BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

TRẦN QUỐC KHÁNH

XÂY DỰNG CHƯƠNG TRÌNH TƯ VẤN DU LỊCH TRÊN ĐIỆN THOẠI DI ĐỘNG

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ thông tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 01 năm 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

TRẦN QUỐC KHÁNH

XÂY DỰNG CHƯƠNG TRÌNH TƯ VẤN DU LỊCH TRÊN ĐIỆN THOẠI DI ĐỘNG

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ thông tin

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS.TS. QUẢN THÀNH THƠ

TP. HỒ CHÍ MINH, tháng 01 năm 2016

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học : PGS. TS. Quản Thành Thơ

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày 20 tháng 03 năm 2016

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

TT	Họ và tên	Chức danh Hội đồng
1	PGS. TS. Võ Đình Bảy	Chủ tịch
2	GS. TSKH Hoàng Văn Kiếm	Phản biện 1
3	TS. Lê Tuấn Anh	Phản biện 2
4	TS. Lê Văn Quốc Anh	Ủy viên
5	TS. Nguyễn Thị Thúy Loan	Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận văn sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày15tháng 09 năm 2015

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: TRẦN QUỐC KHÁNH — Giới tính: NAM

Ngày, tháng, năm sinh: 09/07/1984 — Nơi sinh: KHÁNH HÒA

Chuyên ngành: Công nghệ thông tin — MSHV: 1441860013

I- Tên đề tài:

XÂY DỰNG CHƯƠNG TRÌNH TƯ VẤN DU LỊCH TRÊN ĐIỆN THOẠI DI ĐỘNG.

II- Nhiệm vụ và nội dung:

Tìm hiểu các yếu tố đánh giá một Tour du lịch. Hiện thực giải thuật Apriori và kỹ thuật matrix factorization để tự động hóa ma trận điểm tương tác. Xây dựng giải thuật tìm kiếm Tour và chiến lược đặt câu hỏi cho hợp lý. Xây dựng được hệ thống tư vấn Tour trên điện thoại di động (thiết bị Android).

III- Ngày giao nhiệm vụ: 20/08/2015

IV- Ngày hoàn thành nhiệm vụ: 15/01/2016

V- Cán bộ hướng dẫn: PGS. TS. QUẢN THÀNH THƠ

CÁN BỘ HƯỚNG DẪN **KHOA QUẢN LÝ CHUYÊN NGÀNH**

PGS. TS. QUẢN THÀNH THƠ

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

Trần Quốc Khánh

LỜI CÁM ƠN

Luận văn được thực hiện tại Khoa Công nghệ thông tin - Trường Đại học Công nghệ TP HCM, dưới sự hướng dẫn khoa học của PGS.TS. Quản Thành Thơ.

Trước tiên tôi xin bày tỏ lòng chân thành biết ơn sâu sắc tới thầy PGS.TS. Quản Thành Thơ. Thầy đã tận tình giảng dạy, hướng dẫn giúp tôi tiếp cận và đạt được thành công trong công việc nghiên cứu của mình. Thầy đã luôn tận tâm, động viên, khuyến khích và chỉ dẫn giúp tôi hoàn thành được luận văn này.

Tôi cũng xin cảm ơn bạn Trịnh Văn Giang đang công tác tại trường Đại Học Bách Khoa TP. HCM đã nhiệt tình hỗ trợ tôi hoàn thành luận văn này.

Tôi xin bày tỏ lòng biết ơn tới các Thầy Cô thuộc Khoa Công nghệ thông tin và cán bộ phòng Quản lý khoa học và đào tạo sau đại học - Trường Đại học Công nghệ TP HCM, đã tạo mọi điều kiện thuận lợi giúp đỡ tôi trong quá trình học tập và nghiên cứu tại trường.

Tôi xin chân thành cảm ơn các Thầy Cô trong Hội đồng đánh giá luận văn Thạc Sĩ đã đóng góp ý kiến quý báu giúp tôi hoàn thiện bản luận án.

Học viên thực hiện Luận văn

Trần Quốc Khánh

TÓM TẮT

Thế giới vận động không ngừng dẫn đến một lượng lớn thông tin được đưa lên Internet hằng ngày. Cùng với nhu cầu tìm kiếm thông tin trên Internet ngày càng trở nên phổ biến; một vấn đề đặt ra là một thông tin nên hay không nên đọc, chia sẻ cho một đối tượng người sử dụng khác trên Internet? Và làm thế nào để xác định được thông tin đó có khả năng được người dùng đánh giá cao hay thấp?

Để giải quyết vấn đề này đã có rất nhiều nghiên cứu thực hiện trên các lĩnh vực khác nhau, với những đối tượng thông tin khác nhau. Kết quả của các bài nghiên cứu đó nhằm hỗ trợ đưa ra một Hệ thống tư vấn (Recommender System - RS) phù hợp nhất. Hệ thống tư vấn có thể là một chương trình, một tập hợp các kỹ thuật nhằm đưa ra các khuyến nghị về các đối tượng cho người dùng khi nó có khả năng được sử dụng nhiều nhất.

Việt Nam đã chứng kiến sự bùng nổ của Internet, các hình thức thanh toán trực tuyến và thương mại điện tử đang dần thay đổi thói quen tiêu dùng. Trong đó, mô hình OTA – Online Travel Agencies được hiểu là các doanh nghiệp cung cấp sản phẩm, dịch vụ trực tuyến: vé máy bay, vé tàu, du lịch … cũng đã được nhiều công ty áp dụng mang lại nhiều lợi ích cho khách hàng. Cùng với đó là sự bùng nổ và sự gia tăng nhanh chóng của thiết bị di động, đặt biệt là điện thoại di động thông minh. Nắm được tình hình hiện tại và mục tiêu xây dựng một hệ thống mới giúp ích cho người dùng, nên tôi quyết định chọn đề tài "Xây dựng chương trình tư vấn du lịch trên điện thoại di động". Đây là một hệ thống tư vấn trên điện thoại di động (smartphone), đề xuất ra một Tour du lịch phù hợp với sở thích của khách hàng khi họ muốn đi đâu đó mà chưa định sẵn nơi mình muốn đến.

ABSTRACT

World Campaign constantly led to a large amount of information posted on the Internet every day. Along with the need to find information on the Internet is becoming increasingly popular; a problem arises that information should or should not read, share, for users on the Internet? And how to identify the information that the users is likely to be rated high or low?

To solve this problem have been many researches done on different areas, with different information objects. The results of the researches to support it launched a Recommender System (RS) that best fit for users. Recommender System can be a program, a set of techniques to make recommendations on the subject to the users when it is likely to be used most.

Vietnam has witnessed the rise of the Internet and other forms of online payment and e-commerce are changing consumer habits. In particular, model OTA - Online Travel Agencies are understood as enterprises providing products and services online: airfare, train tickets, tourism ... many companies have also been applied to bring many benefits to customer. Along with the boom and the rapid rise of mobile devices, especially smart mobile phone. Understand the current situation and the goal of building a new system benefits for the customer, so I decided to choose the thesis "Building a Tourism Recommender System program on mobile phone." This is an recommender system on mobile phones (smartphones), proposed a Tour suit for customer preferences when they want to go somewhere but they don’t know where they will visit.

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

Ký hiệu	Tiếng Anh	Tiếng Việt
TRS	Tourism Recommeder System	Hệ thống tư vấn du lịch
GPS	Global Positioning System	Hệ thống định vị toàn cầu
POI	Points of Interest	Các điểm quan tâm
TTDP	Tourist trip design problem	Lỗi tối thiểu
OP	Orienteering problem	Bài toán chạy định hướng
TOP	Team Orienteering Problem	Bài toán chạy định hướng nhóm
TOPTW	TOP with Time Windows	
PEU	Product – Enviroment – User	Mô hình tác động của 3 yếu tố tổng quát
RS	Recommender Systems	Hệ thống tư vấn
RDF	Resource Description Framework	Khung mô tả tài nguyên
RDFS	Resource Description Framework Schema	Lược đồ khung mô tả tài nguyên
OWL	Ontology Web Language	Ngôn ngữ dùng để mô tả các hệ cơ sở tri thức
OTA	Online Travel Agencies	Đại lý du lịch trực tuyến

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH VẼ, ĐỒ THỊ

CHƯƠNG 1
GIỚI THIỆU

1.1. LÝ DO CHỌN ĐỀ TÀI

Ngày nay du lịch đã trở thành một nhu cầu không thể thiếu của con người bởi khi đời sống của con người được nâng cao thì nhu cầu nghỉ ngơi thư giãn sau những giờ làm việc căng thẳng trở nên ngày càng cần thiết. Người dùng có xu hướng tìm kiếm thông tin về du lịch để tham khảo và chọn lựa cho mình chuyến du lịch phù hợp nhất. Với môi trường internet tràn ngập thông tin, làm cho người dùng rất khó tự quyết định, hoặc sẽ mất rất nhiều thời gian và công sức để chắc lọc được thông tin phù hợp với nhu cầu sở thích của mình. Thực tế này dẫn đến sự cần thiết, vai trò quan trọng của hệ thống tư vấn du lịch cho người dùng nắm bắt thông tin, quyết định nhanh chóng dễ dàng cho chuyến du lịch ưng ý của mình.

Thông tin trên mạng rất phong phú và đa dạng, chính vì vậy người dùng thường tìm kiếm thông qua các website tìm kiếm nổi tiếng (google.com, bing.com, search.yahoo.com…), các website cung cấp du lịch trực tuyến của các công ty du lịch cung cấp cho người dùng các Tour du lịch được lọc theo những thuộc tính nhất định như nơi đến, giá cả, thời gian đi, cự ly tuyến...

Nếu bạn có thời gian để đến trực tiếp các đại lý cung cấp Tour du lịch, họ sẽ cung cấp cho bạn các Tour cụ thể về địa điểm, giá cả theo yêu cầu của bạn, sẽ hướng dẫn bạn các thắc mắc cụ thể để đề xuất các Tour du lịch cho bạn, tuy nhiên sẽ rất mất thời gian, công ty chỉ cung cấp các Tour trong giới hạn mà mình cung cấp và làm việc trong giờ hành chánh, khách hàng không thể so sánh với các Tour du lịch của các hãng khác.

1.2. MỤC TIÊU CỦA ĐỀ TÀI

Tìm hiểu các hệ tư vấn (recommendation system), các khái niệm, phương pháp

xây dựng 1 hệ tư vấn.

Xây dựng được hệ thống tư vấn Tour trên điện thoại di động (thiết bị Android).

Xây dựng hệ thống tư vấn Tour du lịch có giải thuật tìm kiếm Tour và một cơ chế đưa ra câu hỏi thông minh, dựa trên câu trả lời của người dùng mà đưa ra câu hỏi tiếp theo phù hợp.

1.3. PHẠM VI CỦA ĐỀ TÀI

Đề tài được hiện thực trong phạm vi như sau:

- Xây dựng được hệ thống tư vấn Tour trên điện thoại di động (thiết bị Android).
- Hiện thực giải thuật Apriori và kỹ thuật matrix factorization để tự động hóa ma trận điểm tương tác.
- Hiện thực giải thuật tính điểm Tour và chiến lược đặt câu hỏi cho hợp lý.
- Dùng Ontology để lưu trữ dữ liệu và dùng SPARQL để truy xuất dữ liệu.
- Xây dựng hệ thống Web Service.

1.4. CẤU TRÚC CỦA LUẬN VĂN

Chương 1 giới thiệu tổng quan về đề tài, chương này sẽ giới thiệu chung về đề tài, mục tiêu và phạm vi giới hạn của đề tài. Chương 2 trình bày các nghiên cứu liên quan trong lĩnh vực mà đề tài đang thực hiện như hệ thống tư vấn nói chung và hệ thống tư vấn du lịch nói riêng. Chương 3 trình bày về cơ sở lý thuyết, chương này giới thiệu những kiến thức nền tảng, công nghệ, kỹ thuật cần thiết sử dụng trong quá trình thực hiện đề tài. Chương 4 trình bày các giải thuật, về các thuật toán, chiến lược đặt câu hỏi và lựa chọn câu hỏi tiếp theo được áp dụng trong đề tài. Chương 5 sẽ trình bày lý do chọn triển khai ứng dụng trên điện thoại di động Android, mô hình và hoạt động của hệ thống, thiết kế Ontology, hiện thực giải thuật Apriori và kỹ thuật

Matrix Factorization, thiết kế các chức năng mà mockup ứng dụng Android. Cuối cùng, chương 6 trình bày kết quả kiểm tra, đánh giá hệ thống và giải thuật. Các kết quả đạt được và hạn chế của ứng dụng, từ đó đưa ra kết luận, hướng phát triển và đề xuất hướng mở rộng cho đề tài.

CHƯƠNG 2
CÁC NGHIÊN CỨU LIÊN QUAN

2.1. HỆ THỐNG TƯ VẤN

Recommender System (RS) [1] là hệ thống chọn lọc thông tin cần thiết nhằm đưa ra gợi ý, dự báo phù hợp người dùng về vấn đề thông tin (như là sách, âm nhạc, phim) hoặc vấn đề xã hội (là người, nhóm người) mà người đó có thể chưa xem xét. Các hệ thống RS giới thiệu các khuyến nghị có thể phù hợp tốt hơn với thị hiếu, cá nhân người dùng và hạn chế việc thông tin tràn ngập, quá tải làm rối người dùng.

Nguồn dữ liệu của RS là dạng tường minh (explicitly) được user mô tả, chia sẽ khi yêu cầu việc gợi ý. Dữ liệu input là dạng ngầm định (implicitly) thông qua hoạt động tham dò khảo sát, nhận phản hồi từ hình thức hỏi như "bạn đã mua sản phẩm đó, thì bạn có thể cũng mua cái này".

Hệ thống tư vấn là ngành đặc thù mới, tạo nền tảng cho triển khai tiếp vào các nhóm ngành cụ thể như thương mại điện tử về hàng hóa, dịch vụ, tư vấn, du lịch, đào tạo. Hệ thống tư vấn được nghiên cứu phát triển nhiều và hình thức khá đa dạng, tuy nhiên dựa vào mục tiêu ứng dụng, tri thức được sử dụng, giải thuật xử lý, và cách hệ thống hóa các tư vấn.

Hệ tư vấn được sử dụng nhiều trên các website thương mại của nhiều công ty kinh doanh lớn nhằm dự báo và giúp người sử dụng hướng đến các sản phẩm phù hợp với nhu cầu và sở thích của họ hơn, từ đó nâng cao uy tín, doanh thu và lợi nhuận của công ty. Có thể kể đến một vài hệ tư vấn của một số công ty nổi tiếng như:

- Hệ tư vấn trên Amazon.com: khi xem một sản phẩm trên website của hãng này, hệ thống sẽ đề nghị một danh sách các sản phẩm cộng thêm dựa trên ma trận sản phẩm mà những người mua hàng trước đó đã mua kèm với sản phẩm đang được xem.

- Netflix (*Hình 2.1*) lại quan tâm đến việc dự báo những phim người tiêu dùng thích xem dựa trên kết quả bình chọn trước đó, thói quen xem phim và các đặc tính của phim (thể loại phim chẳng hạn).
- Last.fm cung cấp dịch vụ radio trên internet miễn phí. Hệ tư vấn của Last.fm dựa trên các băng tần mà người sử dụng đã nghe cũng như thói quen nghe đài của họ.
- Ngoài ra, còn có khá nhiều hệ tư vấn của nhiều công ty thuộc nhiều lĩnh vực kinh doanh khác nhau như: Yahoo, YouTube, MovieLens, Morse, Polylens, Gab, Fab, v.v…

Đặc điểm của các tư vấn là mang tính chất cá nhân hóa nghĩa là nó chỉ phù hợp với một số người dùng (hay nhóm người dùng) có cùng một số đặc tính đã được khảo sát trước đó. Điều này cũng phù hợp với thực tế bởi lẽ không thể có được một lời khuyên chung nhất cho mọi đối tượng.

Các hệ tư vấn ngày càng đóng vai trò quan trọng trong việc thúc đẩy phát triển các giao dịch trên mạng. Vai trò đó được thể hiện qua một số chức năng cơ bản sau đây:

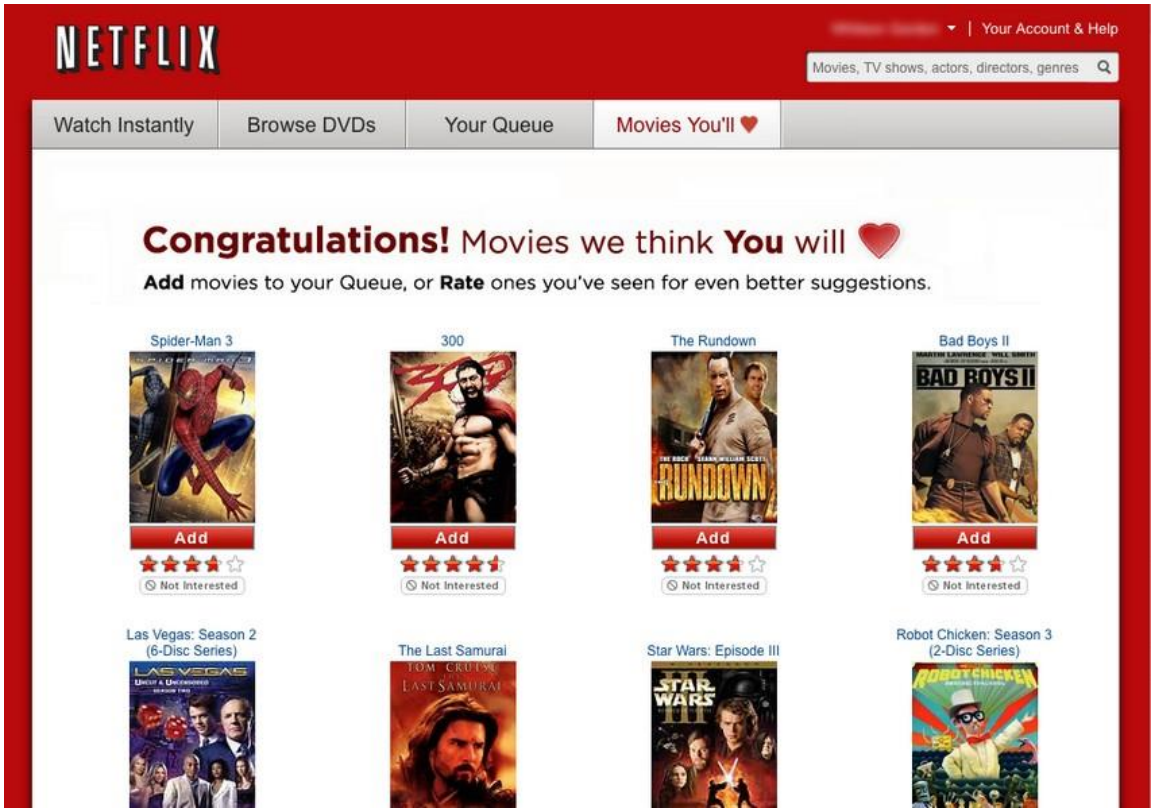
- Gia tăng doanh số bán hàng: nhờ tư vấn mà người kinh doanh có thể bán được một nhóm sản phẩm phù hợp với nhu cầu và thị hiếu của người dùng so với việc không tư vấn chỉ bán được các sản phẩm đơn lẻ.
- Gia tăng thỏa mãn khách hàng: khách hàng càng được thỏa mãn, họ càng gia tăng việc sử dụng các tiện ích khác của công ty (nếu có).
- Tăng độ trung thành của khách hàng.

Đối với người sử dụng, hệ tư vấn mang lại một số lợi ích sau đây:

- Giúp họ giới hạn phạm vi tìm kiếm trong số vô vàn các thông tin thực và ảo.

- Tự tin và quyết đoán hơn khi quyết định mua sản phẩm được tư vấn. Nắm bắt kịp thời các xu hướng sử dụng hiện hành.

Trong hầu hết các trường hợp, bài toán khuyến nghị được coi là bài toán ước lượng trước hạng (rating) của các sản phẩm (phim, cd, nhà hàng …) chưa được người dùng xem xét. Việc ước lượng này thường dựa trên những đánh giá đã có của chính người dùng đó hoặc những người dùng khác. Những sản phẩm có hạng cao nhất sẽ được dùng để khuyến nghị.



Hình 2.1: Ví dụ về một hệ tư vấn trên website Netflix

Bài toán khuyến nghị được mô tả như sau:

Gọi U là tập tất cả người dùng (users); I là tập tất cả các sản phẩm (items) có thể tư vấn. Tập I có thể rất lớn, từ hàng trăm ngàn (sách, CD…) đến hàng triệu (như website). Hàm r (u, i) đo độ phù hợp (hay hạng) của sản phẩm i với user u:

$$r: U \times I \rightarrow R$$

trong đó R là tập các đánh giá (rating) được sắp thứ tự. Với mỗi người dùng $u \in U$, cần tìm sản phẩm $i \in I$ sao cho hàm r (u, i) đạt giá trị lớn nhất.

2.2. HỆ THỐNG TƯ VẤN DU LỊCH

Hệ thống khuyến nghị du lịch (TRS - Tourism Recommender Systems) [1] là một nhánh cụ thể của hệ thống khuyến nghị, có khả năng xử lý dữ liệu input là user profile, đặc điểm cá nhân, sở thích, mối quan hệ cá nhân trong mạng xã hội (social networking), thông tin ngữ cảnh về thời tiết, khí hậu, lễ hội, vị trí địa lý vùng miền, GPS thu được từ nhiều nguồn như internet, mobile phone, facebook, blog, social network,… nhằm đưa ra các gợi ý về chuyến du lịch, lời khuyên về chuyến đi, các điểm quan tâm (POI – Points of Interest), đề nghị về gói du lịch đang có, theo các 3 tiêu chí phù hợp cá nhân về độ hấp dẫn, giá cả, khoảng thời gian, chi phí phù hợp ngân sách cá nhân người dùng.

Thách thức đối với TRS là dữ liệu input liên quan người dùng có tường minh (explicitly) và ngầm định (implicitly) qua xử lý khai thác dữ liệu hoạt động trực tuyến của người dùng (user online activity), cảm xúc, ý kiến người dùng cũng có thay đổi nhanh chóng theo từng thời điểm khác nhau. Sự thay đổi sở thích người dùng có thể thay đổi theo ngữ cảnh, tâm trạng, môi trường kinh tế xã hội, khí hậu, thời tiết.

Các hệ thống TRS hiện tại [1], thường được áp dụng thực tế tại các đơn vị du lịch đạt mức tìm kiếm đưa ra gợi ý cho chuyến du lịch muốn đến dựa trên giới hạn về thời gian, ngân sách, nhu cầu cụ thể, hoặc thông tin user nhập vào. Hệ thống tính tương quan các lựa chọn của user với các điểm đến định sẵn theo các thông số các vector xác định trước.

Cùng với sự phổ dụng của mobile như điện thoại thông minh, thiết bị di động

nhỏ gọn, đã có thể tích hợp sẵn các cảm biến, định vị địa lý, suy diễn user, môi trường xã hội và ngữ cảnh, là điều kiện thuận lợi khai thác thông tin và khá nhiều nghiên cứu vài năm gần đây giới thiệu các kết quả đáng giá khi gợi ý người dùng theo các môi trường ngữ cảnh, đặc trưng cá nhân hóa [1]. Hệ thống cụ thể là VISIT (Mehaan, 2013), EnoSigTur (Simó, 2012), DieToRecs (Bauernfeind, 2003), TripMatcher MePrint (Ricci, 2002), TripAdvisor (Sigala, 2016), TripSay (Gavalas, 2012). Dựa theo các đặc điểm về kiến trúc hệ thống (Web Application, Mobile), mức độ quan tâm và nhu cầu người dùng (User Involvement), tiêu chí nguồn cơ sở khuyến nghị (Deriving Recommendation), các hệ thống này được xếp vào loại hình dịch vụ khác nhau. Dù không tách bạch rõ ràng thấu đáo, nhưng ta có thể phân loại tương đối, các mục tiêu, dịch vụ được cung cấp gồm có 5 loại chính:

1. Đầu tiên là Attractions (POIs) Recommendations, gợi ý điểm quan tâm, điểm đến của một chuyến du lịch.
2. Hai là Tourist Services Recommendations, hệ thống này lọc các thông tin dựa trên ràng buộc để gợi ý thông tin về nhà hàng, khách sạn, giao thông, trung tâm thông tin, chi phí.
3. Ba là Collaborative Filtering Recommendations, loại hệ thống này nhắm đến các gợi ý có tính khám phá, mới lạ vượt mong đợi, ngạc nhiên cho người dùng từ ý kiến, sở thích, nhận định của nhiều người khác chia sẽ. Nguồn dữ liệu xử lý từ kho tàng thông tin cá nhân của mạng xã hội, nơi chia sẽ, kho lưu trữ trên thiết bị của người dùng.
4. Bốn là Routes and Tours Recommendations, dựa trên trích lọc thông tin vị trí có từ thiết bị như GPS, Wi-Fi, cell-id, RFID, hệ thống này gợi ý giúp người dùng đường đi từ điểm hiện tại đến nơi quan tâm.
5. Năm là Personalized Multiple-days Tour Planning, người du lịch có thể muốn tham quan, trải nghiệm tối đa nhiều nơi, nhiều ngày, nhưng do có giới

hạn về thời gian, ngân sách, hệ thống này gợi ý giúp người dùng tham quan chọn các điểm (POIs) hấp dẫn nhất, xứng đáng nhất phù hợp điều kiện cho phép và thông tin cá nhân, sở thích của họ. Ý tưởng này dẫn đến bài toán thiết kế Tour khách du lịch (TTDP, Tourist Trip Design Problem), các giải thuật heuristic hiệu quả được dùng để giải bài toán này cho các ứng dụng trực tuyến vì không thể giải trong thời gian đa thức (Vansteenwegen, 2011). Các nghiên cứu cố gắng đơn giản hóa bài toán TTDP đã được thực hiện như mô hình TTDP đơn giản nhất là bài toán chạy định hướng (OP, Orienteering Problem) được giới thiệu năm 1984 bởi Tsiligirides [1], các mở rộng là bài toán TOP (Team Orienteering Problem) năm 1996 bởi Chao, TOP with Time Windows (TOPTW) bởi Vansteenwegen, 2009, và nghiên cứu gần đây là Timedependent TOPTW bởi Garcia, 2013.

Vài năm gần đây, nhiều hệ thống TRS đã được tích hợp và hoạt động trong các cổng thông tin portal du lịch uy tín, có tên tuổi lớn, khẳng định ý nghĩa thực tiễn của TRS. Ta thấy xuất hiện trong các hệ thống web du lịch nổi tiếng như sau:

1. TripAdvisor (Sigala, 2016), là website du lịch cung cấp tư vấn các chuyến đi chơi, vị trí, hoạt động cho mỗi người, và có một thành phần xã hội (social component) cho phép nhiều yếu tố được reviewed, commented, rated bởi các người dùng khác để trợ giúp quy trình ra quyết định phức tạp liên quan ngành du lịch.
2. DieToRecs (Bauernfeind, 2003), hỗ trợ sự lựa chọn các sản phẩm du lịch (hotel, museum, climbing school) và tạo một giỏ du lịch (travel bag) là một nhóm các sản phẩm du lịch hài hòa nhau.
3. Heracle (Gavalas, 2012), hiện thực content-based filtering dựa trên thông tin du lịch khai phá từ nhiều nguồn dữ liệu online và search engine.

4. TripSay (Gavalas, 2012), dùng phương pháp collaborative filtering để kết hợp điểm đến, nơi, cảnh quan, nội dung và hoạt động, dựa vào mạng kết nối bạn bè người dùng có tham gia như social networking hoặc tương tự.

Các nhu cầu người dùng đa dạng và thay đổi nhanh, dữ liệu xử lý quá nhiều và khắp nơi, môi trường ngữ cảnh có thay đổi, vì vậy để đáp ứng nhu cầu thực tế trong ngành du lịch là một thách thức lớn đối với người làm việc khoa học máy tính cùng hệ thống thông tin, giải thuật phù hợp.

2.3. PHÂN LOẠI HỆ TƯ VẤN

RS có thể được phân thành 6 loại (Gavalas, 2014):

1. Collaborative filtering (Breece, 1998): loại này được dùng nhiều trong ecommerce, social media. Người dùng mục tiêu được gợi ý các món hàng, tiết mục, item tương tự với các thứ được chọn bởi những người khác có tương đồng về sở thích, thị hiếu. Các cá nhân có tương quan (correlate) với mỗi người khác nhau. Về cơ bản, một cặp người có tương quan nhau là thể hiện mức độ tương đồng nhau về cá tính, sở thích, qua sự đánh giá, sự lựa chọn của quá khứ trước đây.

2. Content-based filtering (Pazzani, 1999), hệ thống tư vấn loại này dựa trên nội dung các món hàng, tiết mục mà người dùng mục tiêu đã từng chọn trong các lần giao tác trước đó. Cụ thể là nhiều món hàng đề cử được so sánh với các món được đánh giá (rate) trước đó bởi người dùng, và món hàng phù hợp nhất được chọn đưa ra gợi ý.

3. Knowledged-based filtering (Trewin, 2000), loại này theo hướng dựa trên cơ sở tri thức để tạo một tư vấn bằng cách suy diễn về các món hàng đáp ứng được yêu cầu người dùng (ví dụ một tư vấn về một chiếc xe hơi sẽ xem xét dựa trên tiêu chí nào trọng yếu hơn như tính kinh tế tiết kiệm xăng hay tính

tiện nghi, thoải mái, sang trọng đối với người dùng mục tiêu). Tri thức được xây dựng từ thu thập các lựa chọn, sở thích người dùng, hoặc qua hỏi đáp người dùng để cung cấp thông tin liên quan đến các lựa chọn. Hàm tương tự thể hiện mức độ nhu cầu người dùng tương quan với nội dung của các món hàng tùy chọn, item options. Giá trị của hàm tương tự thường thể hiện mức độ hữu ích của mỗi gợi ý.

4. Demographic filtering (Pazzani, 1999): hệ thống này được dùng nhiều trong ngành marketing để gợi ý món hàng dựa trên dữ liệu nhân khẩu học của user. Dữ liệu này như là số lần xem một món hàng cụ thể liên quan đến vùng miền, ngôn ngữ, tuổi, giới tính.

5. Matrix factorization (Koren, 2008): loại này gồm biến thể của collaborative filtering và dùng thông số đường cơ sở (baseline parameter) cho mỗi user và món hàng. Baseline là các tham số mô hình cộng thêm mô tả cho mỗi user và món hàng, chúng thể hiện độ lệch tổng quát của mức đánh giá (rating) người dùng hay món hàng so với trung bình toàn cục (global average). Ví dụ, đường cơ sở người dùng, user baseline của một người có xu hướng mức đánh giá (rate) cao hơn trung bình dân số sẽ là số dương (positive number).

6. Hybrid RSs (Burke, 2002): loại này dùng kết hợp các phương pháp trên bằng cách khai thác điểm mạnh của kỹ thuật này để bù đắp điểm yếu của cái kia, vì vậy nâng cao hiệu quả hiệu suất tổng thể. Lai ghép hóa có thể được thực hiện bằng nhiều cách, ví dụ tạo dự đoán theo cách content-based và collaborativebased độc lập rồi kết hợp kết quả lại; hoặc thêm khả năng của content-based vào collaborative-based và ngược lại; hợp nhất các phương pháp lại thành một mô hình tổng thể.

Mỗi loại RS có khác nhau về ưu nhược điểm, tùy vào đặc thù ngành nghề, mức độ chính xác mà RS phù hợp được chọn. Trên thực tế có 3 loại RS được quan tâm áp dụng nhiều, và ta đánh giá rõ hơn về ưu nhược điểm như *Bảng 2.1*:

Bảng 2.1: Ưu nhược điểm của 3 loại tư vấn cơ bản

Kỹ thuật	**Dữ liệu nền**	**Quy trình**	**Ưu điểm**	**Nhược điểm**
Collaborative	Đánh giá, bình chọn của users. Không cần thu thập đặc trưng items.	Nhận diện users profile trong hệ thống giống với users mục tiêu.	Có thể gợi ý items đến các người dùng tương đồng trong nhóm. Không gặp trở ngại sở thích của user dù có thay đổi.	Khó khăn với items mới, users mới vì chưa có dữ liệu rating.
Content-based	Các đặc trưng của hàng hóa, items. Dữ liệu đánh giá, bình chọn của users.	Phân lớp các items nhằm làm khớp với bình chọn, hành vi users.	Không trở ngại với items mới. Có thể giới thiệu chính xác items hợp với profile.	Khó khăn với users mới chưa có profile. Khó khăn nếu user có nhiều sở thích đa chiều, trung du. Thiếu ngạc nhiên đột phá vì không thể gợi ý items nằm ngoài user profile.
Knowledged	Đặc trưng items tri thức về tính đáp ứng nhu cầu user của items đó.	Tìm kiếm sự phù hợp giữa người dùng và đặc trưng item.	Không cần thống kê dữ liệu profile người dùng. Thích nghi với sở thích cá tính user nếu có thay đổi.	Kỹ thuật xử lý tri thức phức tạp. Khả năng khuyến nghị là tĩnh lặng (static), không ảnh hưởng trên các tập dữ liệu nhiều item.

CHƯƠNG 3
CƠ SỞ LÝ THUYẾT

3.1. ONTOLOGY

3.1.1. KHÁI NIỆM VỀ ONTOLOGY

Trong khoa học máy tính, một ontology [4] là một mô hình dữ liệu biểu diễn một lĩnh vực và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng. Ontology cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một miền mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán hỗn tạp khác. Ba chức năng hữu ích chính của ontology là:

1. Trợ giúp giao tiếp giữa nhiều người trong dự án.
2. Có khả năng giao tiếp giữa các hệ thống phần mềm.
3. Tăng cường linh hoạt thiết kế và chất lượng của hệ thống phần mềm vì việc xây dựng phát triển ontology có thể độc lập với việc lập trình

Các ontology được sử dụng như là một biểu mẫu trình bày tri thức về thế giới hay một phần của nó. Các ontology thường miêu tả:

1. Các cá thể: Các đối tượng cơ bản, nền tảng
2. Các lớp: Các tập hợp, hay kiểu của các đối tượng
3. Các thuộc tính: Thuộc tính, tính năng, đặc điểm, tính cách, hay các thông số mà các đối tượng có và có thể đem ra chia sẻ.
4. Các mối liên hệ: Các con đường mà các đối tượng có thể liên hệ tới một đối tượng khác.

Bộ từ vựng ontology được xây dựng trên cơ sở tầng RDF và RDFS, cung cấp khả năng biểu diễn ngữ nghĩa mềm dẻo cho tài nguyên Web và có khả năng hỗ trợ lập luận.

3.1.2. CÁC THÀNH PHẦN TRONG ONTOLOGY

Các cá thể (Individuals) - Thể hiện

Các cá thể là các thành phần cơ bản, nền tảng của một ontology. Các cá thể trong một ontology có thể bao gồm các đối tượng cụ thể như con người, động vật, cái bàn… cũng như các cá thể trừu tượng như các thành viên hay các từ. Một ontology có thể không cần bất kỳ một cá thể nào, nhưng một trong những lý do chính của một ontology là để cung cấp một ngữ nghĩa của việc phân lớp các cá thể, mặc dù các cá thể này không thực sự là một phần của ontology.

Các lớp (Classes) - Khái niệm

Các lớp là các nhóm, tập hợp các đối tượng trừu tượng. Chúng có thể chứa các cá thể, các lớp khác, hay là sự phối hợp của cả hai.

Các ontology biến đổi tuỳ thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ, một lớp có thể xếp gộp vào hoặc bị xếp gộp vào bởi các lớp khác. Mối quan hệ xếp gộp này được sử dụng để tạo ra một cấu trúc có thứ bậc các lớp, thường là với một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp rất rõ ràng kiểu 2002, Ford ở phía dưới cùng.

Các thuộc tính (Properties)

Các đối tượng trong ontology có thể được mô tả thông qua việc khai báo các thuộc tính của chúng. Mỗi một thuộc tính đều có tên và giá trị của thuộc tính đó. Các thuộc tính được sử dụng để lưu trữ các thông tin mà đối tượng có thể có. Ví dụ, đối với một cá nhân có thể có các thuộc tính: Họ_tên, ngày_sinh, quê_quán, số_cmnd…

Giá trị của một thuộc tính có thể có các kiểu dữ liệu phức tạp.

Các mối quan hệ (Relation)

Một trong những ứng dụng quan trọng của việc sử dụng các thuộc tính là để mô tả mối liên hệ giữa các đối tượng trong ontology. Một mối quan hệ là một thuộc tính có giá trị là một đối tượng nào đó trong ontology.

Một kiểu quan hệ quan trọng là kiểu quan hệ xếp gộp (subsumption). Kiểu quan hệ này mô tả các đối tượng nào là các thành viên của các lớp nào của các đối tượng.

Hiện tại, việc kết hợp các ontology là một tiến trình được làm phần lớn là thủ công, do vậy rất tốn thời gian và đắt đỏ. Việc sử dụng các ontology là cơ sở để cung cấp một định nghĩa thông dụng của các thuật ngữ cốt lõi có thể làm cho tiến trình này trở nên dễ quản lý hơn. Hiện đang có các nghiên cứu dựa trên các kỹ thuật sản sinh để nối kết các ontology, tuy nhiên lĩnh vực này mới chỉ hiện hữu về mặt lý thuyết.

3.2. NGÔN NGỮ OWL

3.2.1. OWL LÀ GÌ

OWL [9] (The Web Ontology Language) là một ngôn ngữ gần như XML dùng để mô tả các hệ cơ sở tri thức. OWL là một ngôn ngữ đánh dấu dùng để xuất bản và chia sẻ dữ liệu trên Internet thông qua những mô hình dữ liệu gọi là “ontology”. Ontology mô tả một lĩnh vực (domain) và diễn tả những đối tượng trong lĩnh vực đó cùng những mối quan hệ giữa các đối tượng này. OWL là phần mở rộng về từ vựng của RDF và được kế thừa từ ngôn ngữ DAML+OIL Web ontology – một dự án được hỗ trợ bởi W3C. OWL biểu diễn ý nghĩa của các thuật ngữ trong các từ vựng và mối liên hệ giữa các thuật ngữ này để đảm bảo phù hợp với quá trình xử lý bởi các phần mềm.

OWL được xem như là một kỹ thuật trọng yếu để cài đặt cho Semantic Web trong tương lai. OWL được thiết kế đặc biệt để cung cấp một cách thức thông dụng

trong việc xử lý nội dung thông tin của Web. Ngôn ngữ này được kỳ vọng rằng sẽ cho phép các hệ thống máy tính có thể đọc được thay thế cho con người. Vì OWL được viết bởi XML, các thông tin OWL có thể dễ dàng trao đổi giữa các kiểu hệ thống máy tính khác nhau, sử dụng các hệ điều hành và các ngôn ngữ ứng dụng khác nhau. Mục đích chính của OWL là sẽ cung cấp các chuẩn để tạo ra một nền tảng để quản lý tài sản, tích hợp mức doanh nghiệp và để chia sẻ cũng như tái sử dụng dữ liệu trên Web. OWL được phát triển bởi nó có nhiều tiện lợi để biểu diễn ý nghĩa và ngữ nghĩa hơn so với XML, RDF và RDFS, và vì OWL ra đời sau các ngôn ngữ này, nó có khả năng biểu diễn các nội dung mà máy có thể biểu diễn được trên Web.

3.2.2. CÁC PHIÊN BẢN CỦA OWL

Hiện nay có ba loại OWL [9]: OWL Lite, OWL DL (description logic), và OWL Full.

- OWL Lite: hỗ trợ cho những người dùng chủ yếu cần sự phân lớp theo thứ bậc và các ràng buộc đơn giản. Ví dụ: Trong khi nó hỗ trợ các ràng buộc về tập hợp, nó chỉ cho phép tập hợp giá trị của 0 hay 1. Điều này cho phép cung cấp các công cụ hỗ trợ OWL Lite dễ dàng hơn so với các bản khác.
- OWL DL (OWL Description Logic): hỗ trợ cho những người dùng cần sự diễn cảm tối đa trong khi cần duy trình tính tính toán toàn vẹn (tất cả các kết luận phải được đảm bảo để tính toán) và tính quyết định (tất cả các tính toán sẽ kết thúc trong khoảng thời gian hạn chế). OWL DL bao gồm tất cả các cấu trúc của ngôn ngữ OWL, nhưng chúng chỉ có thể được sử dụng với những hạn chế nào đó (Ví dụ: Trong khi một lớp có thể là một lớp con của rất nhiều lớp, một lớp không thể là một thể hiện của một lớp khác). OWL DL cũng được chỉ định theo sự tương ứng với logic mô tả, một lĩnh vực nghiên cứu trong logic đã tạo nên sự thiết lập chính thức của OWL.
- OWL Full muốn đề cập tới những người dùng cần sự diễn cảm tối đa và sự

tự do của RDF mà không cần đảm bảo sự tính toán của các biểu thức. Ví dụ, trong OWL Full, một lớp có thể được xem xét đồng thời như là một tập của các cá thể và như là một cá thể trong chính bản thân nó. OWL Full cho phép một ontology gia cố thêm ý nghiã của các từ vựng được định nghĩa trước (RDF hoặc OWL).

Các phiên bản này tách biệt về các tiện ích khác nhau, OWL Lite là phiên bản dễ hiểu nhất và phức tạp nhất là OWL Full.

Mối liên hệ giữa các ngôn ngữ con của OWL:

- Mọi ontology hợp lệ dựa trên OWL Lite đều là ontology hợp lệ trên OWL DL.
- Mọi ontology hợp lệ dựa trên OWL DL đều là ontology hợp lệ trên OWL Full.
- Mọi kết luận hợp lệ dựa trên OWL Lite đều là kết luận hợp lệ trên OWL DL.
- Mọi kết luận hợp lệ dựa trên OWL DL đều là kết luận hợp lệ trên OWL Full.

3.3. NGÔN NGỮ SPARQL

3.3.1. KHÁI NIỆM SPARQL

SPARQL [5] là một ngôn ngữ truy vấn RDF, tên của nó là một từ viết tắt của giao thức SPARQL và ngôn ngữ truy vấn RDF (A viết tắt đệ quy). SPARQL được tạo ra là một chuẩn để truy cập dữ liệu RDF làm việc theo nhóm (DAWG) của World Wide Web Consortium, và được coi là một trong những công nghệ chủ chốt của web semantic. Ngày 15 tháng 2008, SPARQL đã trở thành một khuyến cáo chính thức của W3C [5].

SPARQL cho phép cho một truy vấn bao gồm mô hình ba mẫu, liên từ, sự phân cách (disjunctions), và các mô hình mẫu tùy chọn.

RDF là một hướng dẫn, chứa các nhãn định dạng dữ liệu đồ thị để biểu diễn

thông tin trong trang Web. RDF thường được sử dụng để đại diện cho nhiều thứ như thông tin cá nhân, mạng xã hội, metadata về kỹ thuật số cũng như cung cấp một nguồn tích hợp dữ liệu trên các nguồn khác nhau của thông tin. Và SPARQL là ngôn ngữ truy vấn để định nghĩa cú pháp và ngữ nghĩa cho RDF.

3.3.2. CÁC KIỂU TRUY VẤN

Ngôn ngữ SPARQL [5] đặc tả 4 loại truy vấn khác nhau cho các mục đích khác nhau:

- Truy vấn SELECT: Sử dụng để trích xuất các giá trị thô từ SPARQL endpoint, các kết quả được trả về trong một định dạng bảng.
- Truy vấn CONSTRUCT: Sử dụng để trích xuất thông tin từ SPARQL endpoint và chuyển kết quả thành dạng RDF hợp lệ.
- Truy vấn ASK: Sử dụng để cung cấp các kết quả dạng True/False đơn giản cho các truy vấn trên SPARQL endpoint.
- Truy vấn DESCRIBE: Sử dụng để trích xuất một đồ thị RDF từ SPARQL endpoint, các nội dung đó được đưa tới endpoint để quyết định dựa trên những thông tin có ích.

Mỗi dạng truy vấn đều dùng khối lệnh bên trong từ khóa WHERE để hạn chế truy vấn mặc dù trường hợp truy vấn DESCRIBE từ khóa WHERE là tùy chọn.

3.3.3. VÍ DỤ

Một ví dụ về truy vấn SPARQL các mô hình câu hỏi "tất cả các thủ đô quốc gia ở châu Phi là gì?":

```
PREFIX abc: <http://example.com/exampleOntology#>

SELECT ?capital ?country

WHERE {
```

```
  ?x abc:cityname ?capital;
     abc:isCapitalOf ?y.
  ?y abc:countryname ?country;
     abc:isInContinent abc:Africa.
}
```

Các biến được chỉ định bởi tiền tố "?" hoặc "$". Truy vấn rằng buộc các dữ liệu là ?capital và ?country sau đó trả về kết quả.

Để thực hiện các truy vấn ngắn gọn, SPARQL cho phép định nghĩa các tiền tố và các URI cơ sở trong một cách như Turtle. Trong truy vấn này, các tiền tố "abc" là viết tắt của đường dẫn "http://example.com/exampleOntology #".

3.4. NGÔN NGỮ JAVA

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch hư Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy.

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương

trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lí bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sở dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

3.5. ANDROID

3.5.1. KHÁI NIỆM ANDROID

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Googlevà sau này được chính Google mua lại vào năm 2005.Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache.Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa

trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốcdành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

3.5.2. GIAO DIỆN

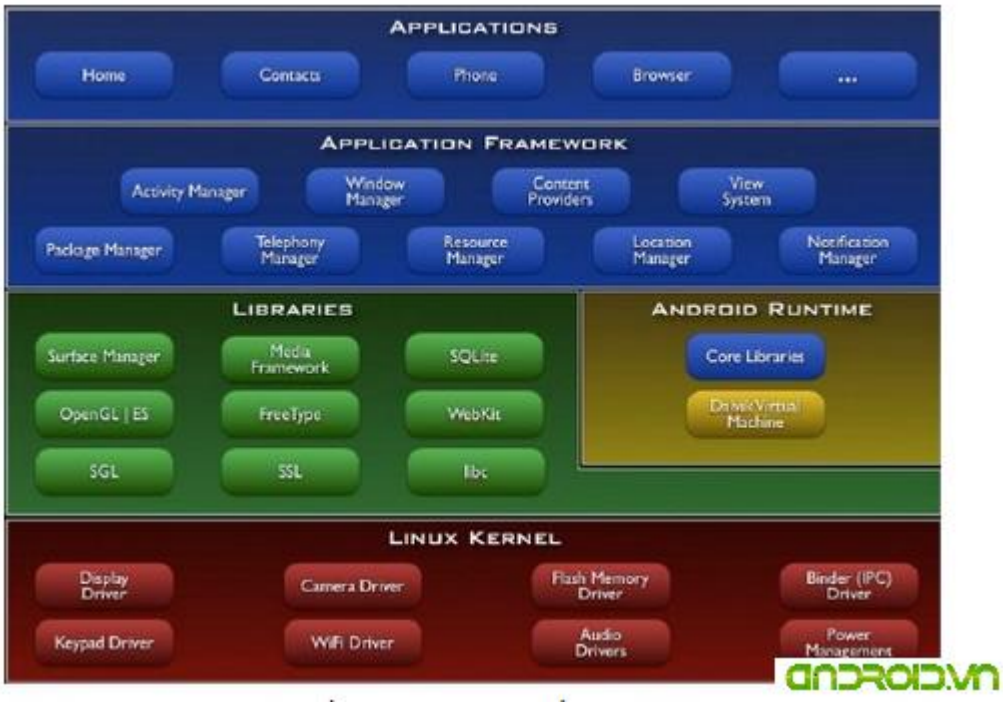
Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo dãn và thu lại để xử lý các đối tượng trên màn hình.

Hình 3.1: Màn hình chính Android 6.0 Marshmallow

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop trên máy tính để bàn. Màn hính chính Android thường gồm nhiều biểu tượng (*icon*) và tiện ích

(*widget*). Giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích.

3.5.3. KIẾN TRÚC ANDROID



Hình 3.2: Sơ đồ về kiến trúc của Android

Kiến trúc android chia làm 4 tầng ứng dụng:

Tầng 1: Tầng Application là tầng ở trên cùng cách xa với phần cứng nhất: Chứa các ứng dụng mà lập trình viên phát triển như: browser, Contacts, media.

Tầng 2: Application Framework

- Activity Manager - quản lý vòng đời của các ứng dụng.
- Windows Manager - quản lý form của các ứng dụng.
- Content Providers - cho phép các ứng dụng truy cập dữ liệu từ các ứng dụng khác hoặc để chia sẻ dữ liệu của riêng ứng dụng
- Google xây dựng cho các developer để phát triển các ứng dụng của họ trên

Android chỉ bằng cách gọi các API.

- View UI - để xây dựng layout của ứng dụng bao gồm: list view, text field, button, dialog, form …
- Resource Manager - cung cấp cách thức truy cập đến non-code resources như các asset, graphic, image, music, video …
- Notification Manager - cho phép tất cả các ứng dụng hiển thị thông báo của mình trên hệ điều hành.

Tầng 3: Libraries

Là các thư viện được viết bằng ngôn ngữ C/C++ sẽ được các developer phát triển ứng dụng android thông qua tầng Android Framework. Có thể kể ra đây một số thư viện quen thuộc với các lập trình viên như:

- Media Libraries – mở rộng từ PacketVideo's OpenCORE. Hỗ trợ nhiều định dạng video và image phổ biến: MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- Surface Manager – quản lý việc hiển thị và kết hợp đồ họa 2D và 3D.
- LibWebCore – dùng webkit engine cho việc render trình duyệt mặc định của HDH Android browser và cho dạng web nhúng (như HTML nhúng)
- OpenGL|ES – thư viện đồ họa 2D và 3D
- SQLite – quản lý database của ứng dụng
- Runtime Android
- gồm một tập hợp các thư viện Java Core.
- Máy ảo Dalvik thực thi các file định dạng .dex (Dalvik Excutable)
- Mỗi ứng dụng Android chạy trên tiến trình riêng của máy ảo Dalvik. Dalvik được viết để chạy nhiều máy ảo cùng một lúc một cách hiệu quả trên cùng một thiết bị.

Tầng 4: Kernel Linux layer

Dựa trên Kernel Linux version 2.6 bởi nó cung cấp các trình điều khiển các thiết bị phần cứng(driver), quản lý tiến trình, quản lý tài nguyên, bảo mật,... như sau:

- Security system
- Memory management
- Process management
- Network stack
- Driver model.

3.6. DỊCH VỤ WEB (WEB SERVICE)

3.6.1. GIỚI THIỆU DỊCH VỤ WEB

Theo định nghĩa của W3C (World Wide Web Consortium), Dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. XML là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một Dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ Dịch vụ Web. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Trước hết, có thể nói rằng ứng dụng cơ bản của là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này, các ứng dụng cần được tích hợp với cơ sở dữ liệu (CSDL) và các ứng dụng khác, người sử dụng sẽ giao tiếp với CSDL để tiến hành phân tích và lấy dữ liệu. Trong thời gian gần đây, việc phát triển mạnh mẽ của thương mại điện tử và B2B cũng đòi hỏi các hệ thống phải có khả năng tích hợp với CSDL của các đối tác kinh doanh (nghĩa là tương tác với hệ thống bên ngoài – bên cạnh tương tác với các thành phần bên trong của hệ thống

trong doanh nghiệp).

3.6.2. ĐẶC ĐIỂM DỊCH VỤ WEB (WEBSERVICE)

Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau. Ví dụ, đặt Web server cho ứng dụng trên một máy chủ chạy hệ điều hành Linux trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể chạy và xử lý bình thường mà không cần thêm yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.

Phần lớn kĩ thuật của được xây dựng dựa trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.

Một Dịch vụ Web bao gồm có nhiều mô-đun và có thể công bố lên mạng Internet.

Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra những lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp khác và cả những cá nhân thông qua mạng Internet.

Một ứng dụng khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía server ví dụ như PHP, Oracle Application server hay Microsoft.Net…

Ngày nay Dịch vụ Web đang rất phát triển, những lĩnh vực trong cuộc sống có thể áp dụng và tích hợp là khá rộng lớn như dịch vụ chọn lọc và phân loại tin tức (hệ thống thư viện có kết nối đến web portal để tìm kiếm các thông tin cần thiết); ứng dụng cho các dịch vụ du lịch (cung cấp giá vé, thông tin về địa điểm…), các đại lý bán hàng qua mạng, thông tin thương mại như giá cả, tỷ giá hối đoái, đấu giá qua mạng…hay dịch vụ giao dịch trực tuyến (cho cả B2B và B2C) như đặt vé máy bay, thông tin thuê xe…

Các ứng dụng có tích hợp đã không còn là xa lạ, đặc biệt trong điều kiện thương

mại điện tử đang bùng nổ và phát triển không ngừng cùng với sự lớn mạnh của Internet. Bất kì một lĩnh vực nào trong cuộc sống cũng có thể tích hợp với Dịch vụ Web, đây là cách thức kinh doanh và làm việc có hiệu quả bởi thời đại ngày nay là thời đại của truyền thông và trao đổi thông tin qua mạng. Do vậy, việc phát triển và tích hợp các ứng dụng với Dịch vụ Web đang được quan tâm phát triển là điều hoàn toàn dễ hiểu.

3.6.3. KIẾN TRÚC DỊCH VỤ WEB (WEBSERVICE)

Hình 3.3: Kiến trúc của dịch vụ web

Web service provider (bên cung cấp dịch vụ)

Web service consumer (bên sử dụng dịch vụ)

Web service broker (bên môi giới dịch vụ)

Ba thành phần kể trên tương tác với nhau bởi ba cơ chế, đó là:

- **Service**: là cơ chế cho phép client xác định và triệu gọicác dịch vụ từ xa thông qua mạng mà không phụ thuộcvào vị trí địa lí, hệ điều hành sử dụng hay ngôn ngữ lập trình được sử dụng.

- **Message**: là phương tiện giao tiếp giữa bên cung cấp dịch vụ và bên sử dụng dịch vụ. Một message có thể làmột yêu cầu từ bên sử dụng dịch vụ gửi đến bên cung cấp dịch vụ hay là một phản hồi từ bên cung cấp dịch vụ về cho bên sử dụng dịch vụ. Các message này được định nghĩa bằng ngôn ngữ đánh dấu độc lập nền tảng là XML.
- **Dynamic discovery**: là cơ chế được cài đặt dựa trên directory service. Về phía bên cung cấp, chúng sẽ sử dụng directory service để tự đăng kí những dịch vụ mà chúng cung cấp. Còn về phía bên sử dụng, chúng sẽ truy vấn để tìm ra các dịch vụ theo nhu cầu từ directory service thông qua mạng. Điều này làm giảm sự lệ thuộc của bên sử dụng dịch vụ vào bên cung cấp dịch vụ.
- **Publish** (xuất bản): để có thể truy cập được thì một Dịch vụ Web cần phải được công bố (mô tả) để các Service consumer có thể tìm thấy nó. Việc công bố có thể khác nhau tùy thuộc vào từng ứng dụng cụ thể. Nhưng thông thường, một mô tả dịch vụ (service description) bao gồm các thông tin sau: các interface, các kiểu dữ liệu,các toán tử, các thông tin kết nối, vị trí của dịch vụ có thể truy cập được trên mạng, siêu dữ liệu, v.v…
- **Find** (tìm kiếm): trong thao tác tìm kiếm, Service consumer sẽ lấy mô tả về dịch vụ đang được yêu cầu một cách trực tiếp hoặc thông qua Service broker. Thao tác tìm kiếm này có thể diễn ra trong hai pha vòng đời của một Dịch vụ Web consumer, đó là pha thiết kế xây dựng (lập trình viên cần biết mô tả, interface của dịch vụ) và pha thực thi (xác định vị trí và tiến hành triệu gọi dịch vụ).
- **Bind** (triệu gọi): để sử dụng được dịch vụ thì cần phải triệu gọi nó. Trong thao tác bind, Dịch vụ Web consumer khi thực thi sẽ gọi hoặc khởi tạo một luồng tương tácvới dịch vụ dựa trên các thông tin trong mô tả dịch vụ mà nó thu được trước đó như: vị trí dịch vụ, cách liên lạc và tương tác với dịch vụ,…

3.7. CÔNG CỤ PROTÉGÉ

3.7.1. GIỚI THIỆU PROTÉGÉ

Protégé là công cụ phần mềm biên tập ontology mã nguồn mở (được phát triển tại Trường ĐH Stanford) [8] sử dụng đối với việc xây dựng các hệ thống thông minh. Protégé được hỗ trợ bởi cộng đồng lớn bao gồm: các viện nghiên cứu, các tổ chức chính phủ và những người sử dụng cộng tác. Các đơn vị, cá nhân này sử dụng Protégé để xây dựng các giải pháp dựa trên tri thức trong các lĩnh vực chuyên sâu như là: y sinh học, thương mại điện tử và mô hình hóa tổ chức.

Protégé hiện bao gồm 02 phiên bản:

- Phiên bản chạy trên Web (web-based), người sử dụng có thể thực hiện việc biên tập, xây dựng ontology trực tiếp tại địa chỉ: http://protege.stanford.edu/products.php#web-protege. Ngoài ra người dùng cũng có thể tải bộ cài đặt phiên bản web về để cài trên Webserver của mình và sử dụng trong mạng nội bộ, đường dẫn tải bộ cài và hướng dẫn cài đặt tại: https://github.com/protegeproject/webprotege/releases.
- Phiên bản chạy trên Desktop (desktop-based): phiên bản này với mục đích biên tập ontology trên máy tính cá nhân, có thể tải về tại địa chỉ: http://protege.stanford.edu/products.php#desktop-protege.

3.7.2. CÁC ĐẶC ĐIỂM CỦA PROTÉGÉ

Hỗ trợ đầy đủ ba phiên bản của ngôn ngữ OWL là OWL-Full, OWL-Lite và OWL-DL.

Nhờ sử dụng mô hình hướng đối tượng của ngôn ngữ Java, Protégé tỏ ra rất hiệu quả trong việc mô hình các lớp, thực thể, quan hệ..

Giao diện thiết kế trực quan có tính tương tác cao. Người sử dụng có thể định nghĩa các thành phần của Ontology trực tiếp từ các form.

Cho phép biểu diễn trực quan Ontology dưới dạng các sơ đồ.

Cho phép xây dựng Ontology từ nhiều nguồn khác nhau.

Protégé tự động lưu một bản tạm của Ontology. Nếu có lỗi phát sinh trong quá trình thao tác thì Ontology cũ sẽ tự động được phục hồi. Người thiết kế cũng có thể chuyển qua lại giữa hai bản Ontology này bằng chức năng Revert to a Previous Version và ActiveCurrent Version.

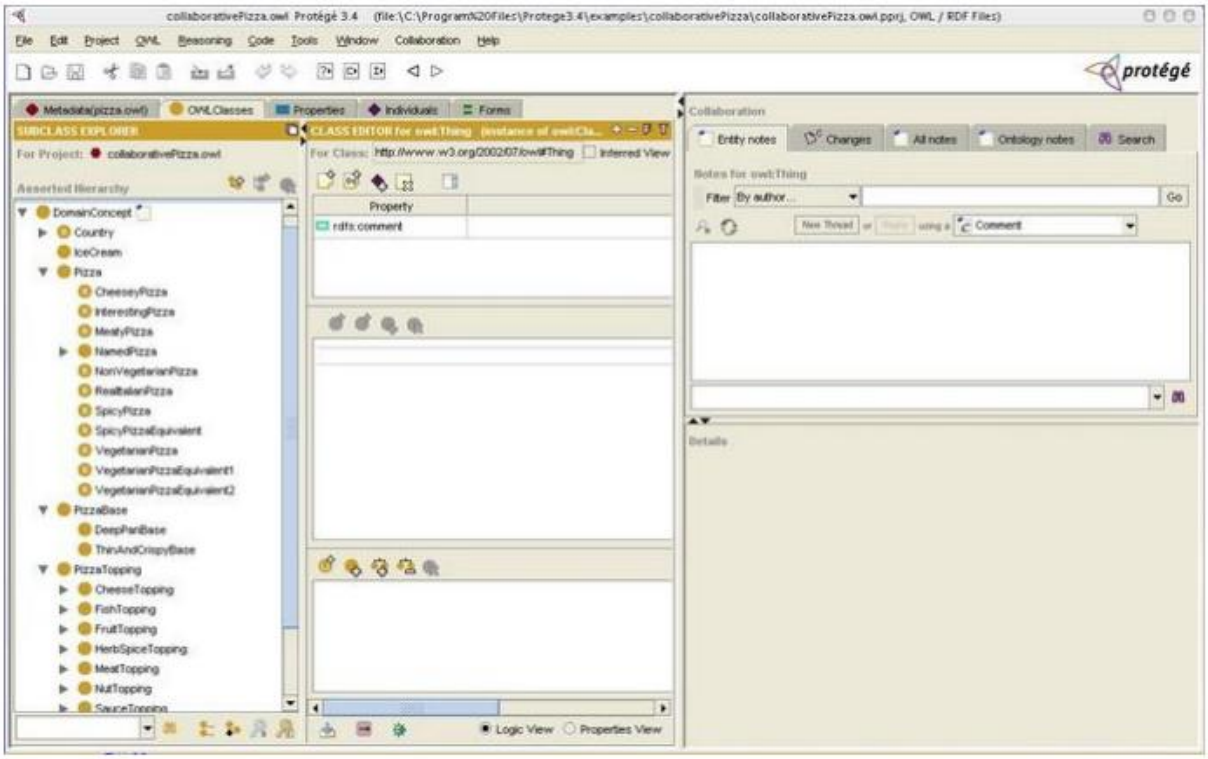
Cung cấp chức năng tìm kiếm lỗi, kiểm tra tính nhất quán và đầy đủ của Ontology. Để sử dụng, người thiết kế chọn chức năng Run Ontology Test và Check Consistency.

Cho phép các lớp và thuộc tính của Ontology này có thể được sử dụng trong một Namespace khác mà chỉ cần sử dụng các URL để tham khảo. Để sử dụng, chọn chức năng Move Resource to Namespace.

Hỗ trợ suy luận trực tiếp trên Ontology dựa trên Interface chuẩn DL Implementation Group (DIG).

Hỗ trợ sinh mã tự động. Protégé cho phép chuyển Ontology thành mã nguồn RDF/XML, OWL, DIG, Java, EMF Java Interfaces, Java Schema Classes. Các mã này có thể được nhúng trực tiếp vào ứng dụng và là đầu vào cho các thao tác trên Ontology khi cần.

Cung cấp đầy đủ chuẩn giao tiếp cho các Plug-in.



Hình 3.4: Giao diện Protégé 3.4

3.8. JSON

3.8.1. JSON LÀ GÌ

JSON (JavaScript Object Noattion) là một định dạng hoán vị dữ liệu nhanh. Chúng dễ dàng cho chúng ta đọc và viết. Dễ dàng cho thiết bị phân tích và phát sinh. Chúng là cơ sở dựa trên tập hợp của Ngôn Ngữ Lập Trình JavaScript, tiêu chuẩn ECMA-262 phiên bản 3 - tháng 12 năm 1999. JSON là một định dạng kiểu text mà hoàn toàn độc lập với các ngôn ngữ hoàn chỉnh, thuộc họ hàng với các ngôn ngữ họ hàng C, gồm có C, C++, C#, Java, JavaScript, Perl, Python, và nhiều ngôn ngữ khác. Những đặc tính đó đã tạo nên JSON 1 ngôn ngữ hoán vị dữ liệu lý tưởng.

3.8.2. CẤU TRÚC JSON

JSON được xây dựng trên 2 cấu trúc:

- Là tập hợp của các cặp tên và giá trị name-value. Trong những ngôn ngữ khác nhau, đây được nhận thấy như là 1 đối tượng (object), sự ghi (record), cấu trúc (struct), từ điển (dictionary), bảng băm (hash table), danh sách khoá (keyed list), hay mảng liên hợp.
- Là 1 tập hợp các giá trị đã được sắp xếp. Trong hầu hết các ngôn ngữ, nó được nhận thấy như là 1 mảng, véc tơ, tập hợp hay là 1 dãy sequence.

Đây là 1 cấu trúc dữ liệu phổ dụng. Hầu như tất cả các ngôn ngữ lập trình hiện đại đều hỗ trợ chúng trong một hình thức nào đó. Chúng tạo nên ý nghĩa của một định dạng hoán vị dữ liệu với các ngôn ngữ lập trình cũng đã được cơ sở hoá trên cấu trúc này.

Ví dụ 1:

```
[
   {
      "name": "Nguyễn Văn A",
      "age": "21 tuổi"
   },
   {
      "name": "Trần Văn C",
      "age": "22 tuổi"
   },
   {
      "name": "Nguyễn Văn Chính",
      "age": "23 tuổi"
   }
]
```

Ví dụ 2:

```
{
   "sv0001": {
      "toan": "MônToán",
      "ly": "MônLý"
   },
   "sv0002": {
      "toan": "MônToán",
      "anh": "Môn Anh"
   }
}
```

CHƯƠNG 4
CÁC GIẢI THUẬT

4.1. GIẢI THUẬT TÌM KIẾM TOUR

4.1.1. CÁC THUẬT NGỮ

4.1.1.1. THUỘC TÍNH TOUR

Để đánh giá mức độ quan tâm của người dùngvới một Tour. Ngoài các thông tin cơ bản của Tour như giá tiền, địa điểm xuất phát, thời gian Tour, thì còn có các thôngtin như đặc điểm địa lý, khí hậu, hoạt động, người đi với,… từ đó có thể đánh giá chính xác một Tour so với mục đích, yêu cầu của người dùng.

Qua quá trình đánh giá, khảo sát và được tư vấn từ người làm trong lĩnh vực du lịch. Thuộc tính của một Tour được phân chia thành các nhóm như *Bảng 4.1*:

Bảng 4.1: Bảng các thuộc tính Tour

STT	Tên nhóm	Thuộc tính trong nhóm
1	Giá Tour	Dưới 1 triệu; Từ 1 đến 2 triệu; Từ 2 đến 4 triệu; Từ 4 đến 6 triệu; Từ 6 đến 10 triệu; Trên 10 triệu
2	Thời gian Tour	1 buổi; 1 ngày; 1-2 ngày; 2-4 ngày; Trên 4 ngày
3	Cự ly tuyến	Nội thành; Ngoại thành; Khu vực lân cận; Cùng vùng; Khác vùng
4	Khí hậu	Ấm áp, Mát mẻ, Lạnh
5	Đặc điểm địa lý	Đồng bằng; Núi; Biển; Sông nước; Hang động; Khu bảo tồn; Vườn quốc gia; Thôn quê; Đô thị
6	Loại hình du lịch	Nghỉ dưỡng; Sinh thái; Tham quan; Vui chơi – giải trí; Tâm linh; Văn hóa – nghệ thuật; Tình nguyện; Thực tế
7	Hoạt động	Spa; Mua sắm; Cắm trại; Leo núi; Ẩm thực; Ngắm cảnh; Thể thao; Trò chơi

8	Người đi với	Một mình; Gia đình; Người yêu; Bạn bè; Đồng nghiệp; Tổ chức – đoàn thể

4.1.1.2. TRỌNG SỐ

Để đánh giá mức độ quan tâm của người dùng đối với từng thuộc tính trong Tour như giá tiền, người đi chung, hoạt động… thuộc tính nào được người dùng quan tâm hơn. Trọng số được dùng để đánh giá điều đó.

Mỗi kiểu người dùng sẽ có điểm trọng số khác nhau ứng với từng thuộc tính mà họ quan tâm. Hiện tại có 3 thông tin để phân loại người dùng như *Bảng 4.2*:

Bảng 4.2: Bảng phân loại người dùng

STT	Nhóm	Phân loại trong nhóm
1	Công việc	Nhóm thu nhập cao (bác sĩ, kỹ sư, thương nhân,.)
		Nhóm thu nhập trung bình (giáo viên, công nhân viên chức)
		Nhóm thu nhập thấp (Công nhân, học sinh- sinh viên)
2	Độ tuổi	Dưới 30 tuổi
		Từ 30 đến 50 tuổi
		Trên 50 tuổi
3	Giới tính	Nam
		Nữ

Mỗi nhóm người sẽ gồm: Công việc, độ tuổi, giới tính. Từ đó xác định trọng số ứng với từng nhóm người

4.1.1.3. ĐIỂM TƯƠNG TÁC

Thông qua khảo sát đánh giá, ta có thể xác định được các thuộc tính có liên quan, ảnh hưởng đến nhau. Ví dụ: Người lựa chọn là sẽ đi cùng người yêu thì tỉ lệ cao là các có thuộc tính lãng mạn cao sẽ thu hút họ. Bằng cách này có thể xác định được

độ quan tâm của người dùng với nhiều thuộc tính khác nhau chỉ từ một lựa chọn của người dùng.

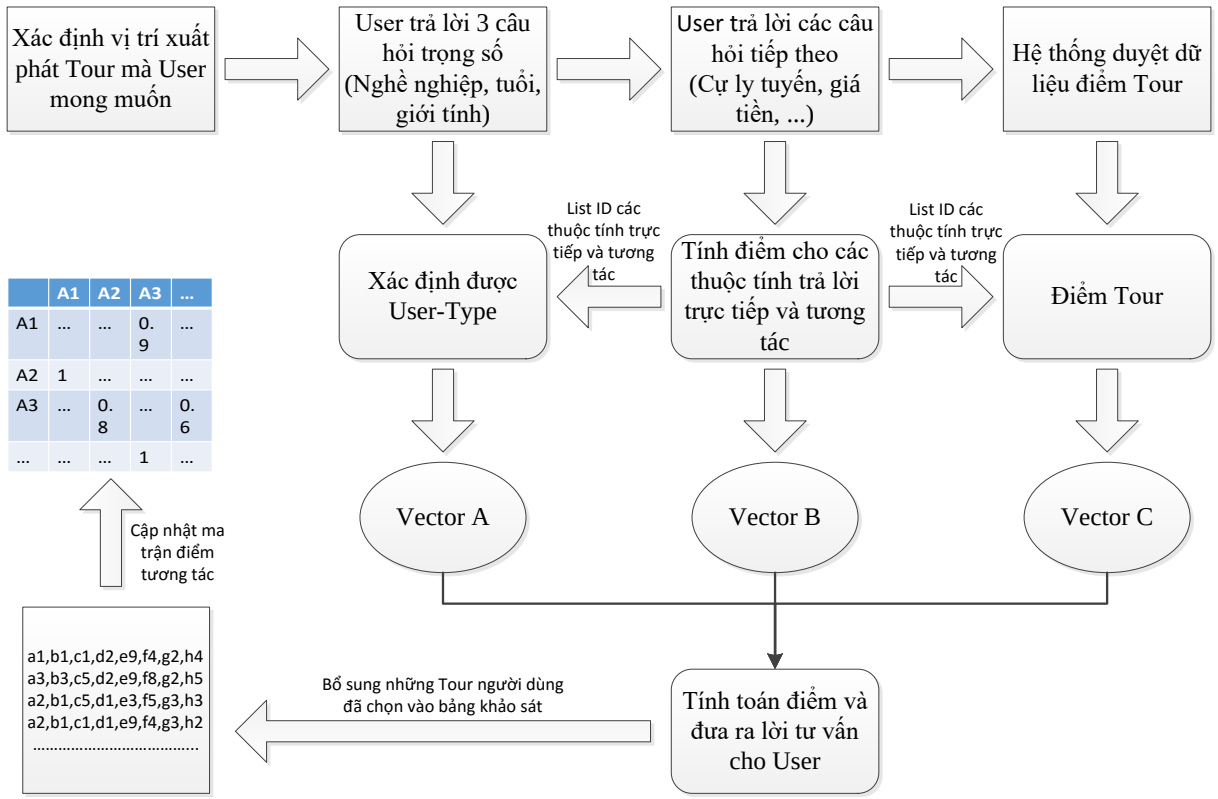
Điểm tương tác có giá trị từ 0 đến 1 (Từ có tương tác thấp đến có tương tác cao) như *Bảng 4.3*:

Bảng 4.3: Ví dụ bảng điểm tương tác các thuộc tính

Thuộc tính 1	Thuộc tính 2	Điểm
1	2	0.7
1	4	0.8
1	5	0.4
1	9	0.7
1	3	0.05
1	4	1.0
1	13	0.35
5	9	0.2
6	12	0.85
9	12	0.6

4.1.2. GIẢI THUẬT TÌM KIẾM TOUR

Giải thuật tìm kiếm Tour là giải thuật tính điểm Tour, đưa ra danh sách Tour gần với nhu cầu người dùng nhất dựa vào nhưng thông tin của người dùng cung cấp. Giải thuật này tôi thừa kế lại của một nhóm đã hiện thực.



	A1	A2	A3	...
A1	...	...	0.9	...
A2	1	...	...	...
A3	...	0.8	...	0.6
...	...	...	1	...

Hình 4.1: Sơ đồ giải thuật tìm kiếm Tour

- **Vector A:** vector trọng số thuộc tính User, thể hiện mức độ quan tâm của User đối với mỗi nhóm thuộc tính
- **Vector B:** chứa giá trị là điểm thuộc tính có được sau khi tính toán trực tiếp và tương tác thuộc tính của User
- **Vector C:** chứa điểm đã được đánh giá, khảo sát thực tế về 1 Tour cụ thể, được lưu sẵn trong Database.

Xác định vector A:

Vector A là hệ số factor thể hiện mức độ quan tâm về các nhóm thuộc tính đối với User. Giá trị vector A có được từ kết quả trả lời trực tiếp của User tại 3 câu hỏi bắt buộc: Q2, Q3, Q4.

Ví dụ: Giả sử với mỗi câu hỏi có các lựa chọn trả lời tương ứng như sau:

Q2: Bạn làm nghề gì ?

➔ Lựa chọn, option, trả lời:

J1, group 1: PM, CEO, dầu khí, ngân hàng, thương gia, thương mại, xuất nhập khẩu,...

→ nhóm thu nhập cao

J2, group 2: Cán bộ, Nhân viên, chuyên viên, kỹ thuật, cơ khí, hóa chất,...

→ nhóm thu nhập trung bình

J3, group 3: Buôn bán nhỏ, công nhân, sinh viên, tự do, thất nghiệp, ...

→ nhóm thu nhập thấp

Q3: Bạn bao nhiêu tuổi ?

➔ Lựa chọn, option, trả lời:

A1, group 1: Dưới 30 tuổi

→ nhóm thu nhập cao

A2, group 2: Từ 30 đến 50 tuổi

→ nhóm thu nhập trung bình

A3, group 3: Trên 50 tuổi

→ nhóm thu nhập thấp

Q4: Giới tính của bạn ?

➔ Lựa chọn, option, trả lời:

G1, group 1: Nam

G2, group 2: Nữ

Bảng 4.4: Bảng dữ liệu trọng số nhóm thuộc tính

Trọng số	Nghề nghiệp			Tuổi			Giới tính	
	J1	J2	J3	A1	A2	A3	G1	G2
Length of Tour	1	2	3	3	2	1	2	1

Climate	1	1	1	1	2	3	1	3
Price	1	2	3	3	1	2	1	2
Distance	2	1	3	1	1	3	3	2
Traveling	2	1	1	1	1	2	1	2
Type	1	1	1	1	1	1	1	1
Interested in	2	3	1	***3***	2	1	2	***3***
Specific Features	2	1	1	3	1	1	2	2

Khi thực hiện hỏi đáp User 3 câu hỏi Q2, Q3, Q4, ta sẽ có được kết quả option trả lời của User. Tiến hành tra với dữ liệu có sẵn ở bảng 1, ta có được kết quả trọng số của User đang quan tâm như *Bảng 4.5*.

Thông tin User: Nhân viên CNTT, 35 tuổi, Nam

Bảng 4.5: Kết quả Factor của User sau hỏi đáp Q2-3-4

Trọng số	Job	Age	Gender	Kết quả
	J2	A2	G1	
Length of Tour	2	2	2	2
Climate	1	2	1	2
Price	2	1	1	2
Distance	1	1	3	3
Traveling	1	1	1	1
Type	1	1	1	1
Interested in	3	2	2	3
Specific Features	1	1	2	2

Xác định vector B:

Vector B là giá trị điểm các thuộc tính của User. Các giá trị điểm thuộc tính User có được qua giải thuật tính điểm trực tiếp từ hỏi đáp Q5 → Q12, và tự tính toán khi kết hợp với giá trị ma trận tương tác thuộc tính.

Ví dụ: Bảng ma trận tương tác của các thuộc tính có được từ giá trị có sẵn theo kinh nghiệm ngành du lịch như *Hình 4.2*.

		Dưới 1 triệ	Từ 1 đế	Từ 2 đ	Từ 4 đế	Trên 10 t	1 buổi	1 ngày	1-2 n	Trên	Nội	Khu	Cùng vù	Khác vù	Khí hậu	Ấm áp	Mát n	Lạnh
		A1	A2	A3	A4	A6	B1	B2	B3	B5	C1	C3	C4	C5	D	D1	D2	D3
Dưới 1 triệu	A1				0.4				0.5							1		
Từ 1 đến 2 triệu	A2						0.5	0.5			1	1						
Từ 2 đến 4 triệu	A3																	0.4
Từ 4 đến 6 triệu	A4									0.6			1	1				
Từ 6 đến 10 triệu	A5									0.8			0.8	1				
1 buổi	B1														1			
1 ngày	B2	1	0.8														0.7	
1-2 ngày	B3		1	0.8														
Trên 4 ngày	B5					1								1				
Nội thành	C1																	
Ngoại thành	C2	0.8	0.9					1										
Cùng vùng	C4																	
Khí hậu	D															1		
Ấm áp	D1																	
Núi	E2																	
Sông	E4																	
Hồ	E5																	
Thành thị	E9																	
Sinh thái	F2																	
Du lịch tâm linh	F6																	
Du lịch tình ngu	F7																	
Du lịch thực thể	F8																	
Spa	G1										1							
Ngắm cảnh	G6																	
Người yêu	H2										0							
Tổ chức-đoàn t	H5	1																

Hình 4.2: Giá trị tương tác các thuộc tính Tour

Xác định vector C:

Vector C là vector chứa điểm thực tế của Tour theo các thuộc tính đã xác định như User. Mức độ quan tâm của User đối với từng Attribute cũng sẽ phản ánh tương đương đối với Attribute đó trên vector C.

4.1.3. CHIẾN LƯỢC ĐẶT CÂU HỎI

4.1.3.1. TÍNH ĐIỂM CÁC THUỘC TÍNH TỪ ĐIỂM TƯƠNG TÁC

Từ câu trả lời của người dùng ta xác định điểm thuộc tính mà người dùng quan tâm. Lúc này mỗi thuộc tính sẽ gồm 2 thông tin đó là điểm và cấp độ (Level) :

- Cấp 1: Thuộc tính được người dùng lựa chọn. Điểm ở thuộc tính này luôn bằng 100
- Cấp 2: Thuộc tính được tính dựa trên thuộc tính cấp 1
- Cấp 3: Thuộc tính được tính dựa trên thuộc tính cấp 2
- …….

- Cấp x: Thuộc tính được tính dựa trên thuộc tính cấp (x-1)

Mức độ ưu tiên của các cấp thuộc tính là: Cấp 1 > Cấp 2 > Cấp 3 > … > Cấp x

Cách tính điểm thuộc tính dựa trên tương tác

Gọi :

- Point(x): Điểm của thuộc tính x
- Interaction(y,x): Điểm tương tác của thuộc tính y đến thuộc tính x

Trường hợp 1: Thuộc tính chịu tương tác từ một thuộc tính khác:

$$\text{Point(x)} = \text{Point(y)}*\text{Interaction(y,x)} \qquad \textbf{(4.1.1)}$$

Trường hợp 2: Thuộc tính chịu tương tác từ nhiều thuộc tính có cùng cấp độ (cùng level)

$$\text{Point(x)} = \text{Point(x)} = \frac{Interaction(y,x)*Point(y)+Interaction(z,x)*Pont(z)}{Interaction(y,x)+Interaction(z,x)} \qquad \textbf{(4.1.2)}$$

Ví dụ:

$$\text{A}(90,\ 3) \xrightarrow{0.5} \text{C}(45,\ 4)$$
$$\text{B}(60,\ 3) \xrightarrow{0.7} \text{C}(42,\ 4)$$
$$Point\,(C) = \frac{0{,}5*90+0{,}7*60}{0{,}5+0{,}7} = 72{,}5$$

Trường hợp 3: Thuộc tính chịu tương tác từ nhiều thuộc tính không cùng cấp độ (khác level). Chỉ tính điểm từ thuộc tính có cấp độ ưu tiên cao hơn. Áp dụng như trường hợp 1 và trường hợp 2 từ thuộc tính cấp độ cao vừa tìm được.

Ví dụ 1:

$$\text{A}(100,\ 1) \xrightarrow{0.5} \text{C}(50,\ 3) \longrightarrow \text{Chọn}$$
$$\text{B}(80,\ 2) \xrightarrow{0.7} \text{C}(56,\ 3) \longrightarrow \text{Không chọn}$$

Ví dụ 2:

$$A(90, 2) \xrightarrow{0.5} C(45, 3) \longrightarrow \text{Chọn}$$
$$B(60, 3) \xrightarrow{0.7} C(42, 4) \longrightarrow \text{Không chọn}$$

4.1.3.2. LỰA CHỌN CÂU HỎI TIẾP THEO

Khi đi tư vấn tại các trung tâm cung cấp Tour du lịch, người đại lý sẽ hỏi các câu hỏi nhằm lấy được nhiều thông tin người dùng nhất, từ đó sẽ đưa ra Tour phù hợp với khách hàng. Mục tiêu của chiến thuật này cũng vậy, sẽ đưa ra những câu hỏi chưa có thông tin và ưu tiên câu hỏi mang lại nhiều thông tin nhất.

Giả sử, hệ thống có được thông tin người dùng từ câu trả lời dưới dạng vector như sau:

Bảng 4.6: Bảng mô tả vector thông tin của người dùng

	A2	**A3**	**A4**	**B2**	**B3**	**B4**	**C2**	**C3**	**C4**	**D2**	**D3**	**D4**
Vector	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null

Cơ chế hoạt động của giải thuật lựa chọn câu hỏi tiếp theo như sau:

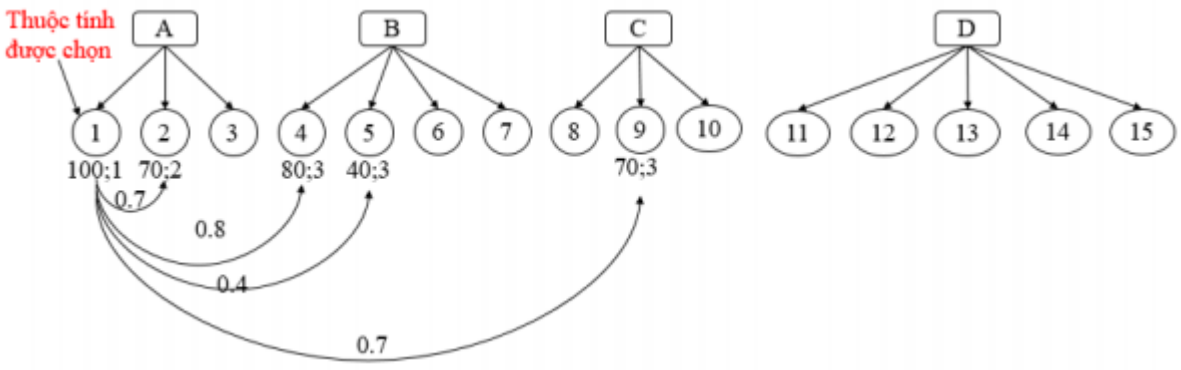
- Xác định thuộc tính người dùng lựa chọn
- Áp dụng công thức tương tác, tính các thuộc tính có liên quan với thuộc tính vừa tìm được
- Áp dụng chiến thuật tham lam, lọc ra các thuộc tính chưa có điểm số, ưu tiên nhóm các thuộc tính chưa có điểm số nhiều nhất. Trường hợp có hai nhóm đều có cùng số thuộc tính chưa tính điểm, lúc đó ta xét các số lượng tương tác đến các thuộc tính khác có được trong đó. Ưu tiên các thuộc tính có nhiều tương tác nhất.
- Đưa ra câu hỏi liên quan đến thuộc tính vừa tìm được.

Ví dụ: Giả sử ta có bảng điểm thuộc tính như *Bảng 4.7* (Thuộc tính 1 tương tác với thuộc tính 2):

Bảng 4.7: Ví dụ bảng điểm tương tác các thuộc tính

Thuộc tính 1	Thuộc tính 2	Điểm
1	2	0.7
1	4	0.8
1	5	0.4
1	9	0.7
1	3	0.05
1	4	1.0
1	13	0.35
5	9	0.2
6	12	0.85
9	12	0.6

- Đầu tiên ta giả sử câu hỏi đầu tiên được hiển thị là câu hỏi A, và đáp án ta lựa chọn là đáp án 1. Lúc này thuộc tính 1 có điểm số là 100 và có level là 1.
- Sau đó từ thuộc tính được chọn, ta tính điểm của các thuộc tính mà thuộc tính level 1tương tác đến (ở đây là các thuộc tính 2,4,5,9). Nếu thuộc tính trong cùng nhóm câu hỏi thì sẽ có level 2, còn thuộc tính không cùng câu hỏi sẽ có level 3.

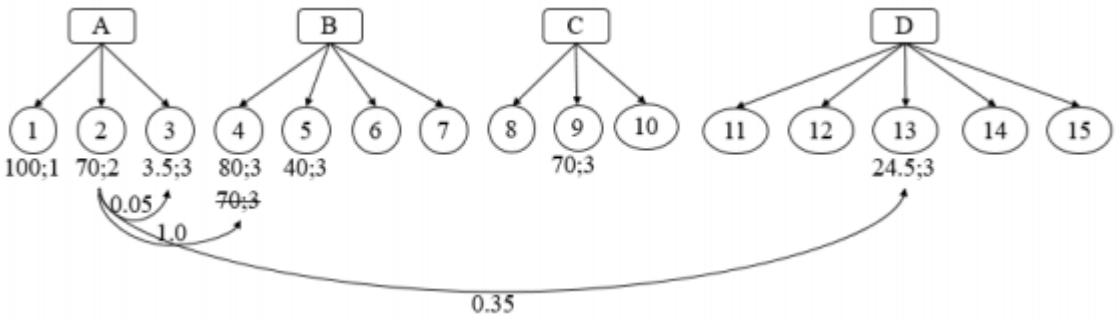


Hình 4.3: Điểm của các thuộc tính mà thuộc tính level 1 tương tác đến

Bảng 4.8: Điểm của các thuộc tính mà thuộc tính level 1 tương tác đến

A			B				C			D				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100;1	70;2		80;3	40;3				70;3						

- Từ các thuộc tính level 2 (ở đây là thuộc tính 2), ta tính các thuộc tính mà nó tương tác đến (ở đây là thuộc tính 3,4,13). Tất cả các thuộc tính được tương tác từ level 2 đều sẽ có level 3.



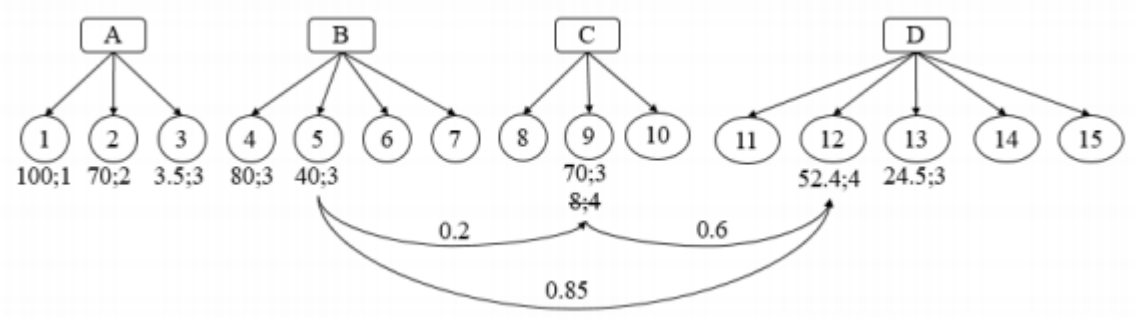
Hình 4.4: Điểm của các thuộc tính mà thuộc tính level 2 tương tác đến

- Thuộc tính 4 do chịu tương tác từ thuộc tính 1 (level 1) và thuộc tính 2 (level 2) đều được level 3, tuy nhiên ở đây ta ưu tiên chọn thuộc tính tương tác tới nócólevel thấp hơn (ở đây là thuộc tính 1) nên tương tác từ thuộc tính 2 không bị ảnh hưởng.

Bảng 4.9: Điểm của các thuộc tính mà thuộc tính level 2 tương tác đến

A			B				C			D				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100;1	70;2	3.5;3	80;3	40;3				70;3				24.5;3		

- Từ các thuộc tính level 3 (ở đây là thuộc tính 3,4,5,9,13), ta tính các thuộc tính mà nó tương tác đến (ở đây là thuộc tính 9,12). Tất cả các thuộc tính được tương táctừ level 3 đều sẽ có level 4.



Hình 4.5: Điểm của các thuộc tính mà thuộc tính level 3 tương tác đến

- Thuộc tính 9 cùng chịu tương tác từ thuộc tính 1 và thuộc tính 5, tuy nhiên ở đâyta ưu tiên chọn thuộc tính tương tác làm cho nó có level thấp hơn (ở đây là thuộc tính 1) nên tương tác từ thuộc tính 5 không bị ảnh hưởng.
- Thuộc tính 12 cùng chịu tương tác từ thuộc tính 5 và thuộc tính 9 (cùng level là 3) và đều tác động cho nó thành level 4 nên ta sẽ áp dụng tính điểm theo công thức.

Bảng 4.10: Điểm của các thuộc tính mà thuộc tính level 3 tương tác đến

A			B				C			D				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100;1	70;2	3.5;3	80;3	40;3				70;3			52.4;4	24.5;3		

- Nếu còn tương tác thì tiếp tục tính cho các level tiếp theo.
- Sau khi đã hoàn thành việc tính toán, ta xác định câu hỏi tiếp theo bằng việc xét xem câu hỏi nào còn nhiều thuộc tính chưa tính điểm nhất để làm câu hỏi tiếp theo (ở đây là câu hỏi D còn 3 thuộc tính chưa tính điểm là nhiều nhất nên sẽ chọn làm câu hỏi tiếp theo).
- Trong trường hợp có 2 câu hỏi cùng có số thuộc tính chưa tính điểm nhiều nhất thìta sẽ chọn câu hỏi nào có nhiều tương tác hơn. Ví dụ câu hỏi B và C có cùng sốthuộc tính chưa tính điểm là nhiều nhất thì ta sẽ xét số tương tác, ở đây câu hỏi B có 2 tương tác, câu hỏi C có 1 tương tác nên ta sẽ chọn câu

hỏi tiếp theo là B.

4.1.3.3. ĐƯA RA LỰA CHỌN PHÙ HỢP

Khi đưa ra câu hỏi cho người dùng, ngoài lựa chọn câu hỏi tiếp theo thì câu hỏi đưa ra phải đảm bảo phù hợp với các câu trả lời trước của người dùng. Ví dụ: Một người lựa chọn đi Nội thành Tp. Hồ Chí Mình, thì đến câu hỏi để xác định người dùng muốn đi một Tour có những đặc điểm gì không nên đưa ra các lựa chọn như Núi, vườn Quốc Gia. Với tiêu chí như trên ta đưa ra chiến thuật lựa chọn thuộc tính phù hợp để xuất ra trong câu hỏi.

Cơ chế hoạt động:

- Lấy thuộc tính người dùng lựa chọn từ các câu hỏi trước
- Lọc trong cơ sở dữ liệu các Tour có điểm số của thuộc tính đó cao nhất
- Áp dụng chiến thuật lựa chọn câu hỏi tiếp theo
- Mỗi lựa chọn của câu hỏi tương ứng với từng thuộc tính trong hệ thống. Kiểm tra thuộc tính này trong các Tour vừa tìm lọc được. Nếu có điểm số lớn hơn 50 thì lựa chọn liên quan đến thuộc tính đó phù hợp để xuất hiện ở câu hỏi tiếp theo

Ví dụ minh họa:

Người dùng lựa chọn là các Tour xuất phát từ Tp. Hồ Chí Minh

Câu hỏi bắt đầu: "Thời gian Tour": Người dùng lựa chọn Tour đi từ 1 đến 2 ngày

	GIÁ TOUR						THỜI GIAN TOUR					Cự ly tuyến					KHÍ HẬU		
	< 1 triệu	1-2 triệu	2-4 triệu	4-6 triệu	6-10 triệu	>10 triệu	1 buổi	1 ngày	1-2 ngày	2-4 ngày	> 4 ngày	Nội thành	Ngoại thành	Lân cận	Cùng vùng	Khác vùng	Ấm áp	Mát mẻ	Lạnh
SGN09	100	70	40	10	0	0	40	70	100	70	40	40	70	100	70	10	100	50	0
LON01	70	100	70	40	10	0	10	40	100	70	40	40	70	100	70	10	50	50	0
SGN12	40	70	100	70	40	10	40	70	100	70	40	40	70	100	70	40	100	30	0
TAY04	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	0	100	50	0
SGN27	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	70	100	0	0
SGN29	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	70	30	100	0
SGN13	70	100	70	40	10	0	40	70	100	70	40	0	10	40	70	100	100	20	0
TNI02	100	70	40	10	0	0	70	100	70	40	10	0	100	100	50	0	100	30	0

Hình 4.6: Người dùng lựa chọn thời gian Tour

Câu hỏi thứ 2:

Áp dụng chiến thuật lựa chọn câu hỏi tiếp theo. Giả sử là câu hỏi "Cự ly tuyến"

	GIÁ TOUR						THỜI GIAN TOUR					Cự ly tuyến					KHÍ HẬU		
	< 1 triệu	1-2 triệu	2-4 triệu	4-6 triệu	6-10 triệu	>10 triệu	1 buổi	1 ngày	1-2 ngày	2-4 ngày	> 4 ngày	Nội thành	Ngoại thành	Khu vực lân cận	Cùng vùng	Khác vùng	Ấm áp	Mát mẻ	Lạnh
SGN09	100	70	40	10	0	0	40	70	100	70	40	40	70	100	70	10	100	50	0
LON01	70	100	70	40	10	0	10	40	100	70	40	40	70	100	70	10	50	50	0
SGN12	40	70	100	70	40	10	40	70	100	70	40	40	70	100	70	40	100	30	0
TAY04	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	0	100	50	0
SGN27	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	70	100	0	0
SGN29	70	100	70	40	10	0	40	70	100	70	40	10	40	70	100	70	30	100	0
SGN13	70	100	70	40	10	0	40	70	100	70	40	0	10	40	70	100	100	20	0
TNI02	100	70	40	10	0	0	70	100	70	40	10	0	100	100	50	0	100	30	0

Hình 4.7: Người dùng lựa chọn cự ly tuyến

Dựa vào bảng thống kê ta thấy với thời gian đi 1-2 ngày thì Tour chỉ hỗ trợ đi các nơi là: Khu vực lân cận, ngoài vùng, khác vùng (Điểm số 100).

Với thang điểm của Tour là 0 đến 100, thì các thuộc tính có điểm số lớn hơn 50 cũng có nổi bật. Có thể đưa lựa chọn liên quan đến thuộc tính này vào câu hỏi. Điểm số của nội thành trong danh sách này cao nhất là 70 (lớn hơn 50)

Lúc này câu hỏi cự lý tuyến, sẽ gồm các lựa chọn:

- Ngoại thành
- Khu vực lân cận
- Cùng vùng
- Khác vùng

Tương tự câu hỏi 1: Lấy thuộc tính người dùng lựa chọn, chọn các Tour có điểm số thuộc tính này cao nhất. Áp dụng tương tự với các câu hỏi còn lại

Nhận xét: Ưu tiên được câu trả lời trước, lọc được các lựa chọn cho câu hỏi tiếp theo. Việc lựa chọn các thuộc tính có điểm số lớn hơn 50: Giúp cho người dùng có nhiều sự lựa chọn hơn, nhưng vẫn giữ được tính hợp lý so với các câu trả lời trước của người dùng.

4.2. THUẬT TOÁN APRIORI

4.2.1. GIỚI THIỆU

Các công ty bán lẻ hiện nay phải lưu một số lượng dữ liệu bán hàng khổng lồ. Một mẫu tin trong cơ sở dữ liệu này có thể gồm: ngày thực hiện việc mua bán, các loại hàng hóa mua bán trong giao dịch này,… Từ cở sở dữ liệu này cần tìm các mối quan hệ giữa các thuộc tính, ví dụ như: 95% khách hàng mua áo sơ mi thì thường mua quần tây. Do đó các công ty thành công thường tìm kiếm những luật như vậy (được gọi là luật kết hợp) để biết được tiến trình của thị trường và từ đó đưa ra những chương trình và chiến lược kinh doanh phù hợp.

Luật kết hợp là cấu trúc thông dụng trong việc biểu diễn tri thức và đặc biệt phổ biến trong các hệ tri thức. Ưu điểm chính của luật là tính phù hợp và dễ đọc. Các luật có thể bổ sung hoặc loại bỏ một cách dễ dàng, do tri thức được biểu diễn bằng các luật là có khả năng thích hợp cao.

4.2.2. BÀI TOÁN TÌM LUẬT KẾT HỢP

Gọi I = $\{A_1, A_2, A_3, \ldots, A_n\}$ là tập hợp các trường, còn gọi là các mục dữ liệu(items). Trong đó A_i, (i=1. .n) chỉ lấy giá trị 0 hoặc 1.

D là tập các giao tác (Transaction), trong đó mỗi giao tác T chứa một tập con các item $T \subseteq I$ và mỗi giao tác có một giá trị định danh là TID.

Ta gọi giao tác T chứa X nếu $T \subseteq I$, với X là tập vài item trong I.

Tỷ lệ phần trăm giữa số lượng các giao tác chứa tập item X trên tổng số các giao tác trong D được gọi là độ hỗ trợ (supp) của tập item đó, tức là:

$$supp(X) = \frac{Card(X)}{Card(D)}$$

Luật kết hợp là mối liên hệ điều kiện giữa 2 tập con các mục dữ liệu X và Y theo dạng if X then Y, và ký hiệu là $X \Rightarrow Y$.

Ta có luật kết hợp X ⇒ Y, nếu:

$$X \subset I,\ Y \subset I \text{ và } X \cap Y = \phi$$

Luật X ⇒ Y có độ hỗ trợ (supp) là s nếu có s% số giao tác trong D chứa X ∪ Y hay:

$$supp(X => Y) = s = \frac{Card(X \cup Y)}{Card(D)}$$

Luật X ⇒ Y có độ tin cậy (conf) là c, nếu có c% số giao tác trong D chứa X ∪ Y so với số giao tác trong D chứa X, tức là:

$$conf(X => Y) = c = \frac{Card(X \cup Y)}{Card(X)}$$

Ví dụ: D là cơ sở dữ liệu đơn đặt hàng với tập I = {A_1, A_2, A_3, A_4, A_5, A_6} biểu diễn việc đặt hàng các sản phẩm A_1, A2, A_3, A_4, A_5, A_6.

Bảng 4.11: Ví dụ bảng cơ sở dữ liệu đơn đặt hàng

TID	A1	A2	A3	A4	A5	A6
1	1	1	0	0	0	1
2	1	1	0	1	1	1
3	1	0	1	1	1	1
4	0	0	1	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	1	1

Với dữ liệu trên ta có:

Với X={A_1, A_2}: số hàng có giá trị tại A_1 và A_2 = 3

$$supp(X) = (3/6) = 0.5$$

Với Y=A6:

$$supp(X \Rightarrow Y) = supp(X \cup Y) = (3/6) = 0.5$$

$$conf(X \Rightarrow Y) = (3/3) = 1$$

Với một tập các giao tác D, bài toán tìm luật kết hợp là bài toán tìm tất cả các luật có độ hỗ trợ và độ tin cậy lớn hơn các ngưỡng cho trước tương ứng là độ hỗ trợ nhỏ nhất (minsupp) và độ tin cậy nhỏ nhất (minconf) do người dùng xác định.

Bài toán tìm luật kết hợp được chia thành hai bài toán con sau:

- Tìm tất cả item (gọi là itemSet) có độ hỗ trợ lớn hơn hoặc bằng minsupp. Khi đó các tập này được gọi là tập phổ biến (Large itemSet).
- Sử dụng các tập phổ biến để phát sinh các luật kết hợp có độ tin cậy lớn hơn hoặc bằng minconf. Với mỗi tập phổ biếnL và A là một tập con khác rỗng của P, nếu tỷ lệ (%) giữa supp(P) so với supp(A) lớn hơn bằng minconf thì ta có luật kết hợp A $\Rightarrow$ (P\A)

4.2.3. THUẬT TOÁN TÌM LUẬT KẾT HỢP

Trong lĩnh vực khám phá tri thức trên cơ sở dữ liệu chuỗi thời gian có thể sử dụng nhiều kỹ thuật khám phá luật khác nhau trên cơ sở dữ liệu đã qua giai đoạn tiền xử lý. Trong luận văn này sẽ trình bày thuật toán khám phá luật kết hợp Apriori.

Luật kết hợp là một dạng rất phổ biến trong việc biểu diễn tri thức. Luật kết hợp có ưu điểm dễ đọc, có thể bổ sung hay loại bỏ dễ dàng. Đây cũng là nội dung quan trọng trong khám khá tri thức từ cơ sở dữ liệu lớn. Biểu diễn tri thức dưới dạng luật kết hợp có thể thích ứng với nhiều ứng dụng khác nhau. Dạng tổng quát của luật kết hợp

là: Nếu X thì Y, trong đó X là giả thiết và Y là kết luận. Các luật kết hợp như vậy mô tả thật dễ hiểu nên nó hỗ trợ tích cực cho các nhà chuyên môn quản lý trong việc ra quyết định hay hoạch định chiến lược phát triển

Nguyên tắc Apriori

- Đếm số lượng của từng Item, tìm các Item xuất hiện nhiều nhất.
- Tìm các cặp ứng viên: Đếm các cặp => cặp item xuất hiện nhiều nhất.
- Tìm các bộ ba ứng viên: Đếm các bộ ba => bộ ba item xuất hiện nhiều nhất. Và tiếp tục với bộ 4, bộ 5, …
- Nguyên tắc chủ yếu: Mọi tập con của tập phổ biến phải là tập con phổ biến.

Giải thuật Apriori

- **Input:**
  - Tập tất cả các thuộc tính trong cơ sở dữ liệu.
  - Giá trị minsup ∈ [0;1] là ngưỡng giá trị hỗ trợ tối thiểu.
  - Giá trị minconf ∈ [0;1] là ngưỡng gí trị tin cậy tối thiểu
- **Output:** là tất cả các tập phổ biến thỏa mãn điều kiện minsup, minconf và điểm tương tác giữa các thuộc tính trong từng tập phổ biến đó.

Sơ đồ giải thuật Apriori:

Bước 1: Tính độ hỗ trợ cho mỗi item sau đó so sánh độ hỗ trợ cho mỗi item với minsup, từ đó ta chọn ra các item là tập phổ biến.

Bước 2: Bắt đầu từ tập hạt giống là các tập phổ biến đã tìm ở trên, phát sinh các tập phổ biến mới gọi là tập ItemSet ứng viên (C) và tính độ hỗ trợ cho mỗi tập C trên cơ sở dữ liệusau đó so sánh độ hỗ trợ cho mỗi tập C với minsup, từ đó chọn ra các tập C là tập phổ biến thực sự và dùng làm hạt giống cho bước kế tiếp.

Bước 3: Lặp lại bước 2 ở trên cho đến khi không còn tìm được tập phổ biến nào nữa.

4.2.4. VÍ DỤ

Ta có bảng dữ liệu nhị phân, mỗi vị trí là một giá trị 0 hoặc 1, với 1 là giá trị được lựa chọn và 0 là giá trị không được lựa chọn.

		A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5
1	A lot of cheap stuff			1				1	1				1				
2	Of country-specific brand				1	1		1	1				1				
3	Local ingredients			1	1			1	1				1		1		
4	The famous chef				1	1		1					1		1		
5	Topic of restaurants		1	1				1					1				
6	Local food		1	1				1					1				
7	Sport, etc.		1	1				1	1				1	1	1		
8	Massage, etc.	1	1					1					1				
9	Mountaineering	1						1	1							1	1
10	Experience of local culture		1	1					1	1			1	1	1		
11	Touch with animals	1	1					1								1	

Với các giá trị của các thuộc tính tương ứng như sau:

A	Giá Tour	B	Thời gian Tour	C	Cự lý tuyến
A1	Dưới 1 triệu	B1	1 buổi	C1	Nội thành
A2	Từ 1 đến 2 triệu	B2	1 ngày	C2	Ngoại thành
A3	Từ 2 đến 4 triệu	B3	1-2 ngày	C3	Khu vực lân cận
A4	Từ 4 đến 6 triệu	B4	2-4 ngày	C4	Cùng vùng
A5	Từ 6 đến 10 triệu	B5	Trên 4 ngày	C5	Khác vùng
A6	Trên 10 triệu				

Ví dụ:

- Với giá trị “A lot of cheap stuff” (mua sắm đồ giá rẻ) có tập giá trị là: A3, B1, B2,C1 là các giá trị được lựa chọn xuất hiện cùng nhau (giá trị là 1).
- Với giá trị “Of country-specific brand” (mua sắm hàng hiệu của quốc gia) có tập giá trị là: A4, A5, B1, B2, C1 là các giá trị được lựa chọn xuất hiện cùng nhau.

- Với giá trị "Local ingredients" (mua sắm đồ địa phương) có tập giá trị là: A3, A5,B1, B2, C1, C3 là các giá trị được lựa chọn xuất hiện cùng nhau.
- ...

Bước 1: Từ bảng dữ liệu, ta có: L1 =A1, A2, A3, A4, A5, A6, B1, B2, B3, B4, B5, C1, C2, C3, C4, C5 Ta tính được supp(X) của từng phần tử trong L1 tương ứng như

	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5
fr(X)	3	6	6	3	2	0	10	6	1	0	0	9	2	4	2	1
sup(X)	=3/11 ≈0.27	0.55	0.55	0.27	0.18	0	0.91	0.55	0.09	0	0	0.82	0.18	0.36	0.18	0.09

sau:

Bước 2: So sánh các giá trị vừa tìm được với minsup, ta được tập thỏa điều kiện supp(X) ≥ minsup, ta được tập như sau: A2, A3, B1, B2, C1.

Bước 3: Lấy tổ hợp của tậpvừa tìm được, ta được tập 2-itemset: L2 = A2-A3, A2-B1, A2-B2, A2-C1, A3-B1, A3-B2,A3-C1, B1-B2, B1-C1, B2-C1.

Bước 4: Tính supp(X ∪ Y) của từng phần tử trong L2 và so sánh với minsup:

	A2_A3	A2_B1	A2_B2	A2_C1	A3_B1	A3_B2	A3_C1	B1_B2	B1_C1	B2_C1
fr(X)	4	5	2	5	5	4	6	5	8	5
sup(X)	=4/11 ≈0.36	0.45	0.18	0.45	0.45	0.36	0.55	0.45	0.73	0.45

So sánh với minsup, ta được tập thỏa điều kiện là: A3-C1, B1-C1.

Bước 5: Tính độ tin cậy conf(X=>Y) của các tập thỏa điều kiện ở bước 4, ta được:

X_Y	conf(X → Y)	conf(Y → X)
A3_C1	1	0.67
B1_C1	0.8	0.89

Bước 6: So sánh với minconf, nếu thỏa điều kiện conf(X=>Y) ≥ minconf thì ta được danh sách các luật cần tìm.

conf(A3=>C1) = 1

conf(C1=>A3) = 0.67

conf(B1=>C1) = 0.8

conf(C1=>B1) = 0.89

4.3. KỸ THUẬT MATRIX FACTORIZATION

4.3.1. MÔ HÌNH PHÂN RÃ MA TRẬN

Về cơ bản, phân rã ma trận đặc trưng cho cả người dùng và hạng mục bởi các vector các yếu tố được suy diễn từ các mô hình đánh giá hạng mục. Sự tương đồng cao giữa các yếu tố người dùng và hạng mục sẽ tạo ra tư vấn. Phương pháp này trở nên phổ biến trong những năm gần đây do xử lý tốt dữ liệu có kích thước lớn và cho ra các khuyến nghị chính xác, cũng như tính linh động khi tạo mô hình cho nhiều tình huống trong đời sống thực.

Dữ liệu đầu vào cho các hệ tư vấn có nhiều kiểu và thường được biểu diễn trong một ma trận 2 chiều với một chiều biểu diễn cho các người dùng và chiều còn lại là các hạng mục được người dùng quan tâm. Có 2 phương pháp thu thập dữ liệu. Thu thập dữ liệu trực tiếp qua thông tin phản hồi trực tiếp (*explicit feedback*) từ phía người dùng về quan tâm của họ dành cho sản phẩm. Các quan tâm này thường được đặc trưng bởi các con số được gọi là điểm đánh giá. Ma trận điểm đánh giá từ thu thập trực tiếp thường là các ma trận thưa vì bất kỳ một người dùng nào cũng thường có xu hướng chỉ đánh giá một tỉ lệ nhỏ các hạng mục có sẵn. Trường hợp không thu thập dữ liệu trực tiếp được, hệ tư vấn sẽ phải thu thập dữ liệu gián tiếp (*implicit feedback*) để suy diễn ra các quan tâm của người dùng bằng cách quan sát thái độ của người dùng trong quá khứ từ các dữ liệu lịch sử đi mua, duyệt các dữ liệu này để tìm kiếm các mô hình, … Dữ liệu gián

tiếp thường là có hoặc không có một sự kiện nào đó nên ma trận dữ liệu là một ma trận dầy đặc. Điểm mạnh của phương pháp phân rã ma trận là khả năng cho phép kết hợp thông tin bổ sung, tránh hiện tượng ramp-up cho hệ tư vấn.

Giả sử ta có mỗi người dùng đã cho điểm đánh giá cho một số hạng mục trong hệ thống, ta sẽ dự báo xem các người dùng sẽ cho điểm đánh giá như thế nào cho các hạng mục mà họ chưa đánh giá, nghĩa là ta sẽ tư vấn (dự báo) cho các người dùng này đánh giá các hạng mục chưa được họ đánh giá. Các thông tin đánh giá sẽ được lưu trữ trong một ma trận. *Bảng 4.12* là một ví dụ về ma trận điểm đánh giá gồm có 5 người dùng và 4 hạng mục.

Tác vụ dự báo được xem như là công việc lấp đầy cho ma trận. Ta có thể khám phá các đặc tính (yếu tố) tiềm ẩn qua việc dự báo điểm đánh giá mà một người dùng nào đó đánh giá một hạng mục nào đó, vì các đặc tính mà người dùng quan tâm sẽ trùng khớp các đặc tính của hạng mục đó.

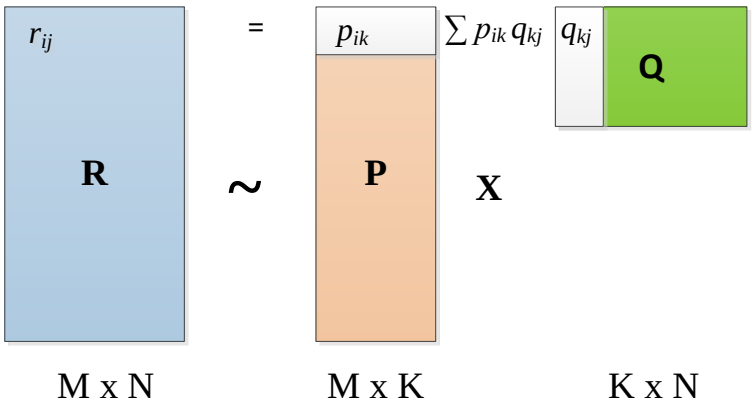
Bảng 4.12: Ma trận điểm đánh giá

	D1	D2	D3	D4
U1	5	3	-	1
U2	4	-	-	1
U3	1	1	-	5
U4	1	-	-	4

Dấu (-) có nghĩa là người dùng U_i chưa cho điểm đánh giá hạng mục D_j.

Phân rã ma trận ánh xạ người dùng và hạng mục sang một không gian có hướng *f* các yếu tố tiềm ẩn kết hợp với nhau, nghĩa là tương quan người dùng-hạng mục được mô hình thành tích vô hướng bên trong không gian đó. Đặt *U* là tập các người dùng, *D* là tập các hạng mục, khi đó ma trận ***R*** có kích thước |*U*| x |*D*| là ma trận chứa tất cả các điểm đánh giá thu thập được mà các người dùng đã đánh giá các hạng mục.

Ý tưởng chính của mô hình phân rã ma trận trong xây dựng hệ tư vấn là xem ma trận điểm đánh giá ***R*** là kết quả nhân từ 2 ma trận có hạng nhỏ hơn ***P*** và ***Q***. Ma trận ***P*** gọi là ma trận cơ sở người dùng với mỗi hàng trong ***P*** tượng trưng cho một người dùng. Các giá trị p_{ik} trong vector hàng i của ***P*** biểu thị mức độ quan tâm của người dùng i đến đặc tính k của hạng mục. Ma trận ***Q*** là ma trận đặc tính của hạng mục với mỗi cột trong ***Q*** tượng trưng cho một hạng mục. Các giá trị q_{kj} trong cột j biểu thị mức độ thuộc về của đặc tính k với hạng mụcj. *Hình 4.8* minh họa cho ý tưởng này. Như vậy từ ma trận ***R*** ban đầu, ta có thể phân rã ra thành 2 ma trận ***P*** và ***Q*** có hạng thấp hơn. Sau đó tìm ***P*** và ***Q*** sao cho phép nhân ***PQ*** xấp xỉ với ***R***.



Hình 4.8: Mô hình Phân rã ma trận

Giả sử ta cần khám phá K đặc tính tiềm ẩn (K< |U|, |D|), khi đó ta sẽ tìm 2 ma trận $\boldsymbol{P}_{|U| \text{ x}K}$ và $\boldsymbol{Q}_{|\text{D}| \text{ x K}}$ sao cho tích ***P***x***Q*** xấp xỉ được ma trận ***R***, nghĩa là:

$$\boldsymbol{R} \approx \boldsymbol{P} \text{x} \boldsymbol{Q}^{\text{T}} = \hat{R} \qquad \textbf{(4.4.1)}$$

Mỗi hàng của ***P*** biểu diễn mức độ quan tâm của người dùng với các đặc tính. Mỗi hàng của ***Q*** biểu diễn mức độ thuộc về của các đặc tính trong các hạng mục. Để dự báo điểm đánh giá của người dùng u_{i} dành cho hạng mục d_{j}, ta sẽ tính tích vô hướng của hai vector tương ứng với u_{i} và d_{j}:

$$\hat{r}_{ij} = p_i q_j^T = \sum_{k=1}^{K} p_{ik} q_{kj} \qquad \textbf{(4.4.2)}$$

Để tìm ***P*** và ***Q***, đầu tiên khởi tạo trị ban đầu cho ***P*** và ***Q***, tính ***M*** = ***PQ*** rồi tối thiểu hóa độ lệch giữa ma trận ***M*** với ***R***. Mỗi lần lặp là một lần điều chỉnh ***P*** và ***Q*** để tối thiểu hóa độ lệch giữa ***M*** và ***R***.

4.3.2. CÁC THUẬT TOÁN HỌC (Learning Algorithms)

Có nhiều phương pháp để giảm lỗi đến cực tiểu trong mô hình phân rã ma trận. Áp dụng các phương pháp này đồng nghĩa với việc tối ưu hóa hàm mục tiêu (objective function). Phương pháp được sử dụng phổ biến trong kỹ thuật phân rã ma trận là giảm gradient ngẫu nhiên (stochastic gradient descent).

4.3.2.1. PHƯƠNG PHÁP GIẢM GRADIENT NGẪU NHIÊN (STOCHASTIC GRADIENT DESCENT)

Phương pháp này nhằm tìm điểm cực tiểu cục bộ của độ lệch bình phương giữa 2 đối tượng. Hàm mục tiêu giảm độ lệch bình phương giữa 2 đối tượng A và B được định nghĩa như sau:

$$\min f(\mathrm{A}||\mathrm{B}) = (\mathrm{A} - \mathrm{B})^2 \qquad \textbf{(4.4.3)}$$

Độ lệch được gọi là lỗi giữa điểm đánh giá ước lượng với điểm đánh giá thực tế, có thể được tính toán bằng công thức sau cho mỗi cặp người dùng-hạng mục:

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 \qquad \textbf{(4.4.4)}$$

Ta xét lỗi bình phương vì điểm đánh giá ước lượng có thể có lúc cao hơn, có lúc thấp hơn điểm đánh giá thực tế gây ra hiện tượng bù trừ khi tính tổng lỗi.

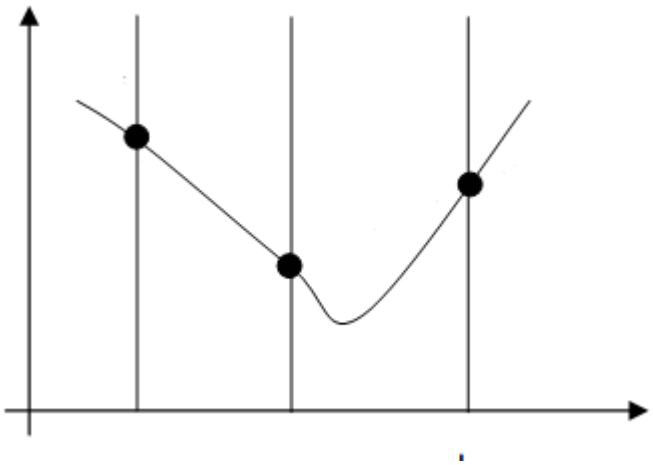
Để cực tiểu lỗi, ta phải biết hướng để chỉnh sửa các giá trị của p_{ik} và q_{kj}. Nghĩa là ta cần phải biết được gradient của các giá trị hiện tại. Lấy đạo hàm riêng cho công thức (*4.4.4*) theo các biến p_{ik} và q_{kj}, ta được:

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij} q_{kj}$$
$$\frac{\partial}{\partial q_{kj}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij} p_{ik} \qquad \textbf{(4.4.5)}$$

Cập nhật gradient cho các pik và qkj, ta có:

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj}$$
$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik} \qquad \textbf{(4.4.6)}$$

trong đó α, còn gọi là bước nhảy, là hằng số mà giá trị của nó xác định tốc độ dần đến điểm cực tiểu. Giá trị α là tùy chọn và thường khá nhỏ (khoảng 0.0002). Nếu chọn bước nhảy quá lớn để tiến đến cực tiểu có thể vượt khỏi điểm cực tiểu, và dao động quanh điểm cực tiểu. Ngược lại, nếu chọn α quá nhỏ tốc độ dần đến cực tiểu quá chậm. *Hình 4.9* cho thấy ảnh hưởng của bước nhảy trong quá trình dần đến điểm cực tiểu.



Hình 4.9: Ảnh hưởng của bước nhảy trong quá trình tiệm tiến đến cực tiểu

Vấn đề nảy sinh ra cho mô hình phân rã ma trận là nếu tìm được các ma trận ***P*** và ***Q*** sao cho ***P***x***Q*** xấp xỉ được ***R*** thì dự báo cho các đánh giá chưa thấy có dần về zero hết hay không? Thực tế, mô hình không đi tìm ***P*** và ***Q*** để có được ***R*** một cách chính xác, mà chỉ tối thiểu hóa lỗi của các cặp người dùng-hạng mục quan sát được. Nói cách khác, nếu đặt ***T*** là tập các bộ có dạng (u_i,d_j,r_{ij}), tức ***T*** chứa tất cả các cặp người dùng-hạng mục cùng với điểm đánh giá, ta sẽ cố gắng cực tiểu lỗi e_{ij} cho mỗi bộ (u_i,d_j,r_{ij}) $\in$***T*** (***T*** chính là tập dữ liệu huấn luyện). Đối với các bộ chưa đầy đủ (thiếu r_{ij}), mô hình sẽ xác định các giá trị này một khi mối quan hệ giữa người dùng, hạng mục và các đặc tính được học.

Sử dụng công thức *(4.4.5)* lặp đi lặp lại nhiều lần cho đến khi lỗi hội tụ về điểm cực tiểu. Có thể kiểm tra lỗi tổng thể sau mỗi lần tính toán theo công thức sau để xác định khi nào có thể dừng được:

$$E = \sum_{(u_i,d_j,r_{ij})\in T} e_{ij} = \sum_{(u_i,d_j,r_{ij})\in T} (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 \qquad \textbf{(4.4.7)}$$

4.3.2.2. PHƯƠNG PHÁP ALTERNATING LEAST SQUARE (ASL)

Do cả hai vector p_{ik} và q_{kj} đều chưa được xác định, hàm *(4.4.4)* không lõm. Tuy nhiên, nếu ta cố định một trong hai vector trên thì hàm *(4.4.4)* trở thành hàm bậc hai và có thể giải được bài toán tối ưu. Kỹ thuật ASL lần lượt luân phiên cố định p_{ik} và q_{kj}. Khi cố định p_{ik}, hệ thống sẽ tính toán lại q_{kj} bằng cách giải quyết vấn đề bình phương tối thiểu và ngược lại. Như vậy mỗi bước thực hiện sẽ giảm e_{ij} xuống cho đến khi hội tụ.

Nhìn chung, phương pháp giảm gradient ngẫu nhiên dễ cài đặt và chạy nhanh hơn phương pháp ASL, nhưng phương pháp ASL có ưu thế hơn trong ít nhất 2 trường hợp sau:

(1) Khi hệ thống có thể thực hiện song song hóa. Khi sử dụng phương pháp ASL, hệ thống sẽ tính mỗi q_{kj} và mỗi p_{ik} một cách độc lập. Điều này có thể làm phát sinh một lượng lớn tính toán song song.

(2) Khi hệ thống tập trung vào xử lý dữ liệu gián tiếp. Trong trường hợp này tập dữ liệu huấn luyện thường rời rạc, nếu thực hiện phương pháp giảm gradient ngẫu nhiên có thể xuất hiện hiện tượng loop trên mỗi trường hợp huấn luyện. Phương pháp ASL có thể xử lý tốt trường hợp này.

4.3.3. HỆ SỐ BIAS

Một ưu điểm của hệ thống phân rã ma trận là tính linh động về khả năng xử lý dữ liệu thuộc nhiều lĩnh vực khác nhau cũng như các yêu cầu có tính đặc thù của ứng dụng. Tuy nhiên, nhiều biến điểm đánh giá quan sát được chịu ảnh hưởng từ phía người dùng hoặc hạng mục, ảnh hưởng này được gọi là thiên hướng (*bias*) và tồn tại một cách độc lập với quan hệ chủ quan giữa người dùng và hạng mục. Thiên hướng này được thể hiện (trong dữ liệu) xu hướng có tính hệ thống, trong đó một số người dùng sẽ cho điểm đánh giá cao hơn số khác và / hoặc một số hạng mục được chấp nhận tốt hơn (hoặc kém hơn) so với các hạng mục khác.

Hệ tư vấn sẽ xác định tỉ lệ các điểm đánh giá chịu ảnh hưởng bởi thiên hướng. Xấp xỉ thiên hướng theo thứ tự ưu tiên ảnh hưởng lên điểm đánh giá như sau: Đặt b_{ij} là hệ số bias ảnh hưởng lên điểm đánh giá r_{ij},

$$b_{ij} = \mu + b_i + b_j \qquad \textbf{(4.4.8)}$$

trong đó μ là điểm đánh giá trung bình chung cho mọi hạng mục trong hệ thống b_i và b_j là độ lệch quan sát được của người dùng i và hạng mục j so với μ.

Ví dụ, giả sử ta muốn ước lượng hệ số bias của người dùng A đối với hạng mục B, biết rằng điểm đánh giá trung bình chung của hệ thống là $\mu = 3.7$. Ngoài ra, hạng mục B có điểm đánh giá cao hơn trung bình là $b_j = 0.5$, A là người thích các hạng mục

có đặc tính mà đa số các hạng mục trong hệ thống không có nên điểm đánh giá của A sẽ thấp hơn trung bình là $b_i = 0.3$. Khi đó, điểm đánh giá của A dành cho hạng mục B sẽ là 3.9 (3.7+0.5-0.3). Công thức *(4.4.2)* được viết lại như sau:

$$\hat{r}_{ij} = \mu + b_i + b_j + p_i q_j^T \qquad \textbf{(4.4.9)}$$

Như vậy một điểm đánh giá gồm có 4 thành phần: trung bình chung của hệ thống, hệ số bias của người dùng, hệ số bias của hạng mục và tương quan người dùng - hạng mục. Mỗi thành phần chỉ giải thích phần thuộc tính liên quan, nhờ đó mà độ chính xác của hệ thống có sử dụng hệ số bias sẽ cao hơn.

4.3.4. REGULARIZATION

Khi thực hiện tối ưu lỗi trong quá trình học của thuật toán có thể xảy ra hiện tượng overfitting trong dữ liệu huấn luyện. Đó là hiện tượng "tối ưu quá" khiến mô hình không đủ tổng quát để mô hình hóa dữ liệu mới. Regularization là kỹ thuật để cân bằng hiện tượng này. Có nhiều phương pháp regularization, đơn giản nhất là thêm vào một tham số β, công thức tính lỗi bình phương *(4.4.4)* được viết lại như sau:

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} (\|P\|^2 + \|Q\|^2) \qquad \textbf{(4.4.10)}$$

Tham số β được dùng để kiểm soát độ lớn của các vector đặc trưng của người dùng và vector đặc trưng của hạng mục sao cho $\boldsymbol{P}$ và $\boldsymbol{Q}$ xấp xỉ tốt tới $\boldsymbol{R}$ mà không chứa các số lớn. Trong thực tế, β là một tập các giá trị trong khoảng 0.02. Công thức cập nhật gradient cho các vector p_{ik} và q_{kj} với lỗi bình phương *(4.4.6)* được viết lại như sau:

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij} q_{kj} - \beta p_{ik}) \qquad \textbf{(4.4.11)}$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij} p_{ik} - \beta q_{kj}) \qquad \textbf{(4.4.12)}$$

4.3.5. PHÂN RÃ MA TRẬN KHÔNG ÂM (NMF)

Khi các giá trị trong ma trận đánh giá không âm, ta có mô hình NMF. Ưu điểm của NMF là cho ma trận kết quả mang ý nghĩa trực quan.

Vì không có phần tử nào âm nên ma trận kết quả của quá trình nhân ma trận để xấp xỉ về ma trận ban đầu sẽ không sinh ra số âm và có thể xem đây là một tiến trình sinh ra dữ liệu ban đầu bằng các tổ hợp tuyến tính các đặc tính tiềm ẩn. Phần lớn các dữ liệu điểm đánh giá thường là các ma trận không âm nên mô hình NMF được sử dụng phổ biến.

Về nguyên tắc, thuật toán NMF cũng được thực hiện theo các quy luật cập nhật *(II.11)* và *(II.12)*, tuy nhiên trong một số trường hợp cần đưa vào các ràng buộc $\boldsymbol{P} \geq 0$ và $\boldsymbol{Q} \geq 0$. *Thuật toán II.1* được viết lại như sau:

Thuật toán II.2. Thuật toán cập nhật nhân tử -NMF (Lee & Seung)

Input: ma trận $\boldsymbol{R}$, α, β, ε (ngưỡng lỗi tối thiểu chấp nhận được)

1: Khởi tạo ngẫu nhiên $\boldsymbol{P}$ và $\boldsymbol{Q}$

2: while k <iter OR e<ε

3: cập nhật $\boldsymbol{P}$ theo công thức (*4.4.11*)

4: cập nhật $\boldsymbol{Q}$ theo công thức (*4.4.12*)

5: tính lại e theo công thức (*4.4.10*)

6: if p_{ij}< 0 OR q_{ij}< 0 then

7: p_{ij} := 0; q_{ij} := 0

Output: ma trận $\hat{R} = PQ$

CHƯƠNG 5
HIỆN THỰC HỆ THỐNG

5.1. HỆ ĐIỀU HÀNH CHO ĐIỆN THOẠI THÔNG MINH

5.1.1. TẠI SAO TRIỂN KHAI TRÊN ĐIỆN THOẠI

Những hệ thống khuyến nghị thông thường được triển khai trên ứng dụng Web. Ví dụ như trang Amazon gợi ý những sản phẩm, trang Youtube gợi ý những bản nhạc đến người dùng … và rất nhiều trang khác. Ở đây, tôi không chọn Web mà quyết định chọn điện thoại thông minh để triển khai ứng dụng vì những lý do sau:

Người dùng luôn đem theo bên mình chiếc điện thoại. Vì vậy khi cần có thể sử dụng ở mọi lúc, mọi nơi. Thay vì họ phải tìm đến nơi có máy vi tính với kết nối Internet mới có thể tra cứu thông tin, như thế thật bất tiện.

Trên điện thoại thông minh có hệ thống GPS với bản đồ giúp người dùng xác định vị trí, đường đi, cùng nhiều tiện ích khác.

Khó đoán trước suy nghĩ và hành động của người dùng. Ở thời điểm này, điều kiện này, họ quyết định thế này. Ở thời điểm khác, điều kiện khác, họ có thể quyết định khác. Vì vậy, luôn mang bên mình chiếc điện thoại, họ sẽ được gợi ý kịp thời, kịp lúc. Đặc biệt là khi đang trong chuyến du lịch, họ có thể "hỏi" chiếc điện thoại của mình, không phải mất nhiều thời gian liên lạc với những ngườitư vấn.

Bên cạnh những lợi thế trên, điện thoại cũng có những nhược điểm. Một trong số đó là giao diện người dùng. Điện thoại đa số có cỡ màn hình khoảng từ 3-6 inches, không phải màn hình rộng (thường trêm 15 inches) như máy tính. Việc bố trí các chức năng, cách hiển thị thông trên một khung nhìn nhỏ cũng phải rất cẩn thận sao cho vừa đầy đủ cũng vừa không gây rối cho người dùng. Thêm nữa phần cứng trên điện thoại không mạnh mẽ như trên máy vi tính (về bộ nhớ, bộ vi xử lý, thời lượng pin…), nên tốc độ xử lý cũng có phần hạn chế. Những kỹ thuật xử lý khó khăn phức tạp sẽ được

thực hiện trên server và trả kết quả về và hiển thị trên điện thoại.

5.1.2. CHỌN LỰA GIỮA ỨNG DỤNG VÀ WEB TRÊN ĐIỆN THOẠI

Mỗi loại đều có ưu nhược điểm riêng, nhưng tôi quyết định chọn ứng dụng trên điện thoại vì những lý do sau:

Có thể can thiệp sâu vào những tính năng phần cứng của điện thoại do API được thiết kế riêng biệt cho nền tảng điện thoại đó mà trên web sẽ có những hạn chế nhất định.

Ứng dụng có khả năng chạy nền khi cần thiết.

Có thể sử dụng offline khi không có Internet (tính năng này chưa được thực hiện).

Trong tương lai, có thể dễ dàng thương mại hóa trên các cửa hàng ứng dụng (App Store).

5.1.3. TẠI SAO CHỌN ANDROID

Hiện nay trên thế giới, những nền tảng nổi bật nhất và chiếm phần lớn thị phần hệ điều hành cho điện thoại thông minh là Android (của Goole), iOS (của Apple) và Windows Phone (của Microsoft). Hệ điều hành iOS từ khi ra đời đã hướng đến mục tiêu phục vụ cho những nhu cầu thiên về giải trí trên điện thoại. Có thể thấy trên kho ứng dụng của Apple, đa phần là các trò chơi. Để lập trình được những ứng dụng chạy trên iOS, không phải lập trình viên nào cũng có điều kiện. Trước hết, cần phải có hệ điều hành MAC OS. Hệ điều hành này được cài sẵn trên các Macbook với giá bán không hề rẻ. Và IDE dùng để lập trình là Xcode cũng không phải miễn phí. Thời gian gần đây, hệ điều hành Windows Phone của Microsoft bắt đầu được phát triển. Hiện tại Windows Phone nhìn chung không hấp dẫn bằng iOS hay Android. Số lượng ứng dụng, số lượng lập trình viên tham gia phát triển, cũng như số lượng người dùng điện thoại thông minh nền tảng Windows Phone còn khá ít so với các nền tảng khác. Hệ

điều hành nổi bật nhất trên thị trường điện thoại thông minh lúc này vẫn là Android.

Android được biết đến như là một hệ điều hành mã nguồn mở trên điện thoại di động. Hiện nay, Android được sử dụng cả trên những thiết bị điện tử khác như: máy tính bảng, thiết bị giải trí đa phương tiện, TV …. Trước đây Android được phát triển dựa trên nền tảng Linux bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005). Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai. Phiên bản Android đầu tiên dành cho các dòng điện thoại thông minh là 1.5. Hiện nay, bản mới nhất là 5.0 - Lollipop.

Dưới đây là những thành phần cốt lõi của hệ điều hành Android:

Applications: Khi bắt đầu cài đặt Android trên điện thoại di động, các ứng dụng cơ bản như email, SMS, lịch, bản đồ, trình duyệt, quản lý danh bạ … được tích hợp sẵn. Tất cả những ứng dụng khác có thể được xây dựng thêm bằng ngôn ngữ lập trình Java và cài đặt vào điện thoại.

Application Framework: cho phép lập trình viên dễ dàng xây dựng những dụng mạnh mẽ có khả năng tái sử dụng cao. Những thành phần của ứng dụng này có thể được kế thừa để sử dụng hoặc phát triển thêm cho những ứng dụng khác. Những người lập trình có toàn quyền truy xuất, sử dụng những sức mạnh phần cứng của chiếc điện thoại trong lúc lập trình ứng dụng (GPS, bluetooth, WiFi, cảm biến gia tốc, la bàn…)

Libraries: gồm một tập các thư viện C/C++ được viết sẵn hỗ trợ xử lý âm thanh, hình ảnh, hiệu ứng đồ họa 2D, 3D, trình duyệt web, cơ sở dữ liệu SQLite …

Android Runtime: mỗi ứng dụng Android chạy trong một thể hiện của máy ảo Dalvik. Trên Android hỗ trợ chạy đa nhiệm. Máy ảo Dalvik thực thi những file ứng

dụng ở dạng .dex (Dalvik Executable) được tối ưu hóa cho bộ nhớ và phần cứng điện thoại.

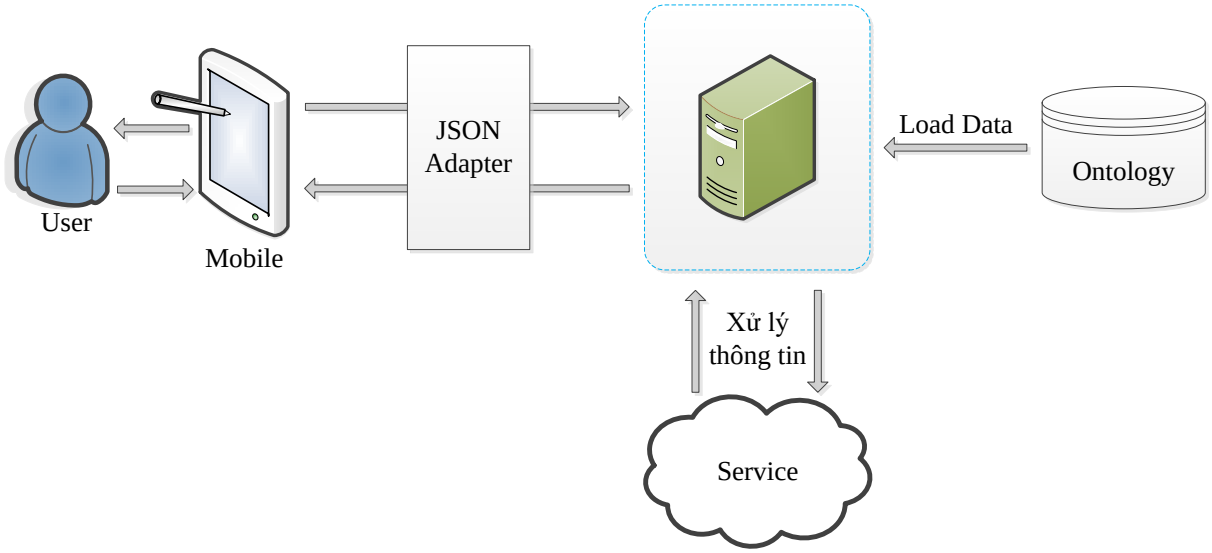
Linux Kernel: Android dựa trên nhân Linux phiên bản 2.6 cung cấp khả năng bảo mật, quản lý bộ nhớ, quản lý tiến trình …

Chi tiết hơn về hệ điều hành Android, người đọc có thể tìm hiểu thêm trên trang http://developer.android.com/.

Vì thế tôi quyết định chọn nền tảng Android vì những lý do chính sau đây:

- Android là hệ điều hành mã nguồn mở do Google xây dựng và phát triển. Nguồn tài liệu tham khảo dồi dào cũng như cộng đồng lập trình viên rất đông đúc trên toàn cầu.
- Điện thoại sử dụng Android ngày càng chiếm thị phần lớn do giá thành rẻ hơn so với các nên tảng khác. Theo khảo sát mới nhất, trong năm 2015, Android dẫn đầu thị trường điện thoại thông minh với tỉ lệ khoảng 79%. Dự kiến sẽ tiếp tục tăng trong thời gian tới.
- Về hiệu năng, Android đáp ứng tốt không thua kém các hệ điều hành khác. Thêm nữa, phía sau là Google với những nền tảng dịch vụ tuyệt vời.
- Chi phí đầu tư để lập trình trên Android miễn phí, đơn giản. Ngôn ngữ lập trình Android xuất phát từ Java, một ngôn ngữ rất phổ biến trên thế giới. Các IDE lập trình được cung cấp miễn phí cho lập trình viên.

5.2. MÔ HÌNH HỆ THỐNG



Hình 5.1: Mô hình hoạt động của hệ thống

Hệ thống hoạt động theo mô hình Client - Server. Client sẽ nhận các thông tin từ người dùng và gửi về Server, Server sẽ nhận, xử lý các thông tin của người dùng và trả kết quả lại cho họ. Client và Server sẽ trao đổi thông tin với nhau bằng phương thức JSON, theo đó, đoạn JSON được gửi đi sẽ kèm theo một mã, để thông báo cho Server biết phải làm gì:

- Mã 1: Yêu cầu 4 câu hỏi cơ bản.
- Mã 2: Yêu cầu câu hỏi thứ 5.
- Mã 3: Yêu cầu các câu hỏi tiếp theo.
- Mã 5: Yêu cầu danh sách Tour.

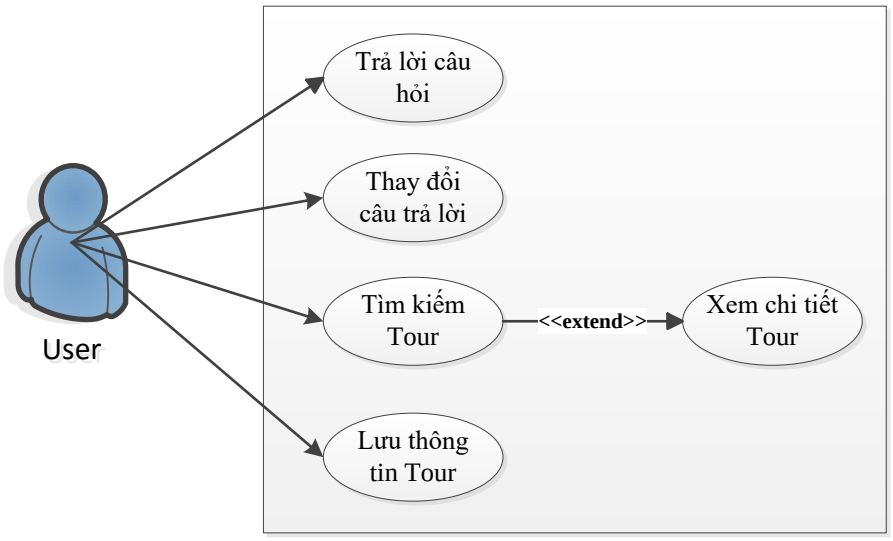
Về cơ bản, một phiên làm việc giữa Client và Server sẽ trãi qua 4 bước như sau:

- Bước 1: Server sẽ gửi câu hỏi về ứng dụng trên điện thoại Android.
- Bước 2: Người dùng trả lời câu hỏi, sau khi trả lời xong, kết quả được tự động gửi lại Server.
- Bước 3: Server xử lý đáp án của người dùng, tùy trường hợp mà trả về câu hỏi tiếp theo hoặc danh sách Tour gợi ý cho người dùng.

- Bước 4: Kết thúc phiên làm việc, thông tin người dùng được Server tự lưu lại nhằm mục đích tự hoàn thiện hệ thống.

Cùng một thời điểm, Server có thể làm việc với nhiều Client khác nhau bằng cáchgán cho mỗi Client một ID nhằm mục đích phân biệt các Client với nhau, rồi sau đó, dựavào ID, Server lưu lại trạng thái làm việc của Client.

5.3. LƯỢC ĐỒ USERCASE



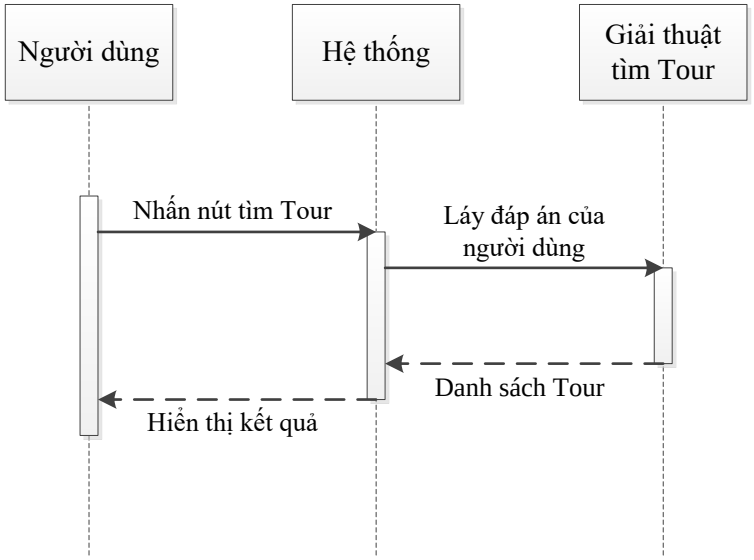
Hình 5.2: Lược đồ Usecase của hệ thống

5.4. CÁC CHỨC NĂNG CHÍNH

Hệ thống có 2 chức năng chính là tìm kiếm Tour và lưu Tour.

5.4.1. TÌM KIẾM TOUR

Tìm kiếm Tour:

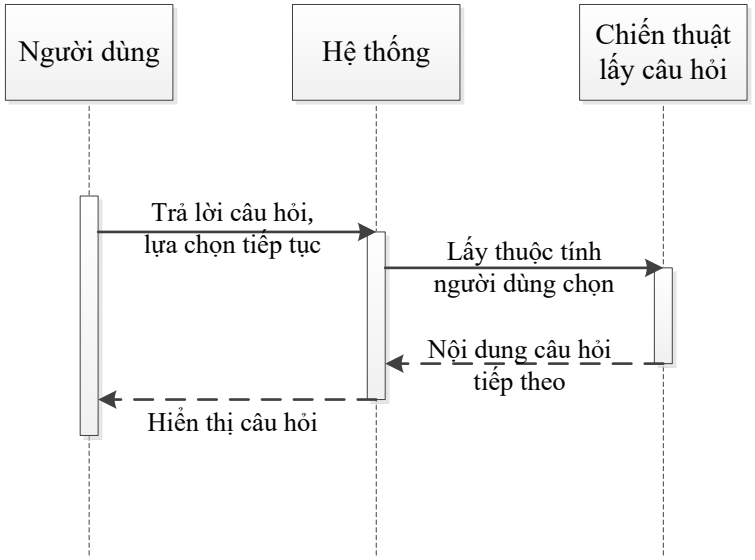


Hình 5.3: Lược đồ tuần tự tìm tour

Mô tả chức năng: Hệ thống nhận yêu cầu tìm Tour. Dựa vào câu trả lời của người dùng, thuật toán tìm kiếm Tour sẽ tìm ra danh sách các Tour phù hợp. Dòng sự kiện:

- Bước 1: Hệ thống nhận yêu cầu của người dùng.
- Bước 2: Dựa vào câu trả lời, giải thuật tìm Tour sẽ trả về danh sách Tour.
- Bước 3: Hiển thị danh sách Tour.

Trả lời câu hỏi:

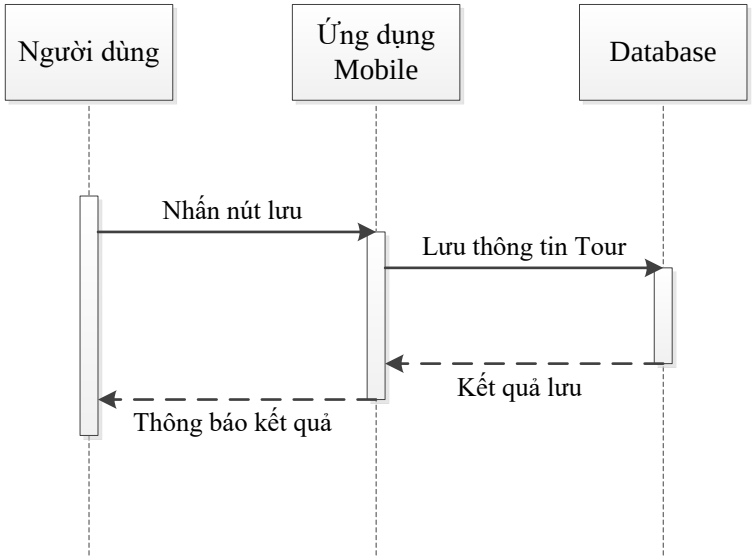


Hình 5.4: Lược đồ tuần tự trả lời câu hỏi

Mô tả chức năng: Hệ thống đưa ra câu hỏi dựa trên câu trả lời trước của người dùng, đưa ra câu hỏi tiếp theo. Mục tiêu là lấy được càng nhiều thông tin càng tốt. Dòng sự kiện:

- Bước 1: Hệ thống đưa ra câu hỏi
- Bước 2: Người dùng trả lời câu hỏi
- Bước 3: Từ câu trả lời của người dùng, áp dụng chiến thuật đưa ra câu hỏi tiếp theo
- Bước 4: Hiển thị câu hỏi vừa tìm được

5.4.2. LƯU THÔNG TIN TOUR



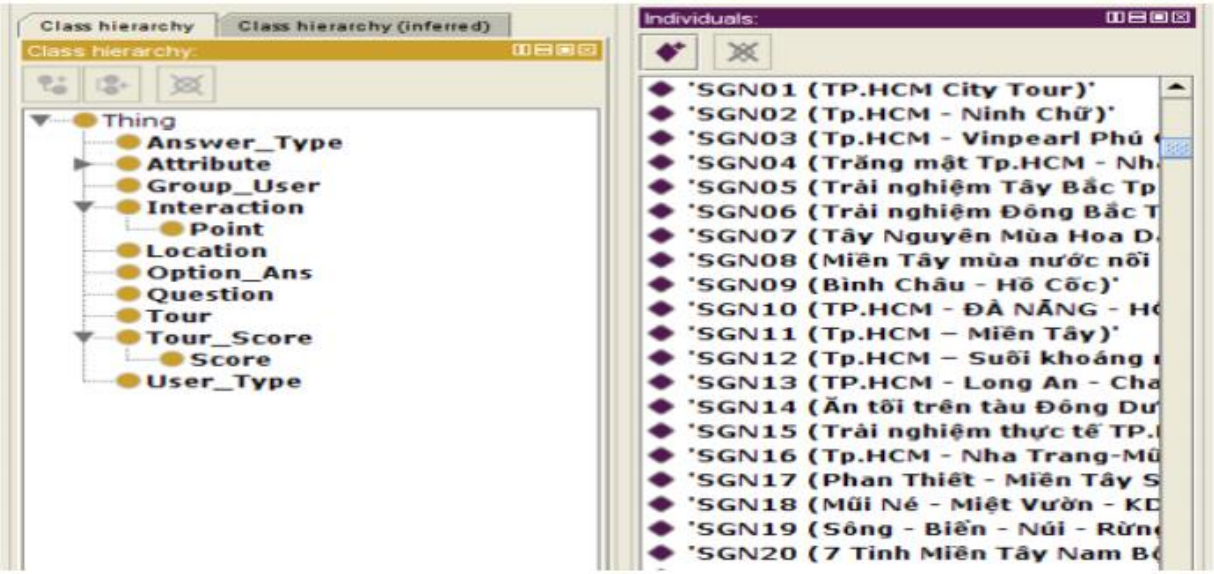
Hình 5.5: Lược đồ tuần tự lưu tour

Mô tả chức năng: hệ thống có khả năng lưu lại thông tin Tour để người dùng có thể xem lại. Dòng sự kiện:

- Bước 1: Hệ thống nhận yêu cầu của người dùng.
- Bước 2: Hệ thống sẽ lưu thông tin Tour vào database.
- Bước 3: Hiển thị kết quả Tour đã lưu.

5.5. THIẾT KẾ ONTOLOGY

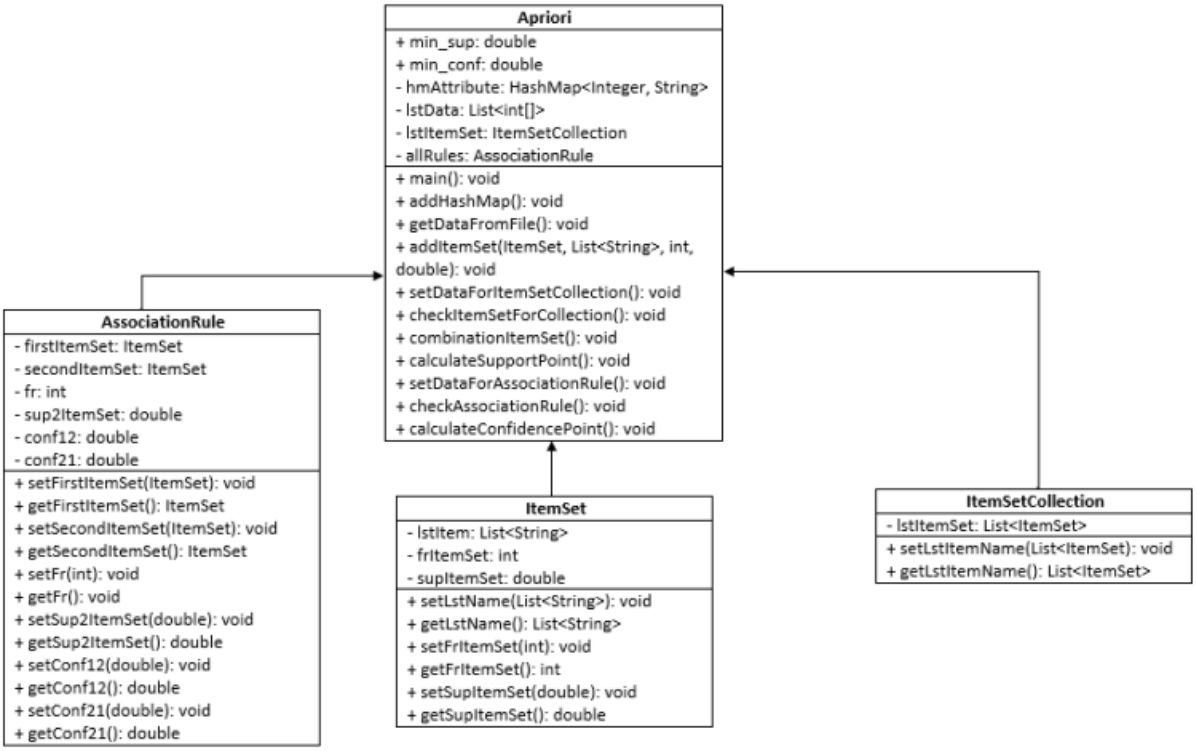
Dữ liệu trên Ontology được chuyển qua từ MySQL, về cơ bản, các mối quan hệ trong database được giữ nguyên khi chuyển qua Ontology bằng cách khai báo các Data Property và Object Property.



Hình 5.6: Dữ liệu trên Ontology

5.6. HIỆN THỰC THUẬT TOÁN APRIORI

5.6.1. CLASS DIAGRAM



Hình 5.7: Class Diagram

5.6.2. MÃ GIẢ

Với thiết kế của Class Diagram được thể hiện như trên thì mã giả của thuật toán Apriori được hiện thực như sau:

Data:

- minsup, minconf
- k-ItemSet: List Items has k elements
- ItemSetCollection: List ItemSet
- AssociationRule: Contain 2 ItemSet and interaction point of it.

Result:

- List<AssociationRule>: List Rules satisfy the conditions

Read data from database;

Set data for 1-ItemSet;

for *(i=0; i<size(1-ItemSet); i++)* **do**

    if *(supp(item(i)) < minsup)* **then**

        delete(item(i));

    end

end

for *(i=0; i<size(1-ItemSet)-1; i++)* **do**

    for *j=i+1; j< size(1-ItemSet); j++* **do**

        combines each pair of 1-ItemSet to AssociationRule;

    end

end

for *all rule on AssociationRule* **do**

calculates support point of each pair of 1-ItemSet of rule;

if *(support point(rule(i)) < minsup)* **then**

delete(rule(i));

end

end

for *(all rule on AssociationRule)* **do**

calculates confidence point of rule, conf(X=>Y), conf(Y=>X);

if *conf(X=>Y) > minconf* **then**

print(conf(X=>Y));

end

if *conf(Y=>X) > minconf* **then**

print(conf(Y=>X));

end

end

5.7. HIỆN THỰC KỸ THUẬT MATRIX FACTORIZATION

5.7.1. ĐẦU VÀO

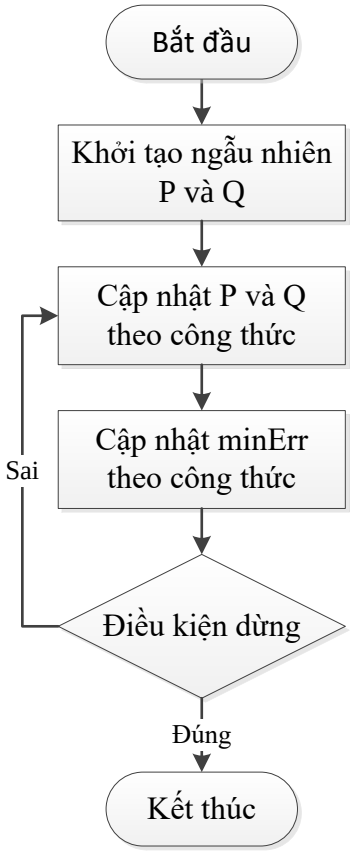
- Ma trận R: Người dùng, Hạng mục, Đánh giá.
- Số lần học.
- Số đặc tính quan tâm.
- Lỗi tối thiểu cho phép.
- Hệ số alpha.

- Hệ số beta.

5.7.2. ĐẦU RA

- Ma trận $\hat{R}$

5.7.3. LƯU ĐỒ THUẬT TOÁN



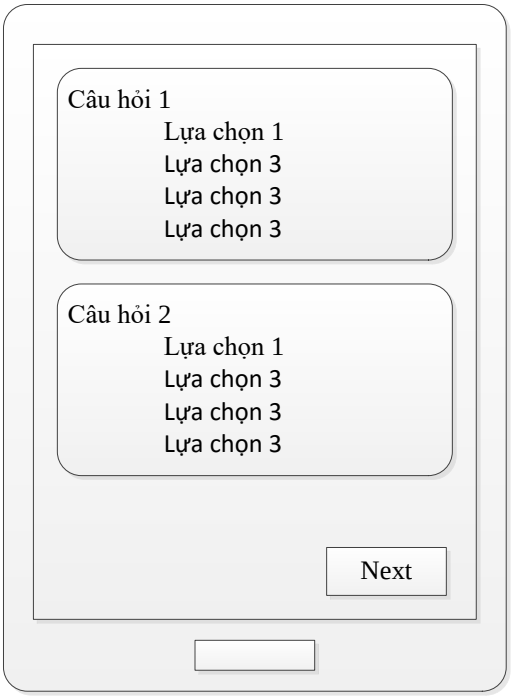
Hình 5.8: Lưu đồ thuật toán NMF

5.8. THIẾT KẾ ỨNG DỤNG TRÊN ĐIỆN THOẠI DI ĐỘNG

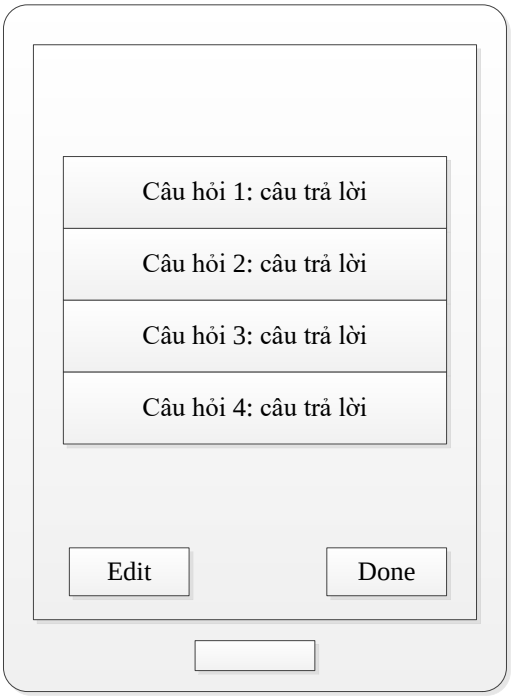
Ứng dụng được xây dựng bằng công cụ Android Studio trên nền Java version 1.8.0_65.

5.8.1. MOCKUP

- Mockup cho 4 câu hỏi đầu tiên:



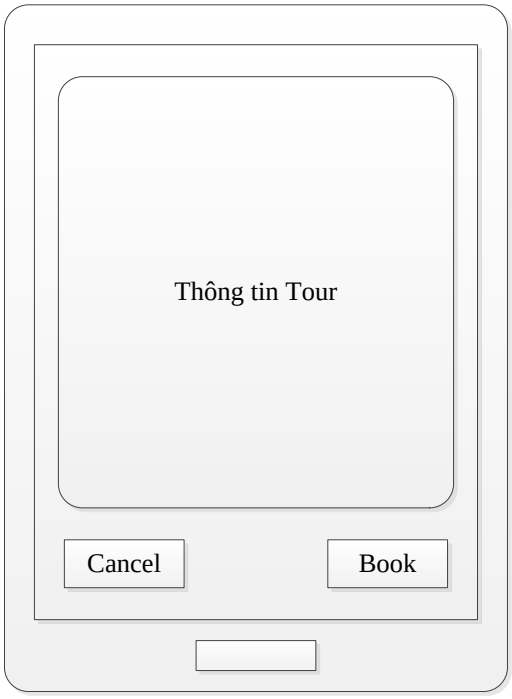
- Mockup hiển thị thông tin người dùng đã chọn cho 4 câu hỏi đầu tiên:



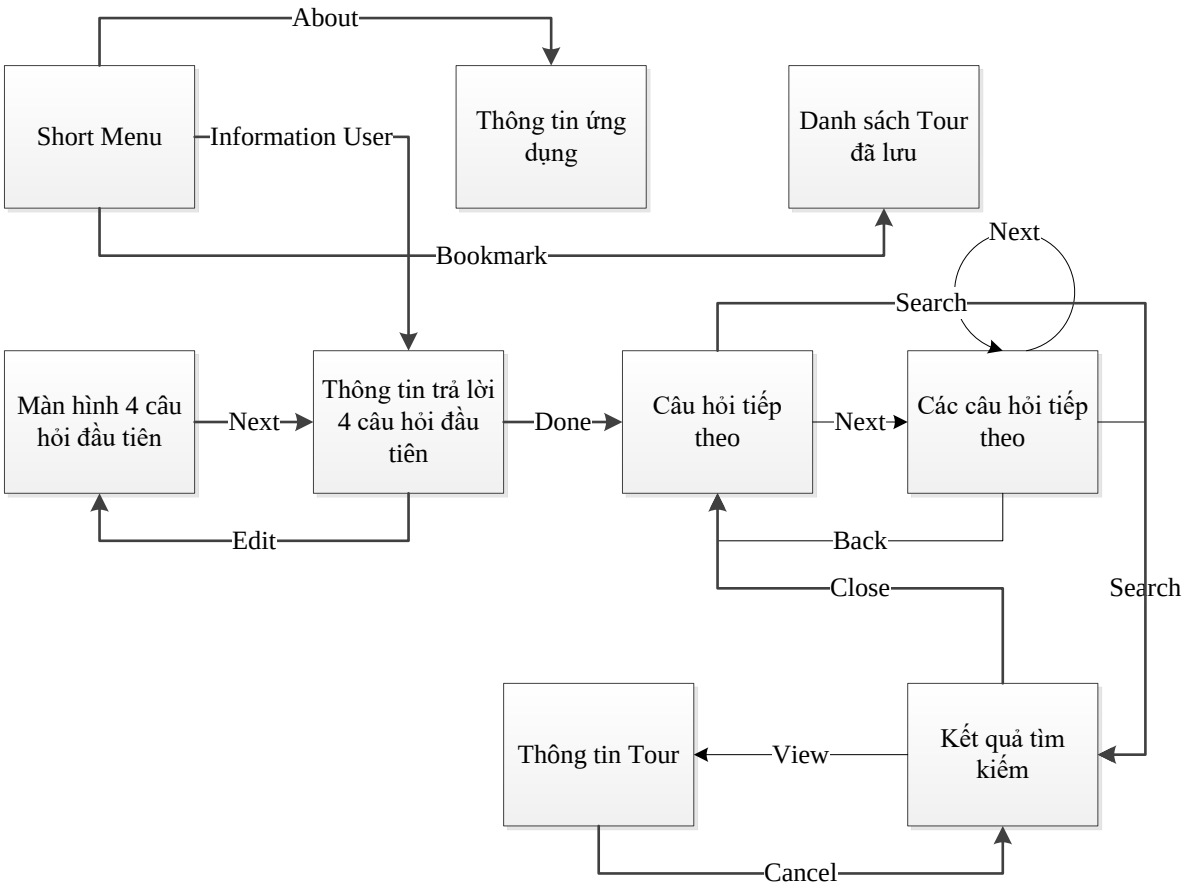
- Mockupcho các câu hỏi tiếp theo:

- Mockup kết quả Tour:

- Mockup thông tin Tour:



5.8.2. SCREEN FLOW



Hình 5.9: Screen flow của ứng dụng

5.8.3. CHỨC NĂNG

Ứng dụng trên Android là phần trung gian giữa người dùng và hệ thống. Ứng dụng có 4 chức năng chính:

- Hiển thị các câu hỏi.
- Ghi nhận câu trả lời của người dùng.
- Hiển thị danh sách các Tour được gợi ý.
- Lưu lại các Tour nếu người dùng muốn xem lại.

5.9. KIỂM TRA, ĐÁNH GIÁ HỆ THỐNG

5.9.1. KIỂM TRA HỆ THỐNG

Việc kiểm tra hệ thống chủ yếu dựa vào việc kiểm tra các chức năng của ứng dụng Android.

Các bước thực hiện:

- Xây dựng bộ testcase.
- Chạy ứng dụng dựa theo bộ testcase và ghi lại kết quả.
- Đánh giá kết quả.

Chi tiết bộ testcase kiểm tra ứng dụng được mô tả ở Bảng 1 của Phụ Lục A.

5.9.2. ĐÁNH GIÁ HỆ THỐNG

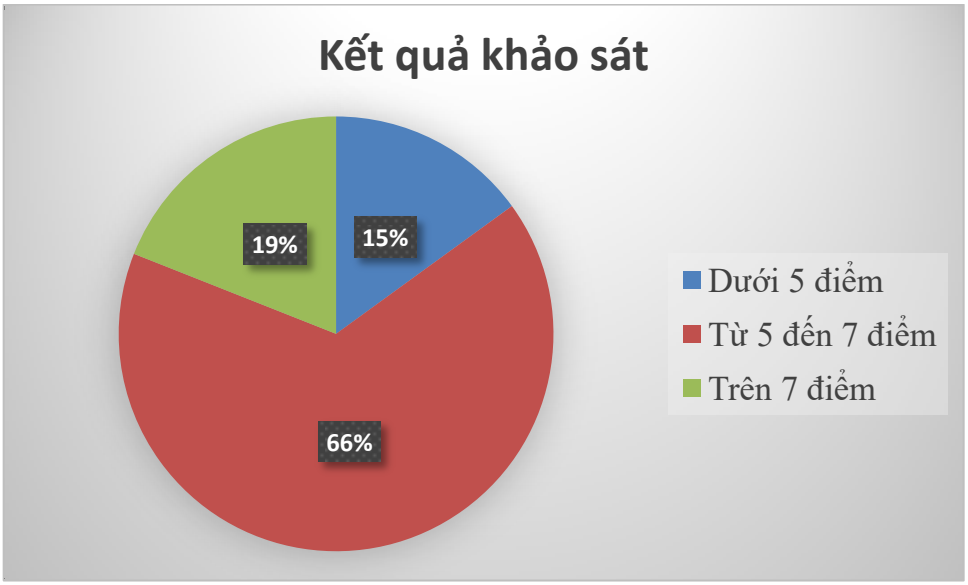
Đề tài được hiện thực dựa trên đề tài có sẵn cùng với một số cải tiến nhằm hoàn chỉnh hệ thống. Đầu tiên là sự thay đổi về lưu trữ dữ liệu, tôi chọn Ontology thay cho MySQL bởi vì một số ưu điểm của Ontology như tôi đã trình bày ở Chương 2. Với Ontology, một người không có nhiều kiến thức về cơ sở dữ liệu cũng có thể dễ dàng hiểu được cấu trúc dữ liệu của hệ thống. Tiếp theo là áp dụng thuật toán Apriori vào hệ thống để tính điểm tương tác giữa các thuộc tính. Áp dụng kỹ thuật Matrix Factorization để lấp đầy bảng ma trận điểm tương tác giữu các thuộc tính cho. Cuối cùng xây dựng được ứng dụng Tour Suggesstion chạy trên nên Android.

Tuy nhiên, việc áp dụng Ontology để lưu trữ dữ liệu hoặc áp dụng Apriori có thể sẽ làm cho hệ thống có nhiều sự khác biệt so với hệ thống cũ, với cùng số lượng câu hỏi, chọn cùng đáp án, danh sách các Tour có thể khác nhau. Mặt khác, khi dữ liệu quá lớn, việc sử dụng Ontology sẽ làm cho hệ thống bị chậm đi do phải load tất cả dữ liệu vào bộ nhớ để xử lý.

- Chạy thử và đánh giá của người dùng:
    - Khảo sát 32 người:
        - Bạn bè

- Người thân
- Đồng nghiệp

- Kết quả thu được:



Hình 5.10: Kết quả khảo sát người dùng

5.9.3. ĐÁNH GIÁ CÁC GIẢI THUẬT

Phân tích đánh giá việc áp dụng Apriori vào hệ thống. Thuật toán được kiểm tra bằng các testcase như *Bảng 5.1*.

Bảng 5.1: Bảng testcase kiểm tra thuật toán Apriori

STT	Mô tả	Kết quả mong muốn	Kết quả
1	Tăng minsup và min-conf	Giảm số lượng các cặp tương tác với nhau	Số lượng các cặp tương tác với nhau giảm theo
2	Giảm minsup và min-conf	Tăng số lượng các cặp tương tác với nhau	Số lượng các cặp tương tác với nhau tăng theo

Đánh giá thuật toán Apriori:

- Apriori là thuật toán đơn giản, dễ hiểu và dễ cài đặt.
- Tuy nhiên, Apriori có các nhược điểm như:
    - Phải duyệt CSDL nhiều lần. Với $I = i_1, i_2, \ldots, i_{100}$, số lần duyệt CSDL sẽ là 100.
    - Số lượng tập ứng viên rất lớn: $2^{100} - 1 = 1.27 * 10^{30}$.
    - Thực hiện việc tính độ phổ biến nhiều, đơn điệu.

Đánh giá kỹ thuật Matrix Factorization:

- Thuật toán xấp xỉ tốt các giá trị > 0 trong ma trận dữ liệu R ban đầu. Các phần tử có giá trị bằng 0 trong R đều đã được lấp đầy bởi các giá trị xấp xỉ.
- Giá trị của các phần tử trong ma trận kết quả tương ứng với các phần tử có điểm đánh giá trong ma trận ban đầu có thể xấp xỉ lớn hơn hoặc xấp xỉ nhỏ hơn.
- Điều kiện dừng minErr rất khó đạt được vì tổng lỗi tăng tỷ lệ thuận với độ lớn tập dữ liệu. Thuật toán chỉ dừng khi chạy đủ số lần học.
- Lỗi bình phương không phụ thuộc vào số lần học. Do đó cần chọn số lần học phù hợp với tập dữ liệu để có lỗi bình phương tốt nhất.
- Lỗi bình phương không phụ thuộc vào hệ số alpha. Do đó cần chọn số lần học phù hợp với tập dữ liệu để có lỗi bình phương tốt nhất.
- Lỗi bình phương giảm dần khi ta giảm hệ số Beta, như vậy lỗi bình phương phụ thuộc vào hệ số Beta.
- Tùy độ lớn tập dữ liệu, chúng ta sẽ quyết định chọn các thông số phù hợp để tối ưu chi phí tính toán và kết quả.

CHƯƠNG 6
KẾT LUẬN

6.1. KẾT QUẢ ĐẠT ĐƯỢC

Kết thúc giai đoạn luận văn, đã đạt được những mục tiêu đề ra ban đầu như:

- Xây dựng được giải thuật tìm Tour và chiến lược đặt câu hỏi phù hợp bằng cách áp dụng thuật toán Apriori và kỹ thuật matrix factorization.
- Xây dựng được ứng dụng Tour suggestion trên thiết bị Android.

6.2. HẠN CHẾ CỦA HỆ THỐNG

- Câu hỏi chỉ ở dạng trắc nghiệm, chưa đa dạng về thể loại.
- Giao diện ứng dụng còn đơn giản.
- Tour được gợi ý có thể chưa phù hợp với nhu cầu của người dùng do các giải thuật chưa có độ chính xác cao.

6.3. ĐỊNH HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Do thời gian thực hiện đề tài có hạn và đề tài mang tính thực tiễn nên kết quả của tôi còn một số hạn chế nhất định. Tôi xin đưa ra hướng phát triển thêm cho đề tài như sau:

- Chia nhóm người dùng 1 cách cụ thể, nhằm loại bỏ một số tùy chọn trong câu hỏi. Ví dụ: khi người dùng là người có thu nhập thấp, ta nên loại bỏ những Tour có giá cao…
- Thêm nhiều câu hỏi để có thể thu thập thông tin người dùng nhiều hơn.
- Hệ thống có thể gợi ý ngay từ đầu bằng những Tour được đặt nhiều hoặc những Tour phù hợp với những sự kiện đang diễn ra.
- Xây dựng một hệ thống hoàn chỉnh, bao gồm cả đặt Tour.
- Xây dựng, bổ xung bộ testcase đầy đủ hơn.

TÀI LIỆU THAM KHẢO

[1] Gavalas, Damianos, et al. "Mobile recommender systems in tourism." Journal of Network and Computer Applications 39 (2014): 319-333.

[2] Ricci, Francesco. "Travel recommender systems." IEEE Intelligent Systems 17.6 (2002): 55-57.

[3] Bauernfeind, Ulrike. "The evaluation of a recommendation system for tourist destination decision making." Proceedings of the XII International Symposium on Tourism and Leisure. 2003.

[4] Uschold, Mike, and Gruninger. "Ontologies: Principles, methods and applications." Knowledge engineering review 11.2 (1996): 93-136.

[5] Prud'Hommeaux, Eric, and Seaborne. "SPARQL query language for RDF." W3C recommendation 15 (2008).

[6] Mair, Alex. "Recommender Systems for Tourism." Advanced Interface Design (2004): 71.

[7] Nguyễn Lê Duy, "Kỹ thuật matrix factorization trong xây dựng hệ tư vấn", Khoa công nghệ thông tin, Trường Đại Học Công Nghệ TP. HCM. (2015).

[8] Knublauch, Holger, et al. "The Protege OWL Experience." OWLED. 2005.

[9] McGuinness, Deborah, and Harmelen. "OWL web ontology language overview." W3C recommendation 10.10 (2004): 2004.

[10] Yoo, Kyung-Hyan, Sigala, and Gretzel. "Exploring TripAdvisor." Open Tourism. Springer Berlin Heidelberg, 2016. 239-255.

[11] Đỗ Quốc Dũng, "Xây dựng hệ thống tư vấn du lịch trực tuyến", Khoa khoa học và kỹ thuật máy tính, Trường Đại Học Bách Khoa TP. HCM. (2014).

[12] Tan, Pang-Ning, Steinbach, and Kumar. "Association analysis: basic concepts and algorithms." Introduction to data mining (2005): 327-414.

[13] Bray, Tim. "The javascript object notation (json) data interchange format." (2014).

[14] Mednieks, Zigurd, et al. Programming Android. "O'Reilly Media, Inc.", 2012.

[15] Sigala, Marianna, Christou, and Gretzel, eds. Social media in travel, tourism and hospitality: Theory, practice and cases. Ashgate Publishing, Ltd., 2012.

PHỤ LỤC A

Bảng 1: Bộ testcase kiểm tra ứng dụng

STT	Miêu tả	Điều kiện	Các bước	Mong muốn	Kết quả
1	Kiểm tra giao diện		Chạy qua tất cả các màn hình	Hiển thị được câu hỏi, các đáp án của câu hỏi, hiển thị được thông tin chi tiết Tour	Đạt yêu cầu
2	Kiểm tra chức năng tìm kiếm		Nhấn nút Search	Trả về danh sách Tour	Đạt yêu cầu
3	Kiểm tra lỗi kết nối	Ngắt kết nối internet		Có thông báo lỗi khi không có mạng và ngược lại	Đạt yêu cầu
4	Lưu đáp án của 4 câu đầu	Đã trả lời 4 câu hỏi đầu	Nhấn nút Done	Hiển thị đáp án 4 câu hỏi đầu	Đạt yêu cầu
5	Sửa thông tin 4 câu đầu	Đã trả lời 4 câu hỏi đầu	Nhấn nút Edit	Hiển thị 4 câu hỏi với các đáp án đã chọn và có thể chọn lại đáp án	Đạt yêu cầu
6	Qua câu hỏi tiếp theo	Trả lời câu hỏi	Nhấn nút Next	Hiển thị câu hỏi tiếp theo	Đạt yêu cầu

7	Trở về câu hỏi trước	Đã trả lời 1 câu hỏi	Nhấn nút Back	Hiển thị câu hỏi trước với các đáp án đã chọn	Đạt yêu cầu
8	Tìm kiếmTour	Trả lời ít nhất 1 câu hỏi	Nhấn nút Search	Hiển thị danh sách Tour phù hợp	Đạt yêu cầu
9	Xem thông tin Tour	Đã tìm Tour	Nhấn vào Tour	Hiển thị chi tiết Tour	Đạt yêu cầu
10	Lưu thông tin Tour	Xem chi tiết Tour	Nhấn vào nút ngôi sao	Thông tin Tour được lưu lại	Đạt yêu cầu
11	Xem lại thông tin Tour đã lưu	Đã lưu thông tin Tour	Chọn danh sách Tour đã lưu	Hiển thị chi tiết Tour	Đạt yêu cầu