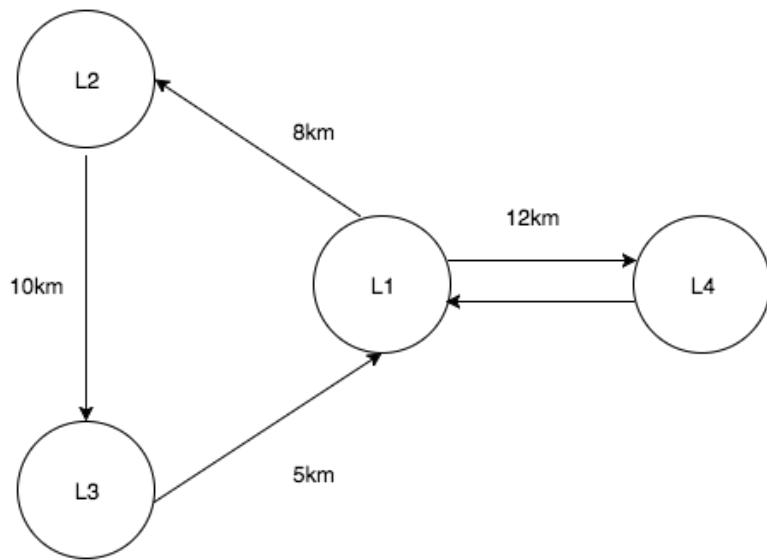
$$r_1 = \frac{d_{31}}{d_{12} + d_{14}} = \frac{5}{8 + 12} = 0.25$$

$$r_2 = \frac{d_{12}}{d_{23}} = \frac{8}{10} = 0.8$$

$$r_3 = \frac{d_{23}}{d_{31}} = \frac{10}{5} = 2$$

$$r_4 = \frac{d_{14}}{d_{41}} = \frac{12}{12} = 1$$

Tỉ lệ khoảng cách giữa đến và đi tại điểm  $l$  cho ta biết được ảnh hưởng của điểm này đến những địa điểm liền kề nó, giúp xác định được điểm đến tiếp theo một cách chính xác hơn.



*Hình 2.3 Sơ đồ di chuyển của một người dùng trong ngày*

Sau khi trích xuất được những đặc trưng địa điểm hiện tại ( $lc\_id$ ), thời gian ( $t\_day, h\_range$ ), histogram ( $h_i$ ) và tỉ lệ khoảng cách giữa đến và đi của địa điểm ( $r_i$ ), chúng tôi tiến hành gán nhãn dữ liệu.

## 2.5. Gán nhãn dữ liệu

Chúng ta hãy xét một trường hợp sau trong bộ dữ liệu: tại thời điểm  $t$  (10AM ngày thứ 2), người dùng  $u$  đang ở vị trí  $l_3$ , trong khoảng thời gian 6 giờ gần đây, họ di chuyển như hình 2.1, biết rằng điểm đến tiếp theo của họ là  $l_1$ , khoảng cách giữa các địa điểm được cho như hình 2.3. và số địa điểm có ý nghĩa của họ là  $n = 12$ . Quy trình thực hiện gán nhãn dữ liệu như sau:

**Bước 1:** Tạo vec-tơ  $v$  có mẫu và kích thước:

$$v = (lc\_id, t\_day, h\_range, h_1, h_2, \dots, h_n, r_1, r_2, \dots, r_n, ln\_id)$$

**Bước 2:** Tính các giá trị:

$$lc\_id = l_3 = 3$$

$$t\_day=1 \text{ (Thứ 2)}$$

$$h\_range = 2 \text{ (Từ 6 đến 12 giờ)}$$

$$h_1 = 0.42, h_2 = 0.23, h_3 = 0.35, h_4, \dots, h_{12} = 0$$

$$r_1 = 0.25, r_2 = 0.8, r_3 = 2, r_4 = 1, r_5, \dots, r_{12} = 0$$

$$ln\_id = l_1=1$$

Vậy:  $v = (3, 1, 2, 0.42, 0.23, 0.35, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.25, 0.8, 2, 1, 0, 0, 0, 0,$   
 $0, 0, 0, 0, 1).$

Một ngày có 24 giờ, chúng tôi tạo 24 vec-tơ tương ứng cho từng giờ trong ngày. Sau  $m$  ngày di chuyển của người dùng, ta sẽ có  $m \times 24$  vec-tơ dữ liệu, chia ra thành 70% dùng cho huấn luyện và 30% cho việc kiểm tra. Qua bước trích đặc trưng và gán nhãn dữ liệu, chúng ta đã có được bộ dữ liệu dùng cho huấn luyện và kiểm tra. Phần tiếp theo trình bày về những mô hình dự đoán cơ bản.

## Chương 3 – MỘT SỐ MÔ HÌNH DỰ ĐOÁN CƠ BẢN

Chương này trình bày những mô hình dự đoán sau: chuỗi Markov, máy học SVM và cây quyết định. Ở đây chúng tôi không có ý định đi sâu chi tiết vào từng mô hình cụ thể mà chỉ giới thiệu một cách cơ bản các khái niệm thông qua các ví dụ cùng với giải thuật huấn luyện. Độc giả có thể tìm hiểu thêm qua phần tham khảo mà chúng tôi đã đính kèm cuối luận văn.

### 3.1.1. Định nghĩa chuỗi Markov

Chuỗi Markov: Trong toán học, một chuỗi Markov (thời gian rời rạc), đặt theo tên nhà toán học người Nga: Andrei Andreyevich Markov [13], là một quá trình ngẫu nhiên thời gian rời rạc với tính chất Markov. Trong một quá trình như vậy, quá khứ không liên quan đến việc tiên đoán tương lai mà chỉ phụ thuộc vào kiến thức về hiện tại.

Xét một chuỗi  $X_1, X_2, \dots, X_n$  gồm các biến ngẫu nhiên. Tập tất cả các giá trị có thể có của các biến này được gọi là *không gian trạng thái*  $S$ , giá trị của  $X_n$  là trạng thái của quá trình (hệ) tại thời điểm  $n$ .

Nếu xác suất có điều kiện của  $X_{n+1}$  khi cho biết các trạng thái quá khứ là một hàm chỉ phụ thuộc  $X_n$ :

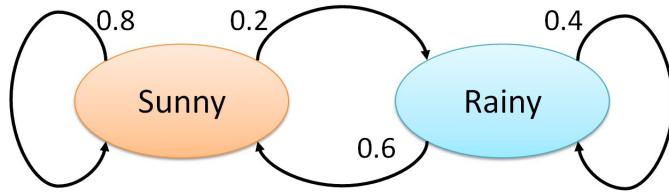
$$P(X_{n+1} = x | X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = x | X_n) \quad (3.1)$$

Thì trạng thái  $X = \{X_1, X_2, \dots, X_n\}$  là chuỗi Markov.

Nếu hệ ở trạng thái  $y$  tại thời điểm  $n$  thì xác suất mà hệ sẽ chuyển tới trạng thái  $x$  tại thời điểm  $n+1$  không phụ thuộc vào giá trị của thời điểm  $n$  mà chỉ phụ

thuộc vào trạng thái hiện tại  $y$ . Do đó, tại thời điểm  $n$  bất kỳ, một chuỗi Markov hữu hạn có thể được biểu diễn bằng một ma trận xác suất, trong đó phần tử  $x, y$  có giá trị bằng  $P(X_{n+1} = x | X_n = y)$  và độc lập với chỉ số thời gian  $n$  (nghĩa là để xác định trạng thái kế tiếp, ta không cần biết đang ở thời điểm nào mà chỉ cần biết trạng thái ở thời điểm đó là gì). Các loại chuỗi Markov hữu hạn rời rạc này còn có thể được biểu diễn bằng đồ thị có hướng, trong đó các cung được gắn nhãn bằng xác suất chuyển từ trạng thái tại đỉnh (*vertex*) đầu sang trạng thái tại đỉnh cuối của cung đó.

Xét ví dụ sơ đồ trạng thái thời tiết  $X$  được ở hình 3.1 dưới đây:



Hình 3.1 Chuỗi Markov cho thời tiết

$X$  gồm hai trạng thái {Sunny (nắng), Rainy (mưa)}. Giả sử tại thời điểm  $t = 0$ . Trời nắng 50% và mưa 50%. Để mô tả thời tiết trong tháng đầu ( $t = 0$ ), chúng ta thiết lập biến ngẫu nhiên  $X(0)$  với quy tắc: nếu trời nắng thì đặt  $X(0) = 1$ , trời mưa thì đặt  $X(0) = 2$ . Lúc đó  $X(0)$  có bảng phân phối xác suất như sau:

Các giá trị của $X(0)$	1	2
Xác suất tương ứng	0.5	0.5

Kí hiệu:  $P[X(0)=1] = \pi_1^{(0)}$ ,  $P[X(0)=2] = \pi_2^{(0)}$  thì vec-tor  $\pi^{(0)} = [\pi_1^{(0)}, \pi_2^{(0)}] = [0.5, 0.5]$  được gọi là vec-tor phân phối xác suất tại thời điểm  $t = 0$  hay vec-tor phân phối ban đầu.

Những tháng sau, ta giả sử xác suất để ngày hôm qua nắng, chuyển sang ngày hôm nay vẫn nắng là 0.8, chuyển sang hôm nay mưa là 0.2. Xác suất để ngày

hôm qua mưa, chuyển sang ngày hôm nay nắng là 0.6, chuyển sang hôm nay mưa là 0.4. Lúc đó các xác suất chuyển thời tiết được thể hiện thông qua *ma trận xác suất chuyển trạng thái*  $P$  (còn gọi là *ma trận chuyển sau một bước*).

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = [p_{ij}]_{2 \times 2}$$

Để tìm bảng phân phối xác suất tại thời điểm  $t = 1$  là  $\pi^{(1)} = [\pi_1^{(1)}, \pi_2^{(1)}]$ .

Ta thực hiện phép nhân ma trận như sau:

$$\pi^{(1)} = \pi^{(0)} P = [0.5, 0.5] \times \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = [0.7, 0.3]$$

Một cách hoàn toàn tương tự, ta cũng có thể tính xác suất chuyển từ  $i$  sang  $j$  sau hai bước là:

$$p_{ij}^{(2)} = p_{i1}^{(1)} \times p_{1j}^{(1)} + p_{i2}^{(1)} \times p_{2j}^{(1)} \quad (3.2)$$

### 3.1.2. Huấn luyện bài toán với mô hình chuỗi Markov

Sự di chuyển người dùng có thể được trừu tượng hóa thành một đồ thị như chuỗi Markov chuẩn. Lịch sử vị trí di chuyển có thể được lưu lại thành một chuỗi quá trình chuyển đổi trạng thái. Sự chuyển đổi xác suất giữa các địa điểm được tính toán dựa trên lịch sử di chuyển theo mô hình chuỗi Markov.

Ta có tập địa điểm sau khi phân cụm là:  $L = \{l_1, l_2, \dots, l_n\}$ . Vì điểm di chuyển tiếp theo của người dùng phụ thuộc vào địa điểm và thời gian hiện tại. Do đó ta phải tạo ra 24 ma trận  $\pi$  và  $P$  khác nhau cho 24 khoảng thời gian trong một ngày cho người dùng. Kí hiệu  $t \in \{0, 1, 2, \dots, 23\}$  thể hiện các khoảng thời gian khác nhau:

1. Tính bảng phân phối xác suất di chuyển trong tháng đầu tiên cho  $n$  địa điểm tại thời điểm  $t$ :  $\pi_t^{(0)} = [\pi_{1t}^{(0)}, \pi_{2t}^{(0)}, \dots, \pi_{nt}^{(0)}]$ .

$$\pi_{it}^{(0)} = P(l = l_i) = \frac{count(l = l_i)}{\sum l}$$

Trong đó:  $count(l = l_i)$  là số check-in của người dùng ở điểm  $l_i$  tại thời điểm  $t$  và  $\sum l$  là tổng số check-in của người dùng trong tháng đầu tiên.

2. Tạo ra ma trận  $P_t$  có kích thước  $n \times n$  để lưu những xác suất di chuyển của người dùng trong 24 khoảng thời gian  $t$  khác nhau trong một ngày. Ta tính  $P_t$  như sau:

$$P_t(i, j) = P_t(l_{t-1} = l_i | l_t = l_j) = \frac{count(l_{t-1} = l_i, l_t = l_j)}{count(l_{t-1} = l_i)}$$

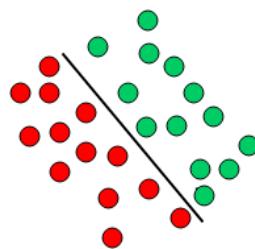
Trong đó:  $count(l_{t-1} = l_i, l_t = l_j)$  là số check-in của người dùng ở địa điểm  $l_j$  tại thời gian  $t$  với ràng buộc lúc  $t-1$  người dùng đã check-in tại điểm  $l_i$ .  $count(l_{t-1} = l_i)$  là số check-in ở điểm  $l_i$  tại thời điểm  $t-1$ .

3. Bảng phân phối xác suất di chuyển tại thời điểm  $t$  bất kì cho  $n$  địa điểm tại ngày di chuyển thứ  $i$  của người dùng được tính như sau:

$$\pi_t^{(i)} = \pi_t^{(i-1)} P_t$$

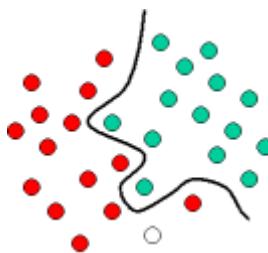
### 3.2.1. Mô hình máy hỗ trợ vec-tor SVM

Máy học véctơ hỗ trợ (SVM) là một giải thuật máy học dựa trên lý thuyết học thống kê do Vapnik và Chervonenkis xây dựng [14]. Bài toán cơ bản của SVM là bài toán phân loại hai lớp: Cho trước  $n$  điểm trong không gian  $d$  chiều (mỗi điểm thuộc vào một lớp kí hiệu là +1 hoặc -1, mục đích của giải thuật SVM là tìm một siêu phẳng (hyperplane) phân hoạch tối ưu cho phép chia các điểm này thành hai phần sao cho các điểm cùng một lớp nằm về một phía với siêu phẳng này. Hình 3.2 cho một minh họa phân lớp với SVM trong mặt phẳng.



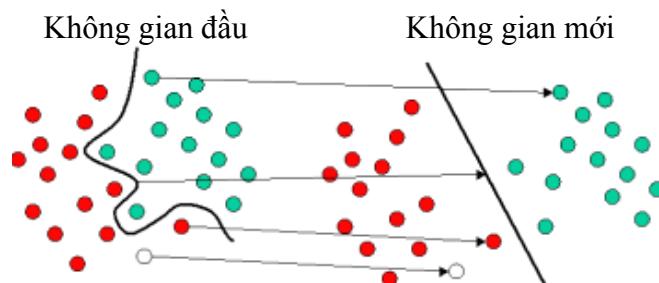
Hình 3.2 Phân lớp dữ liệu với trường hợp đơn giản.

Tuy nhiên, ta thấy ngay mọi chuyện không đơn giản như vậy vì rằng trong thực tế hiếm có các điểm dữ liệu phân bố đẹp như trên để ta chỉ cần dùng một đường thẳng là có thể phân chia được. Hình 3.3 dưới đây minh họa cho điều này.



Hình 3.3 Phân lớp dữ liệu trong trường hợp phức tạp.

Rõ ràng, ở đây dùng đường cong thì sẽ hiệu quả hơn. Vấn đề lại tiếp tục nảy sinh. Đó là việc tìm ra một đường cong như thế là quá phức tạp. SVM giải quyết vấn đề này như sau: từ các điểm dữ liệu ban đầu ta sẽ tìm cách ánh xạ chúng (bằng hàm được gọi là hàm nhân - kernel function) qua một không gian mới (feature space) mà ở đó ta có thể giải quyết được bằng phương pháp tuyến tính.



Hình 3.4 Ánh xạ dữ liệu từ không gian gốc sang không gian đặc trưng cho phép phân chia dữ liệu bởi siêu phẳng.

Một vấn đề nữa là: trong số những siêu phẳng như vậy cần lựa chọn siêu phẳng có lề lớn nhất. Lề ở đây là khoảng cách từ siêu phẳng tới các điểm gần nhất nằm ở hai phía của siêu phẳng (mỗi phía tương ứng với một nhãn phân loại). Lưu ý rằng siêu phẳng nằm cách đều các điểm gần nhất với nhãn khác nhau. Hình 3.5 minh họa siêu phẳng (đường liền nét) với lề cực đại tới các điểm dữ liệu biểu diễn bởi các hình tròn tô đậm và không tô đậm.

Xét tập dữ liệu mẫu có thể tách rời tuyến tính  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  với  $x_i \in R^d$  và  $y_i \in \{\pm 1\}$ . Siêu phẳng tối ưu phân tập dữ liệu này thành hai lớp là siêu phẳng có thể tách rời dữ liệu thành hai lớp riêng biệt với lề (margin) lớn nhất. Tức là, cần tìm siêu phẳng  $H: y = w \cdot x + b = 0$  và hai siêu phẳng  $H_1, H_2$  hỗ trợ song song với  $H$  và có cùng khoảng cách đến  $H$ . Với điều kiện không có phần tử nào của tập mẫu nằm giữa  $H_1$  và  $H_2$ , khi đó:

$$w \cdot x + b \geq +1 \text{ với } y = +1$$

$$w \cdot x + b \geq -1 \text{ với } y = -1$$

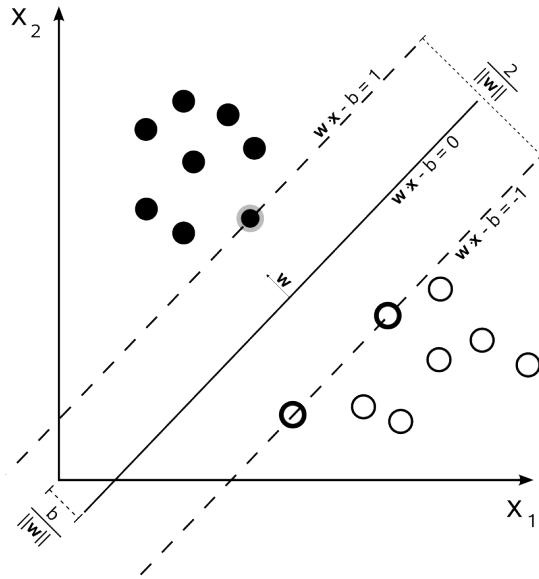
Kết hợp hai điều kiện trên ta có:

$$y(w \cdot x + b) \geq 1$$

Khoảng cách của siêu phẳng  $H_1$  và  $H_2$  đến  $H$  là  $\|w\|$ . Ta cần tìm siêu phẳng  $H$  với lề lớn nhất, tức là giải bài toán tối ưu tìm  $\min_{w,b} \|w\|$  với điều kiện:

$y(w \cdot x + b) \geq 1$ . Người ta có thể chuyển bài toán sang bài toán tương đương nhưng dễ giải hơn là  $\min_{w,b} \frac{1}{2} \|w\|^2$  với ràng buộc  $y(w \cdot x + b) \geq 1$ . Lời giải cho bài toán này là cực tiểu hóa hàm Lagrange:

$$L(w, b, \alpha) = \min_{w,b} \frac{1}{2} \|w\|^2 - \sum_{t=1}^n \alpha_t (y_t(w \cdot x_t + b) - 1).$$



Hình 3.5 Siêu phẳng với lề cực đại cho phép phân chia các hình tròn tô đậm không tô đậm trong không gian đặc trưng.

Trong đó  $\alpha$  là các hệ số Lagrange,  $\alpha \geq 0$ . Sau đó người ta chuyển thành bài toán đổi ngẫu là cực đại hóa hàm  $W(\alpha)$ :

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left( \min_{w,b} L(w, b, \alpha) \right).$$

Từ đó giải để tìm được các giá trị tối ưu cho  $w$ ,  $b$  và  $\alpha$ . Về sau, việc phân loại một mẫu mới chỉ là việc kiểm tra hàm dấu  $\text{sign}(wx + b)$ . Một số hàm nhân  $K$  (kernel), thường dùng được cho trong bảng dưới đây:

Kiểu hàm nhân	Công thức
Linear kernel	$K(x,y) = x \cdot y$
Polynomial kernel	$K(x,y) = (xy + 1)^d$
Radial basis function (Gauss) kernel	$K(x,y) = e^{\frac{- x-y ^2}{2\sigma^2}}$
Hyperebolic tangent kernel	$K(x,y) = \tanh(a \cdot x, y - b)$

Bảng 3.1 Một số hàm nhân thường dùng.

### 3.2.2 Ứng dụng của SVM

SVM có ứng dụng rộng khắp trong các ngành khoa học, sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ, cụ thể:

- Xử lý ngôn ngữ tự nhiên (Natural Language Processing).
- Máy tìm kiếm (Search Engine).
- Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt ...

### 3.2.3. Huấn luyện SVM

Huấn luyện SVM là việc giải bài toán quy hoạch toàn phương SVM. Các phương pháp số giải bài toán quy hoạch này yêu cầu phải lưu trữ một ma trận có kích thước bằng bình phương của số lượng mẫu huấn luyện.

Đối với bài toán dự đoán vị trí di chuyển của người dùng, thuật toán SVM xem mỗi vector  $x_t$  là một vector đặc trưng:

$$x_t = (lc\_id, t\_day, h\_range, h_1, h_2, \dots, h_n, r_1, r_2, \dots, r_n)$$

hay:

$$x_t = (v_1, v_2, \dots, v_{n-1})$$

$$y_t = ln\_id \Leftrightarrow y_t = v_n$$

Một vec-tơ di chuyển mới  $x$  của người dùng được dự đoán theo công thức:

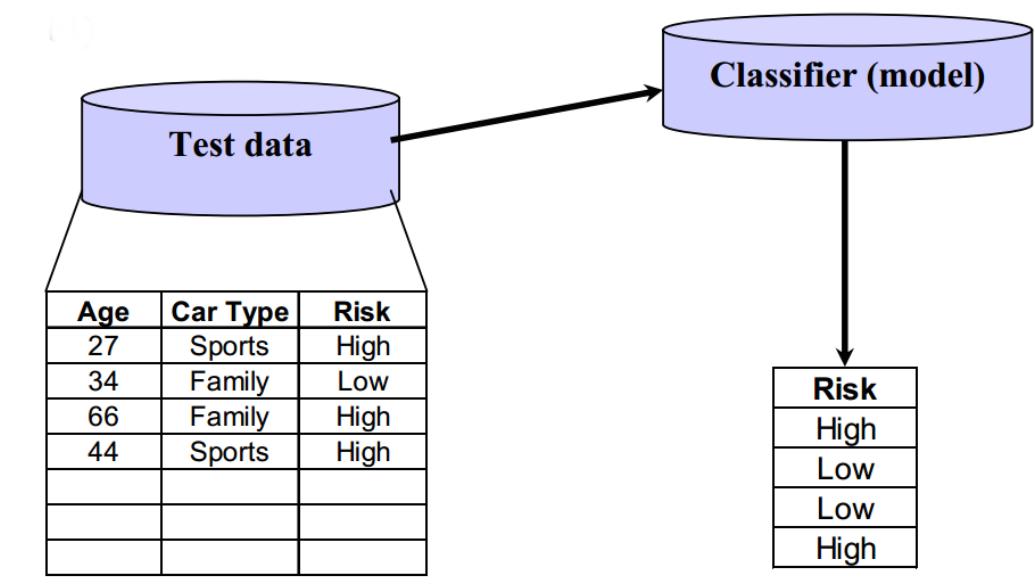
$$f(x) = \text{sign}(wx + b - l_i)$$

trong đó  $l_i \in L = \{l_1, l_2, \dots, l_n\}$

Nếu  $f(x) \geq 0$  thì điểm đến tiếp theo chính là  $l_i$ , ngược lại ta xét tiếp những điểm khác trong tập  $L$ . Kết quả sẽ là tập các địa điểm khả năng mà người dùng  $u$  sẽ đi đến trong thời tiếp theo.

### 3.3.1. Cây quyết định

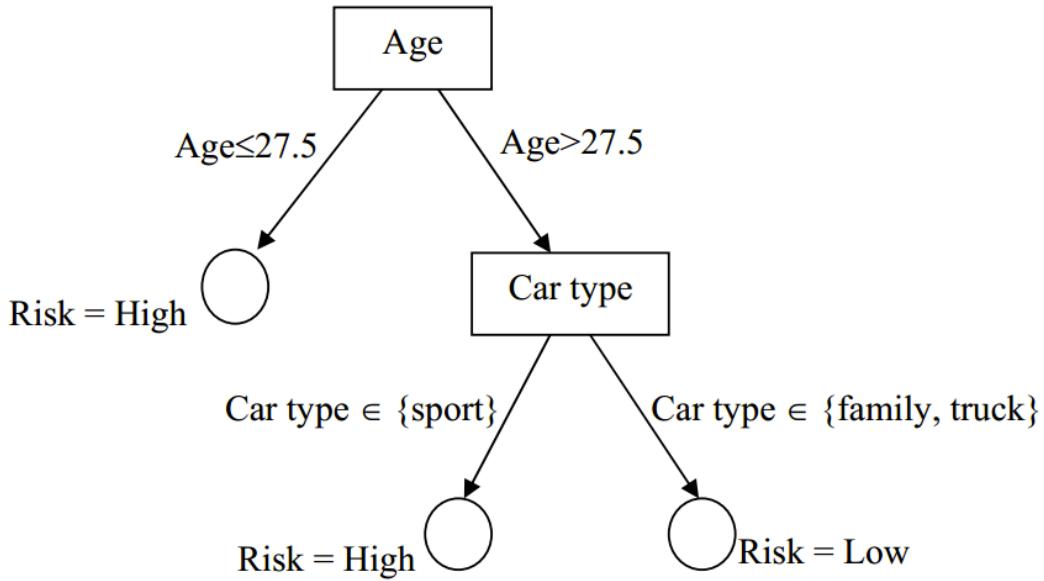
Cho bộ dữ liệu đánh giá độ nguy hiểm khi chạy các loại xe khác nhau ứng với từng độ tuổi như ở hình 3.6. Các loại xe gồm có: thể thao (Sports), gia đình (Family), mức độ nguy hiểm gồm có thấp (low), cao (high). Bài toán đặt ra là phân loại mức độ nguy hiểm khi chạy xe dựa vào thông tin về loại xe và độ tuổi người sử dụng. Để giải quyết những bài toán dạng này, người ta thường dùng phương pháp cây quyết định.



Hình 3.6 Bộ dữ liệu cần phân lớp

Cây quyết định [15] là biểu đồ phát triển có cấu trúc dạng cây, như mô tả trong hình 3.7. Trong cây quyết định:

- Gốc: là node trên cùng của cây.
- Node trong: biểu diễn một kiểm tra trên một thuộc tính đơn (hình chữ nhật).
- Nhánh: biểu diễn các kết quả của kiểm tra trên node trong (mũi tên).
- Node lá: biểu diễn lớp hay sự phân phối lớp (hình tròn).



Hình 3.7 Cây quyết định phân loại dữ liệu.

Để phân loại mức độ nguy hiểm khi chạy xe cho bài toán trên, chúng ta chọn thuộc tính *Age* làm node gốc, nếu  $Age \leq 27.5$  thì  $Risk = High$ , ngược lại ta xét tiếp giá trị thuộc tính *Car type*, nếu giá trị thuộc tính này thuộc tập  $\{sport\}$  thì  $Risk = High$  còn không xét tiếp giá trị này có thuộc tập  $\{family, truck\}$  hay không, nếu giá trị thuộc tập này thì  $Risk = Low$  và kết thúc quá trình phân lớp.

Để phân lớp mẫu dữ liệu chưa biết, giá trị các thuộc tính của mẫu được đưa vào kiểm tra trên cây quyết định. Mỗi mẫu tương ứng có một đường đi từ gốc đến lá và lá biểu diễn dự đoán giá trị phân lớp mẫu đó.

Việc xây dựng cây quyết định phụ thuộc vào việc lựa chọn thuộc tính để phân hoạch. Có 3 loại tiêu chuẩn hay chỉ số để xác định thuộc tính tốt nhất phát triển tại mỗi node:

- Gini-index (Breiman và các đồng sự, 1984 [16]): Loại tiêu chuẩn này lựa chọn thuộc tính mà làm cực tiểu hóa độ không tinh khiết của mỗi phân chia. Các thuật toán sử dụng này là CART, SLIQ, SPRINT.
- Information–gain (Quinlan, 1993 [x]): Khác với Gini-index, tiêu chuẩn này sử dụng entropy để đo độ không tinh khiết của một phân chia và lựa chọn thuộc tính theo mức độ cực đại hóa chỉ số entropy. Các thuật toán sử dụng tiêu chuẩn này là ID3, C4.5.
- $\chi^2$ - bảng thống kê các sự kiện xảy ra ngẫu nhiên:  $\chi^2$  đo độ tương quan giữa từng thuộc tính và nhãn lớp. Sau đó lựa chọn thuộc tính có độ tương quan lớn nhất. CHAID là thuật toán sử dụng tiêu chuẩn này.

Dựa vào tính đơn giản và tiện lợi của C4.5, chúng tôi lựa chọn thuộc tính phân hoạch dựa trên độ lợi thông tin (Information–gain) lớn nhất, đó là hiệu giữa độ hỗn loạn thông tin trước và sau phân hoạch với thuộc tính đó. Độ lợi thông tin được tính toán dựa vào độ hỗn loạn thông tin (Entropy) theo công thức (3.1). Giả sử tập huấn luyện  $S$  chứa các vec-tơ  $v$  có nhãn thuộc  $n$  địa điểm, thì độ hỗn loạn thông tin của tập  $S$  là:

$$Entropy(S) = \sum_{i=1}^n (-p_i \log_2 p_i) \quad (3.3)$$

Trong đó  $p_i$  là xác suất để một phần tử (1 vec-tơ đặc trưng  $v$ ) có nhãn là địa điểm  $l_i$ .  $p_i$  chính là tần suất xuất hiện một vec-tơ đặc trưng  $v$  có nhãn là địa điểm  $l_i$  trong tập  $S$ .

Độ lợi thông tin khi dùng thuộc tính  $a$  phân hoạch tập  $S$  thành các tập con tùy theo giá trị của  $a$  (kí hiệu  $Values(a)$  trong công thức) là:

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{S} Entropy(S_v) \quad (3.4)$$

### 3.3.2. Giải thuật xây dựng cây quyết định

**Đầu vào:**

- Tập  $S$  chứa tất cả các thuộc tính đã mô hình hóa thành các vec-tơ  $v$  trong tập huấn luyện.
- Một tập  $L = \{l_1, l_2, \dots, l_n\}$  chứa tất cả các địa điểm của người dùng.

**Đầu ra:**

- Cây quyết định dạng nhị phân cho việc dự đoán điểm đến kế tiếp của người dùng theo tập  $L$ .

**Giải thuật:**

- Bắt đầu: nút gốc chứa tất cả các vec-tơ huấn luyện.
- Nếu dữ liệu tại nút chỉ thuộc 1 địa điểm  $l$  thì nút là nút lá và được gán nhãn là  $l$ .
- Nếu một nút chứa dữ liệu không thuần nhất (thuộc các lớp khác nhau) thì lựa chọn thuộc tính phân hoạch với độ lợi thông tin lớn nhất (giả sử thuộc tính là  $a$  với giá trị  $y$ ,  $y$  gọi là giá trị phân tách); phân chia nút này một cách đệ quy làm hai tập  $S_1, S_2$ ;  $S_1$  chứa các vec-tơ chứa  $a$  nhưng giá trị thuộc tính nhỏ hơn  $y$ ,  $S_2$  chứa các vec-tơ chứa  $a$  và giá trị thuộc tính lớn hơn bằng  $y$ .

Giải thuật dừng khi tất cả các nút lá đã được gán nhãn. Trong ứng dụng, người ta có thể không tiến hành phân hoạch nút đến khi dữ liệu đồng nhất (chỉ thuộc một lớp), người ta dừng phân hoạch khi số phần tử tại nút còn ít hơn một số lượng nào đó và gán nhãn nút theo luật bình chọn số đông của các phần tử chứa tại nút. Điều này nhằm cải tiến tốc độ xây dựng cây và tránh được tình trạng học vẹt.

### **3.3.3. Xén tia cây quyết định**

Cây quyết định vừa được xây dựng thường là lớn, không mang tính tổng quát mà mang tính học vẹt theo tập huấn luyện. Để tăng tính tổng quát của cây, làm cho cây thích ứng với các mẫu dữ liệu mới, chưa được huấn luyện, người ta cắt bớt các nhánh cây hay còn gọi là xén tia cây với một tập kiểm chứng độc lập với tập huấn luyện. Đây gọi là việc xén tia sau, giải thuật chi tiết như sau:

- Với mỗi nút trong (không phải nút lá), cắt bỏ các nhánh phân hoạch nút biến nút đó thành nút lá và gán nhãn theo luật bình chọn số đông.
- Dùng tập kiểm chứng độc lập để kiểm tra độ chính xác (precision) của cây mới sau mỗi thao tác xén.
- Nếu sau khi xén, độ chính xác của cây được tăng lên thì giữ nguyên việc xén và tiếp tục quá trình xén cho các nút trong còn lại; ngược lại thì trả lại hiện trạng ban đầu (không thực hiện việc xén tia).

Thuật toán dừng khi tất cả các nút đã được xem xét để xén tia.

Việc thực hiện xén tia cây như vậy có độ phức tạp thời gian lớn do phải dùng tập kiểm chứng để ước lượng lỗi sinh ra khi xén tia. Trong thực hành chúng tôi áp dụng giải thuật xây dựng cây với giải pháp bình chọn trên số đông, nếu số đông vượt ngưỡng đặt ra thì dừng việc phân hoạch. Như vậy, chúng tôi không thực hiện thao tác xén tia cây.

## Chương 4 – THỰC NGHIỆM VÀ KẾT QUẢ

Trong chương này chúng tôi trình bày kết quả chạy thực nghiệm trên bộ dữ liệu đã được tạo ra ở chương 2 thông qua ứng dụng *Movement Predictor*. Bộ dữ liệu này của những người dùng trên hệ thống thực hiện thu thập một cách đều đặn qua 3 tháng di chuyển. Một giờ di chuyển của người dùng cho ta 1 dòng dữ liệu, một ngày có 24 dòng dữ liệu, thông qua 3 tháng chúng ta có: 2.160 dòng dữ liệu cho một người dùng.

Để so sánh tỉ lệ chính xác dự đoán của từng mô hình, chúng tôi chia thời gian ra làm 4 khoảng như sau:

**Từ 0h – 6h:** Khoảng thời gian này người dùng ít di chuyển nhất, họ chỉ ở nhà và ngủ nên tỉ lệ dự đoán chính xác sẽ cao.

**Từ 6h – 12h:** Khoảng thời gian này người dùng thường xuyên di chuyển, họ thường đi ăn sáng, đưa con đến trường và đi tới cơ quan làm việc...

**Từ 12h – 18h:** Khoảng thời gian này người dùng thường làm việc tại cơ quan, họ cũng có thể di chuyển tới những địa điểm khác để gặp khách hàng hoặc phục vụ cho công việc.

**Từ 18h – 24h:** Khoảng thời gian này người dùng thường rời khỏi nơi làm việc, họ về nhà ăn tối hoặc có thể di chuyển tới một số địa điểm ăn uống, vui chơi giải trí..

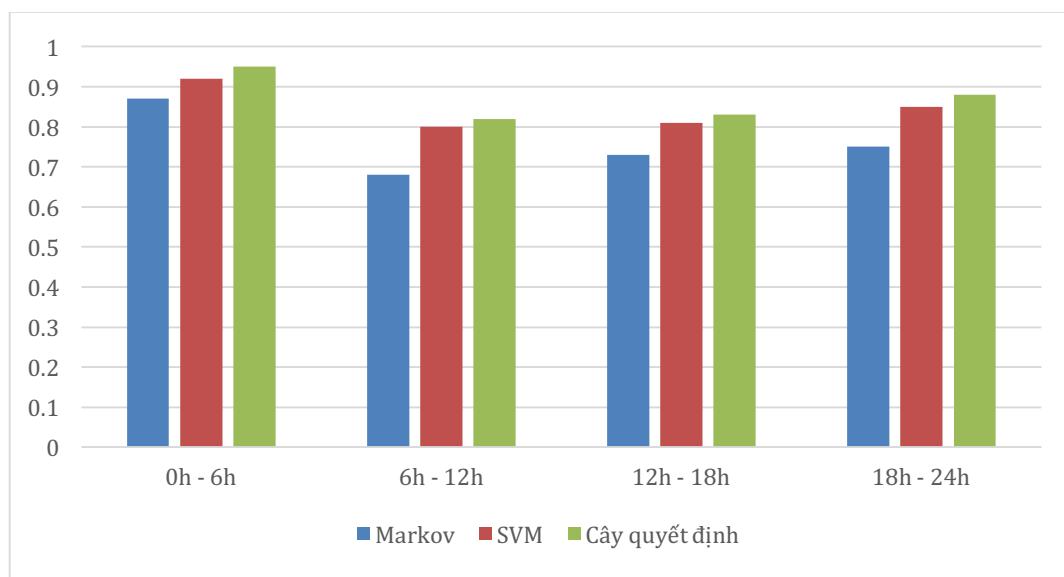
Chúng tôi sử dụng ngôn ngữ Python scikit-learn [22], phiên bản 0.17.1. Thư viện này bao gồm nhiều mô hình thuật toán đã được xây dựng sẵn, Thông qua huấn luyện thực nghiệm trên hệ điều hành Mac OS phiên bản 10.10.5, RAM 8GB.

Kết quả và độ chính xác dự đoán của từng mô hình được thể hiện bằng những bảng và biểu đồ so sánh dưới đây:

Khoảng thời gian	0h – 6h	6h-12h	12h-18h	18h-24h
Chuỗi Markov	87%	68%	73%	75%
SVM	92%	80%	81%	85%
Cây quyết định	95%	82%	83%	88%

Bảng 4.0 Tỉ lệ chính xác dự đoán của từng mô hình.

Để dễ dàng nhận xét tỉ lệ chính xác cũng như so sánh kết quả dự đoán giữa các mô hình, chúng tôi biểu diễn kết quả thông qua biểu đồ sau.



Biểu đồ 4.0 Biểu đồ so sánh kết quả dự đoán của các mô hình.

Biểu đồ cho thấy rằng các mô hình cho kết quả cao nhất trong khoảng 0h – 6h, và thấp nhất trong khoảng 6h – 12h. Cụ thể, chuỗi Markov cho kết quả dự đoán thấp hơn so với hai mô hình đó SVM và cây quyết định, độ chính xác cao nhất là 87% và thấp nhất là 68%, SVM với tỉ lệ cao nhất là: 92% và thấp nhất là 80%, cây quyết định cho kết quả cao nhất là 95% và thấp nhất là 82%.

Bảng kết quả ở trên là sự kết hợp của tất cả những thuộc tính trong tập huấn luyện mà chúng ta đã tạo ra ở chương 2. Tiếp theo, chúng ta xem xét thuộc tính nào có ảnh hưởng đến kết quả dự đoán lớn nhất, bằng cách bỏ đi từng thuộc tính trong tập huấn luyện, chúng ta có được những kết quả dưới đây:

#### **4.1 Bỏ đi thuộc tính: *Địa điểm hiện tại***

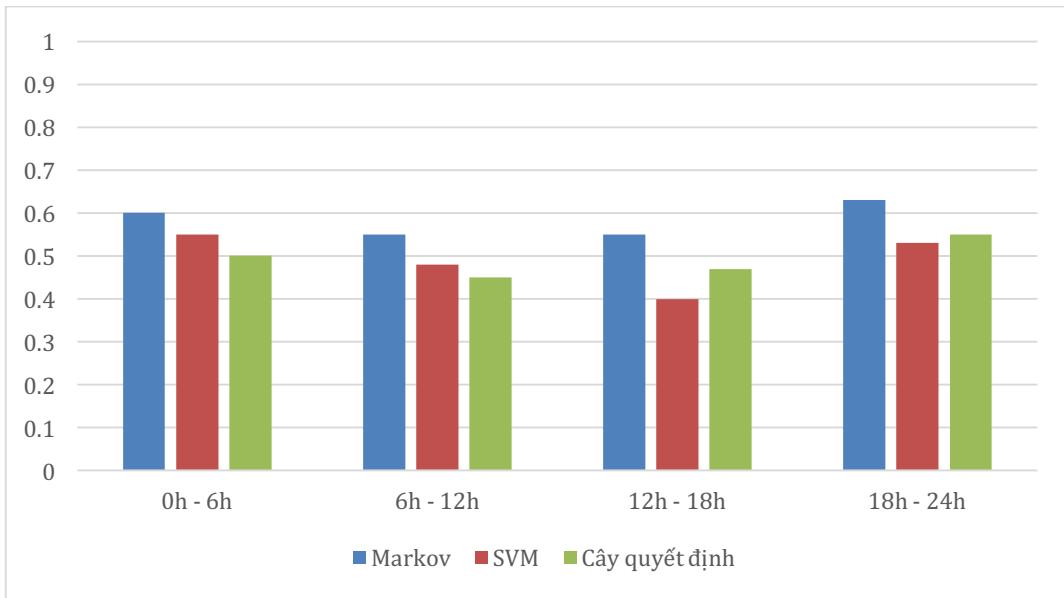
*Địa điểm hiện tại* là một thuộc tính quan trọng cho biết trạng thái của người dùng tại thời điểm dự đoán. Khi bỏ đi thuộc tính *địa điểm hiện tại* trong tập huấn luyện, kết quả dự đoán như sau:

Khoảng thời gian	0h – 6h	6h-12h	12h-18h	18h-24h
Chuỗi Markov	60%	55%	55%	62%
SVM	55%	48%	40%	53%
Cây quyết định	50%	45%	47%	55%

*Bảng 4.1 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính *địa điểm hiện tại*.*

Biểu đồ 4.1 thể hiện kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính *địa điểm hiện tại*. Biểu đồ cho thấy kết quả dự đoán giảm đi đáng kể sau khi bỏ đi thuộc tính này.

Những mô hình cho kết quả cao nhất trong khoảng 18h - 24h và thấp nhất vào khoảng 0h – 6h. Mô hình Markov cho kết quả cao nhất trong khi đó SVM cho kết quả thấp nhất. Cụ thể, kết quả cao nhất và thấp nhất của mô hình Markov là: 62% và 55%, SVM: 55% và 40%, cây quyết định: 55% và 45%.



*Biểu đồ 4.1 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính địa điểm hiện tại.*

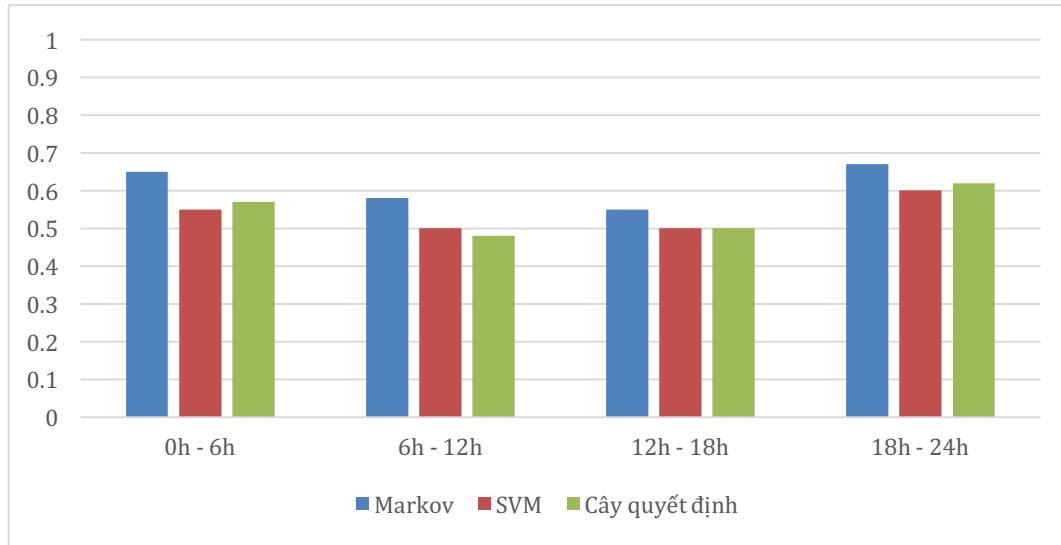
#### **4.2 Bỏ thuộc tính: Thời gian**

Thời gian cũng là một thuộc tính quan trọng cho mô hình dự đoán. Khi bỏ đi thuộc tính thời gian trong tập huấn luyện, tỉ lệ chính xác của các mô hình được thể hiện bảng dưới đây:

Cửa sổ thời gian	0h – 6h	6h-12h	12h-18h	18h-24h
Chuỗi Markov	65%	58%	55%	67%
SVM	55%	50%	50%	60%
Cây quyết định	57%	48%	50%	62%

*Bảng 4.2 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính thời gian.*

Biểu đồ so sánh tỉ lệ chính xác trong dự đoán của các mô hình sau khi bỏ đi thuộc tính *thời gian*:



*Biểu đồ 4.2 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính thời gian.*

Biểu đồ 4.2 cho thấy rằng khi bỏ đi thuộc tính *thời gian*, kết quả dự đoán giảm xuống nhưng so với tỉ lệ chính xác khi bỏ đi thuộc tính *địa điểm hiện tại* thì vẫn cao hơn. Khoảng thời gian cho kết quả cao nhất là từ 18h – 24h, cùng cho kết quả tương đối cao là 0h – 6h, thấp nhất là 6h – 12h.

Mô hình Markov cho kết quả tốt hơn so với hai mô hình còn lại. Cụ thể, kết quả cao nhất và thấp nhất của mô hình Markov là: 67% và 55%, SVM: 60% và 50%, cây quyết định: 62% và 48%.

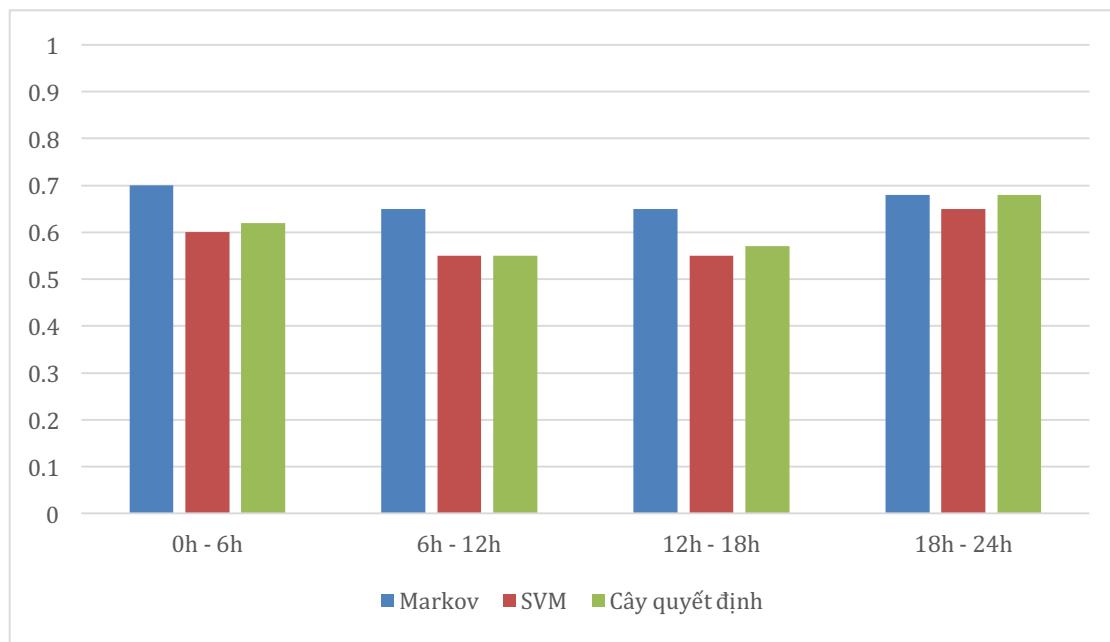
#### **4.3 Bỏ thuộc tính: *Histogram địa điểm***

*Histogram địa điểm* là thuộc tính thống kê tần số xuất hiện của những địa điểm trong khoảng thời gian gần đây, chúng tôi đã giới thiệu ở chương 2. Khi bỏ đi thuộc tính *histogram địa điểm* trong tập huấn luyện, kết quả dự đoán như sau:

Cửa sổ thời gian	0h – 6h	6h-12h	12h-18h	18h-24h
Chuỗi Markov	70%	65%	65%	68%
SVM	60%	55%	55%	65%
Cây quyết định	62%	55%	57%	68%

Bảng 4.3 Tỉ lệ chính xác dự đoán của từng mô hình sau khi bỏ đi thuộc tính histogram địa điểm

Biểu đồ 4.3 so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính histogram địa điểm.



Biểu đồ 4.3 Biểu đồ so sánh kết quả dự đoán của các mô hình sau khi bỏ đi thuộc tính histogram địa điểm.

Biểu đồ này cho thấy rằng, tuy kết quả cao hơn so với bỏ đi thuộc tính *địa điểm hiện tại* hay *thời gian*. Nhưng thuộc tính này cũng ảnh hưởng đến kết quả dự

đoán nhất định. Cụ thể, kết quả cao nhất và thấp nhất của từng mô hình như sau: Markov: 70% và 65%, SVM: 65% và 55%, cây quyết định: 68% và 55%.

#### 4.4 Tổng kết

Thông qua những bảng kết quả của các mô hình ở trên, chúng ta thấy rằng: Về thời gian cho kết quả dự đoán cao nhất là 0h – 6h, thấp nhất là 6h – 12h. Trong khoảng 12h – 28h, tỉ lệ dự đoán đạt ở mức ổn định, còn khoảng 18h – 24h cũng cho kết quả tương đối cao.

Khi có đầy đủ thông tin những thông tin thuộc tính trong dự đoán thì mô hình cây quyết định cho kết quả cao nhất với độ chính xác trung bình là 87%, mô hình Markov cho kết quả thấp nhất với 75% và SVM cho kết quả 84%. Điều này cũng có thể dễ dàng giải thích được vì mô hình Markov chỉ đơn giản là quan sát sự chuyển trạng thái và thống kê sự lặp lại giữa các địa điểm, không có thêm những yếu tố histogram và khoảng cách giữa địa điểm như những mô hình còn lại trong dự đoán. Suy ra, trong bước thu thập dữ liệu, chúng ta phải lấy được thông tin *địa điểm hiện tại* và *thời gian check-in* tại điểm này.

Trong trường hợp bỏ đi thuộc tính *địa điểm hiện tại* kết quả dự đoán của các mô hình thấp nhất, *thời gian* cũng ảnh hưởng nhiều đến kết quả dự đoán, khi bỏ đi thuộc tính này thì mô hình chuỗi Markov cho tỉ lệ dự đoán chính xác cao nhất (67%). Thuộc tính *histogram* *địa điểm* cũng có ảnh hưởng đến kết quả nhất định.

Mô hình Markov cho kết quả dự đoán cao hơn những mô hình khác khi mất thông tin thuộc tính dữ liệu, điều này cho thấy rằng mô hình xác suất sẽ có lợi khi chúng ta không có thông tin về trạng thái hiện tại của người dùng.

## **Chương 5 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Thông qua việc thực hiện đề tài luận văn này, chúng tôi đã đạt được những kinh nghiệm và kết quả nhất định. Sau đây là những ưu khuyết điểm, kết luận và hướng phát triển của đề tài.

### **5.1 Ưu điểm**

- Thu thập được nhiều dữ liệu di chuyển của người dùng, thực hiện chuẩn hoá dữ liệu, khảo sát những thuộc tính ảnh hưởng đến sự di chuyển và đưa ra được những mô hình dự đoán cho bài toán.
- Những mô hình đã dự đoán được vị trí có khả năng nhất người dùng sẽ di đến ở bước tiếp theo khi cho trước thông tin di chuyển trong quá khứ. Về cơ bản, thuật toán đã cho ra kết quả tốt đối với những người dùng ít di chuyển ngẫu nhiên.
- Tạo được ứng dụng *Movement Predictor* chạy trên hệ điều hành android, đem lại một số tiện ích như: tự động chuyển trạng thái điện thoại sang chế độ im lặng theo yêu cầu người dùng tại một số khu vực nhất định mà họ đã cài đặt, nhắc nhở họ lịch làm việc hằng ngày.

### **5.2 Khuyết điểm**

- Bỏ sót những địa điểm có ý nghĩa đối với người dùng trong trường hợp những địa điểm này gần với điểm đại diện cho cụm tai bước phân cụm dữ liệu.
- Chưa định danh được những địa điểm đại diện cụm (tên địa điểm) nếu người dùng không cung cấp, chưa xác định được những hoạt động (ở nhà, đi bộ, đi xe đạp, xe máy, ôtô...) của người dùng giữa các địa điểm.

- Tỉ lệ dự đoán chính xác thấp (50%) đối với những người dùng ít thực hiện check-in hoặc những người dùng thường hay di chuyển ngẫu nhiên.
- Chưa tích hợp được thông tin quan hệ xã hội của người dùng như mối quan hệ bạn bè, bố mẹ ... trong dự đoán.

### 5.3 Hướng phát triển

- Chúng tôi đã triển khai mô hình *Markov* trên ứng dụng *Movement Predictor*, chúng tôi sẽ tiếp tục triển khai những mô hình dự đoán khác trên ứng dụng này và thực hiện đánh giá sự hiệu quả của từng mô hình thông qua số lượng người dùng thực tế.
- Tích hợp những thông tin quan hệ xã hội của người dùng vào trong dự đoán. Những thông tin này chúng tôi thu thập thông qua những mạng xã hội Facebook, Twitter... có liên kết đến ứng dụng *Movement Predictor*. Ngoài ra, chúng tôi cũng xét thêm các yếu tố xã hội khác (sở thích ăn uống, vui chơi, mua sắm...) ảnh hưởng đến hành vi và di chuyển của người dùng.
- Xây dựng thêm những tính năng mới cho ứng dụng *Movement Predictor* với tính năng là dự đoán khả năng bạn bè của người dùng ghé qua địa điểm nào trong thời gian sắp tới, có thể tạo được cuộc gặp gỡ giữa họ hay không. Tạo ra mô hình dự đoán kết hợp giữa những người dùng, giúp họ chia sẻ những thông tin hữu ích chẳng hạn địa điểm đang xảy ra kẹt xe, cửa hàng đang khuyến mãi...

## CÁC TÀI LIỆU THAM KHẢO

- [1] Daniel Ashbrook and Thad Starner, *Learning Significant Locations and Predicting User Movement with GPS*, International Symposium on Wearable Computers (ISWC), p.101-108, 2002.
- [2] Pratap S. Prasad and Prathima Agrawal, *Movement Prediction in Wireless Networks Using Mobility Traces*, Consumer communications and networking conference (CCNC), p.714-718, 2010.
- [3] Lei CAO and James SHE, *Can Your Friends Predict Where You Will Be?* International Conference on Internet of Things (iThings), p.450 - 455, 2014.
- [4] Wesley Mathew, Ruben Raposo and Bruno Martins, *Predicting Future Locations with Hidden Markov Models*, Conference on Ubiquitous Computing (UbiComp), p.911-918, 2012.
- [5] Dave Touretzky and Kornel Laskowski, *Neural Networks for Time Series Prediction*, IEEE, Fall 2006.
- [6] Sherif Akoush and Ahmed Sameh, *Mobile User Movement Prediction Using Bayesian Learning for Neural Networks*, International Conference on Systems and Networks Communications (ICSNC), p.191-196, 2007.
- [7] Saikath Bhattacharya and Sudhansu Sekhar Singh, *Location Prediction Using Efficient Radial Basis Neural Network*, International Conference on Information and Network Technology, p.530-535, 2011.
- [8] Fenglian Liu, *An Improved RBF Network for Predicting Location in Mobile Network*, International Conference on Natural Computation (ICNC), p.345 - 348, 2009.
- [9] N Yang, X Kong, F Wang and PS Yu, *When and Where: Predicting Human Movements Based on Social Spatial-Temporal Events*, International Conference on Data Mining (SIAM), p.515-523, 2014.

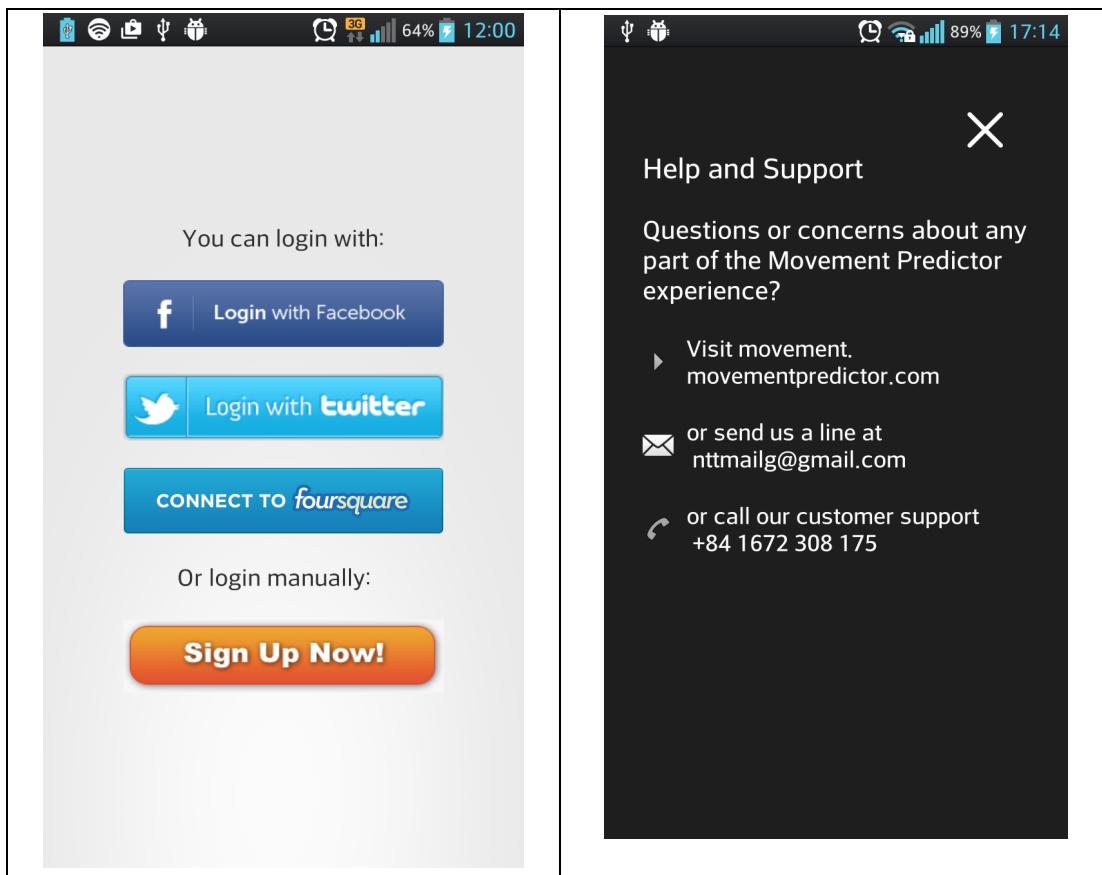
- [10] Anastasios Noulas, Salvatore Scellato, Neal Lathia, Cecilia Mascolo, *Mining User Mobility Features for Next Place Prediction in Location-based Services*, International Conference on Data Mining (SIAM) 2012, p.1038-1043, 2012.
- [11] Lin Liao, Dieter Fox and Henry Kautz, *Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields*, International Journal of Robotics Research, p.119-134, 2007
- [12] He W, Li D, Zhang T, *Mining regular routes from GPS data for ridesharing recommendations*, International Workshop on Urban Computing, p. 79-86, 2012.
- [13] U.H. Abou El-Enien, *Graph Drawing algorithms for Markov Chains*, ISSR, Cairo University, 2006.
- [14] J. Weston, *Support Vector Machine Tutorial*, Independence Way, Princeton, USA, 2011.
- [15] J.R. Quinlan. *Induction of Decision Trees*, Kluwer Academic Publishers, Netherlands, 1986.
- [16] Anurag Srivastava, Eui- Hong Han, Vipin Kumar, Vieet Singh. *Parallel Formulations of Decision-Tree Classification Algorithm*. Kluwer Academic Publisher, 1999.
- [17] <https://developer.android.com>
- [18] <https://foursquare.com>
- [19] <https://www.facebook.com>
- [20] <https://twitter.com>
- [21] <http://www.foody.vn>
- [22] <http://scikit-learn.org>
- [23] <http://diadiemanuong.com>
- [24] <https://www.thegioididong.com/he-thong-dinh-vi-toan-cau-gps.png>
- [25] <https://lenta.ru/articles/2014/05/08/foursquare.png>
- [26] <http://drusticamente.com/wp-content/uploads/2012/07/foody.png>
- [27] <http://burntech.tv/wp-content/uploads/2015/11/SocialWearable.jpg>

# PHỤ LỤC

## Những tính năng của ứng dụng *Movement Predictor*

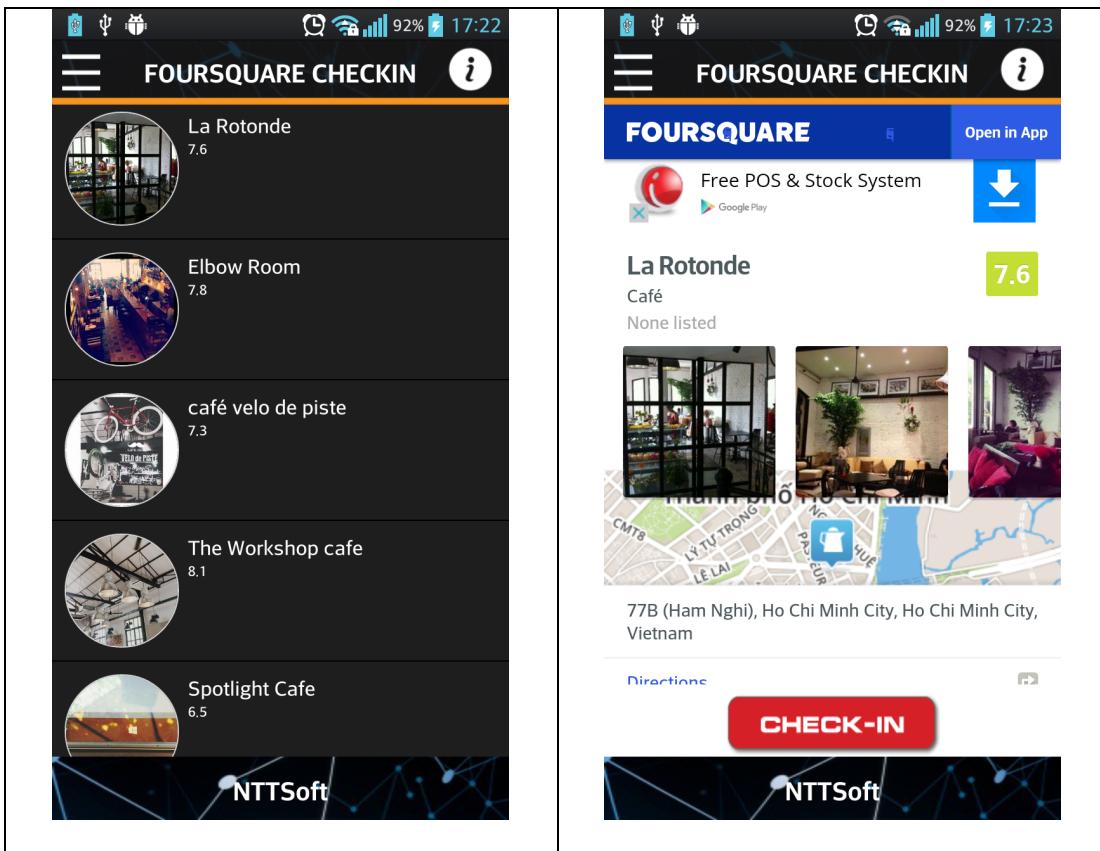
### 1. Đăng ký, đăng nhập

Ứng dụng cho phép người dùng đăng ký thông qua một số tài khoản mà người dùng có sẵn như: Facebook, Twitter, Foursquare hoặc người dùng có thể đăng ký mới trên hệ thống của *Movement Predictor*.



### 2. Tìm địa điểm thông qua API Foursquare

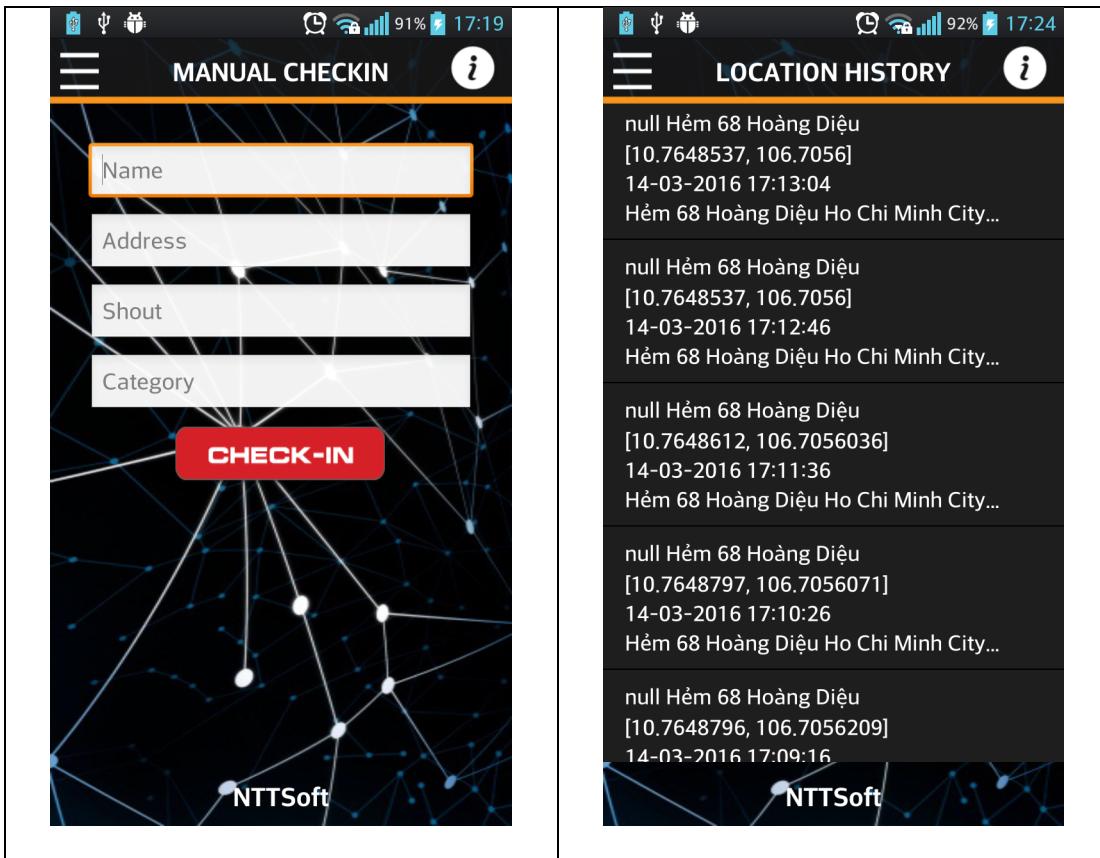
Những API của Foursquare cho phép người dùng tìm kiếm những địa điểm mình yêu thích được tích hợp vào ứng dụng ở vị trí hiện tại như: đồ ăn, đồ uống, coffee, mua sắm, rạp chiếu phim.... hiển thị chi tiết hình ảnh, số lượt đánh giá, số lượt bình luận của người dùng.



### 3. Check-in

Người dùng có thể cài đặt chế độ tự động check-in hoặc chỉ thực hiện check-in tại những địa điểm yêu thích. Để thực hiện check-in, người dùng nhập tên địa điểm, danh mục địa điểm, hình ảnh và kèm theo một vài dòng chú thích của địa điểm này.

Sau khi check-in, địa điểm này sẽ hiển thị trong danh sách các địa điểm đã đi qua. Thông tin hiển thị trong danh sách bao gồm tên địa điểm, tọa độ (*lat, lng*) và thời điểm check-in. Người dùng có thể xem chi tiết địa điểm check-in bằng cách nhấn vào địa điểm này trên danh sách địa điểm đã hiển thị.



#### 4. Dự đoán, nhắc nhở

Đây là tính năng chính của ứng dụng *Movement Predictor*. Tính năng này cho phép xem thông kê địa điểm nào có số lượng check-in nhiều nhất, ít nhất, người dùng thường đi đến những nơi giải trí nào nhiều nhất. Qua đó, ứng dụng sẽ cho biết sở thích chung của người dùng.

Để thực hiện tính năng dự đoán, người dùng phải nhập vào địa điểm (hiện tại), thời gian bắt đầu dự đoán. Sau khi nhập, ứng dụng sẽ gửi thông tin lên trên máy chủ để xử lý, sau đó trả về những địa điểm cùng với phần trăm khả năng mà người dùng đi đến tiếp theo. Sẽ có 3 màu sắc phân biệt tương ứng như sau:

- Màu xanh: Khả năng đi đến điểm này sẽ là cao nhất.
- Màu vàng: Điểm này cũng có khả năng người dùng sẽ đi qua, nhưng thấp hơn.
- Màu đỏ: Là điểm mà khả năng đi đến sẽ thấp nhất.

Thời gian cho dự đoán sẽ tùy thuộc vào lượng dữ liệu trên máy chủ, địa điểm người dùng nhập vào và thời gian.

