**TÀI LIỆU BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

[I. Mở đầu 4](#_Toc435991798)

[II. HỆ THỐNG GỢI Ý SỬ DỤNG PHẢN HỒI TIỀM ẨN 5](#_Toc435991799)

[A. Hệ thống gợi ý 5](#_Toc435991800)

[1.1 Mục đích 6](#_Toc435991801)

[1.2 Các kỹ thuật gợi ý 7](#_Toc435991802)

[III. XÂY DỰNG HỆ THỐNG 11](#_Toc435991803)

[1. Các định nghĩa, viết tắt và mô tả 11](#_Toc435991804)

[2. Các yêu cầu ở mức cao 11](#_Toc435991805)

[2.1 Product Perspective 11](#_Toc435991806)

[2.2 Miêu tả tác nhân và các ca sử dụng 11](#_Toc435991807)

[2.2.1 Biểu đồ UC 11](#_Toc435991808)

[2.2.2 Biểu đồ lớp tham gia UC 13](#_Toc435991809)

[2.2.3 Biểu đồ trình tự 15](#_Toc435991810)

[2.2.2 Mô tả tác nhân 18](#_Toc435991811)

[2.2.3 Mô tả UC 18](#_Toc435991812)

[2.2.4 Biểu đồ UC và Actor 19](#_Toc435991813)

[2.3 Môi trường hệ điều hành 20](#_Toc435991814)

[2.4 Giả thiết và các điều kiện lệ thuộc 20](#_Toc435991815)

[3. Yêu cầu chức năng 20](#_Toc435991816)

[3.1 Login 20](#_Toc435991817)

[3.2 Clustering customer 21](#_Toc435991818)

[3.3 Recommend product 22](#_Toc435991819)

[3.4 Retrieve dynamic data 24](#_Toc435991820)

[4.Thiết kế 25](#_Toc435991821)

[4.1 Thiết kế cơ sở dữ liệu 25](#_Toc435991822)

[4.2 Thiết kế thuật toán 30](#_Toc435991823)

[5. Lập trình 30](#_Toc435991824)

[6. Kiểm thử 31](#_Toc435991825)

[7. Vận hành bảo trì 31](#_Toc435991826)

[8. Các yêu cầu phi chức năng 31](#_Toc435991827)

[8.1 Khả năng thích ứng 31](#_Toc435991828)

[8.2 Kiểm tra khả năng 31](#_Toc435991829)

[8.3 Tính đúng đắn 31](#_Toc435991830)

[8.4. Bảo trì 31](#_Toc435991831)

[8.5. Thời gian đáp ứng 32](#_Toc435991832)

[8.6. Khả năng tương thích 32](#_Toc435991833)

[8.7. Mạnh mẽ 32](#_Toc435991834)

[8.8 Sự an toàn 32](#_Toc435991835)

[9.1 Những hạn chế phần cứng 33](#_Toc435991836)

[9.2 Phần mềm hạn chế 33](#_Toc435991837)

[9.3. High-Level Ngôn ngữ 33](#_Toc435991838)

**NỘI DUNG**

# I. Mở đầu

Hệ thống gợi ý (Recommender System - RS) được ứng dụng khá thành công trong thực tiễn giúp người dùng giải quyết vấn đề quá tải thông tin. Hiện nay, hệ thống gợi ý đang được nghiên cứu và ứng dụng ở nhiều lĩnh vực khác nhau đặc biệt là thương mại điện tử. Trên thế giới, đã có nhiều công ty, tổ chức áp dụng thành công hệ thống gợi ý như là một dịch vụ thương mại của mình nhằm gợi ý các dịch vụ, sản phẩm và các thông tin cần thiết đến người dùng như: website mua sắm trực tuyến Amazon (www.amazon.com) cung cấp cho khách hàng những sản phẩm mà họ có thể quan tâm, cổng video clip YouTube (www.youtube.com), giới thiệu phim của MovieLens (www.movielens.org),... Việc gợi ý sản phẩm phù hợp góp phần làm tăng doanh số bán hàng hoặc số lượng truy cập, download của hệ thống. Đồng thời giúp cho khách hàng có thể tìm kiếm được những thông tin thú vị hoặc những sản phẩm mà họ muốn tìm dễ dàng hơn. Hệ thống gợi ý giúp người dùng chọn lựa được thông tin phù hợp nhất cho mình dựa trên những hành vi/phản hồi (feedbacks) mà người dùng đã thực hiện trong quá khứ. Các phản hồi có thể được xác định một cách tường minh (explicit feedback) như thông qua việc đánh giá/xếp hạng (ví dụ, rating từ 1 đến 5; hay like (1) và dislike (0),…) mà người dùng đã bình chọn cho trên sản phẩm – trong trường hợp này gọi là dự đoán xếp hạng (rating prediction) hoặc các phản hồi có thể được xác định một cách không tường minh hay còn gọi là tiềm ẩn (implicit feedback) như số lần click chuột, số lần chọn mua sản phẩm, thời gian mà người dùng đã duyệt/xem sản phẩm,… Rất nhiều hệ thống lớn thu thập thông tin phản hồi từ khách hàng một cách tường minh, như Ebay, Amazon, LastFM, NetFlix,.. ở đó người dùng bình chọn sản phẩm từ \* đến \*\*\*\*\* (rất thích); hay Youtube thu thập thông tin qua like/ dislike, và các hệ thống khác. Thông qua việc thu thập phản hồi tường minh, hệ thống dễ dàng xác định rõ mức độ yêu thích của người dùng trên sản phẩm, từ đó dự đoán các sản phẩm tiếp theo mà người dùng có thể thích để gợi ý cho họ. Tuy nhiên, điều này có thể gây bất lợi do không phải người dùng lúc nào cũng sẳn sàng để lại các phản hồi của họ, vì vậy hệ thống phải nên tự xác định người dùng cần gì thông qua phản hồi tiềm ẩn. Trong bài viết này, chúng tôi đề xuất một giải pháp xây dựng hệ thống gợi ý cho bán hàng trực tuyến, sử dụng phản hồi tiềm ẩn từ người dùng (như số lần duyệt/xem sản phẩm, số lần mua sản phẩm). Trước hết chúng tôi đề xuất phương pháp thu thập và khai thác thông tin phản hồi tiềm ẩn từ người dùng, sau đó lựa chọn và đề xuất kết hợp các mô hình sử dụng thông tin phản hồi tiềm ẩn. Kế đến là việc xây dựng hệ thống và tích hợp các giải thuật gợi ý vào hệ thống. Sau khi có hệ thống hoàn chỉnh, chúng tôi thu thập dữ liệu từ người dùng thực nhằm đánh giá hiệu quả của hệ thống gợi ý.

# II. HỆ THỐNG GỢI Ý SỬ DỤNG PHẢN HỒI TIỀM ẨN

## A. Hệ thống gợi ý

Hệ thống gợi ý (RS) dựa vào sở thích, thói quen, nhu cầu,... trong quá khứ của người sử dụng để dự đoán sở thích trong tương lai của họ. Trong hệ thống gợi ý người ta quan tâm đến 3 đối tượng: người dùng (user), sản phẩm (item - item gọi chung là mục tin nhưng trong bài viết này liên quan đến gợi ý sản phẩm nên từ đây về sau chúng tôi tạm gọi item là sản phẩm) và các đánh giá sở thích của người dùng trên sản phẩm gọi là rating. Thông thường người ta gọi U là tập tất cả người dùng (users) và u là một người dùng cụ thể nào đó (u∈U). I là tập tất cả các sản phẩm (items) sẽ được gợi ý như máy tính, sách, phim ảnh,.. và i là một sản phẩm cụ thể nào đó (i∈I). I là tập các sản phẩm có thể lên đến hàng trăm, hàng nghìn hoặc thậm chí là hàng triệu sản phẩm trong một số ứng dụng, như việc gợi ý về sách, phim ảnh, âm nhạc. Tương tự như vậy, tập người dùng U cũng có thể rất lớn, lên đến hàng triệu trường hợp. R là một tập hợp các giá trị dùng để ước lượng ‘sở thích’ của người dùng, và rui∈R (R⊂ℜ) là xếp hạng của người dùng u trên sản phẩm i. Giá trị rui có thể được xác định một cách tường minh (explicit feedback) như thông qua việc đánh giá/xếp hạng (ví dụ, rating từ 1 đến 5; hay like/ dislike,…) mà u đã bình chọn cho i – trong trường hợp này gọi là dự đoán xếp hạng (rating prediction) hoặc rui có thể được xác định một cách không tường minh hay còn gọi là tiềm ẩn (implicit feedback) như số lần click chuột, số lần chọn mua sản phẩm, thời gian mà u đã duyệt/xem i,…Nội dung bài viết này, chỉ quan tâm nhiều đến cách xác định rui không tường minh. Các thông tin này được biểu diễn thông qua một ma trận, trong đó mỗi dòng là một người dùng u, mỗi cột là một sản phẩm i, và giao giữa dòng và cột là các đánh giá rating của người dùng như số lần click chuột hay chọn mua sản phẩm,…. Các ô có giá trị là những item mà các user đã xem đến hoặc chọn mua trong quá khứ. Những ô trống là những item chưa được xem đến (điều đáng lưu ý là mỗi user chỉ click xem hoặc chọn mua cho một vài item trong quá khứ, do vậy có rất nhiều ô trống trong ma trận này – còn gọi là ma trận thưa – sparse matrix). Ma trận biểu diễn xếp hạng của người dùng trên sản phẩm (user-item-rating matrix). Nhiệm vụ chính của RS là dựa vào các ô đã có giá trị trong ma trận này (dữ liệu thu được từ quá khứ), để dự đoán các ô còn trống (của user hiện hành), sau đó sắp xếp kết quả dự đoán (ví dụ, từ cao xuống thấp) và chọn ra top-N items theo thứ tự, từ đó gợi ý chúng đến người dùng.

- Gợi ý dựa trên cộng tác: người dùng sẽ nhận gợi ý những sản phẩm được ưa thích xuất phát từ những người có cùng thị hiếu và sở thích với mình. Nhóm này dựa vào các phương pháp chủ yếu:

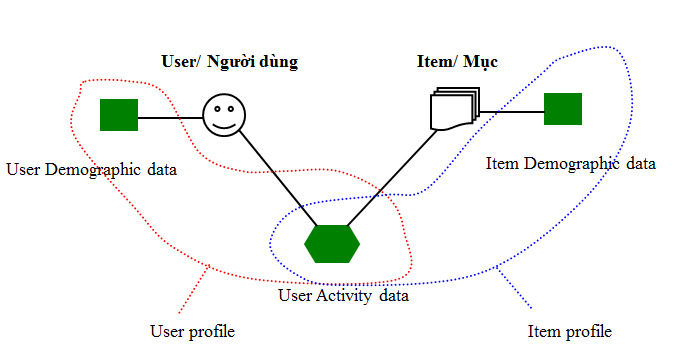
* Phương pháp láng giềng (Neighborhood-based, còn gọi là Memory-based), trong đó hoặc là dựa trên dữ liệu quá khứ của người dùng “tương tự - similarity” (user-based approach), hoặc là dựa trên dữ liệu quá khứ của những item “tương tự” (item-based approach).
* Dựa trên mô hình (Model-based): Nhóm này liên quan đến việc xây dựng các mô hình dự đoán dựa trên dữ liệu thu thập được trong quá khứ. Như mô hình Bayesian, các mô hình nhân tố tiềm ẩn (latent factor models): trong đó kỹ thuật phân rã ma trận (matrix factorization) là một điển hình.
* Gợi ý dựa trên nội dung: người dùng sẽ được gợi ý những sản phẩm tương tự với những sản phẩm đã được người dùng đó ưa thích trước đây. - Gợi ý dựa trên cách tiếp cận kết hợp: kết hợp hai phương pháp tiếp cận dựa trên nội dung và cộng tác. Sau đây chúng tôi tóm lược lại một trong những kỹ thuật trong nhóm lọc cộng tác của hệ thống gợi ý và kỹ thuật sử dụng phản hồi tiềm ẩn, từ đó làm cơ sở cho việc đề xuất mô hình cho hệ thống.

### 1.1 Mục đích

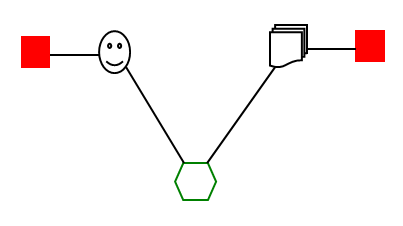
Mục đích của hệ thống gợi ý (RS) là dựa vào sở thích, thói quen, nhu cầu,... trong quá khứ của người sử dụng để dự đoán sở thích trong tương lai của họ. Trong hệ thống gợi ý người ta quan tâm đến 3 đối tượng: người dùng (user), sản phẩm (item) và các đánh giá sở thích của người dùng trên sản phẩm gọi là rating.Tài liệu này chỉ định được dung cho các thành viên trong dự án thực hiện và xác định đúng các chức năng của hệ thống.

### 1.2 Các kỹ thuật gợi ý

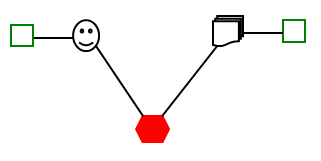
a. Thông tin nào có thể khai thác



Có 2 loại thông tin:

* **Demographic Data**

Là thông tin của Item hoặc User, ví dụ như tên sản phẩm, màu sắc, kích thước, loại sản phẩm, hoặc thông tin của người dùng như là giới tính, tuổi tác… Nhìn chung, loại thông tin này dễ khai thác bởi vì nó sẵn có, khi người dùng đăng kí vào hệ thống thường phải khai báo tên tuối, giới tính.. hoặc khi nhập thông tin sản phẩm thì phải có thông tin chi tiết sản phẩm kèm theo. Tuy nhiên, nó cũng có những nhược điểm nhất định, thứ nhất loại thông tin này có thể ở dưới dạng số hoặc phi số, thông tin dưới dạng số thì dễ dàng xử lý nhưng dưới dàng phi số thì xử lý khó khăn, không tường mình, thứ 2, mức độ phản ánh hành vi các loại thông tin này yếu, không thể hiện hết được mức độ thích hay không thích của người dùng trên sản phẩm cụ thể nào đó.

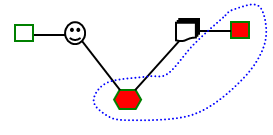
* **Activity Data**

Là thông tin tương tác của người dùng và sản phẩm, ví dụ như sô lượt mua, số lần mua, số lượt xem/ click sản phẩm, hay là phản hồi, thông tin khảo sát do hệ thống đưa ra. Về ưu điểm, loại thông tin này phản ánh mức độ yêu thích sản phẩm mạnh, theo tâm lý người dùng, cứ mua hay xem sản phẩm này có nghĩa là họ quan tâm đến sản phẩm đó, mua càng nhiều hay xem càng nhiều thì mức độ yêu thích càng mạnh. Loại thông tin này chủ yếu được biểu diễn dưới dạng số tuy nhiên nó không sẵn sàng để chúng ta khai thác, nó là những phản hồi tiềm ẩn của người dùng, nếu 1 trang web không thống kê số lượt xem trên 1 sản phẩm tương ứng thì chúng ta mất đi một yếu tố phản ảnh mức độ thích mạnh của người dùng trên sản phẩm đó, chính vì vậy tùy vào từng hệ thống mà chúng ta thiết lập 1 giải thuật khác nhau để đưa ra được sự gợi ý phù hợp.

**b. Các kĩ thuật chính trong RS**

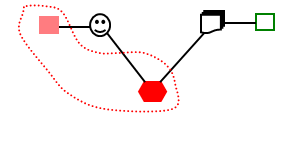
Vì RS là một khái niệm rộng lớn, các kĩ thuật để khai thác cũng rất nhiều, vì vậy trong khuôn khổ bài tập lớn, chúng tôi chỉ trình bày 2 kĩ thuật chính mà nhóm đã áp dụng.

* **Content Based Filtering (CBF)**

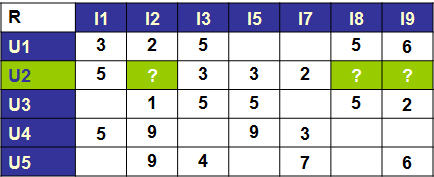
Ở kĩ thuật này, dựa vào thông tin của sản phẩm, hệ thống đưa ra (khuyến nghị) các sản phẩm có đặc điểm giống với sản phẩm mà khách hàng đã mua hoặc từng xem.

Ưu điểm của phương pháp này là nó hoàn toàn trong suốt với người dùng, có nghĩa là nó không bị phụ thuộc vào đối tượng, chỉ cần biết người dùng xem sản phẩm gì chắc chắn hệ thống sẽ gợi ý được sản phẩm tương tự. Nhưng chính vì thế nó cũng là nhược điểm, bởi vì khi đó, khách hàng chỉ được gợi ý ra các sản phẩm giống với sản phẩm đã mua, đã xem chứ các sản phẩm hoàn toàn khác tính chất thì không bao giờ được gợi ý, thứ 2, đối với khách hàng mới đăng kí vào hệ thống, chưa có lịch sử tương tác thì cũng không được gợi ý.

* **Collaborative Filtering (CF)**

Còn được gọi là lọc cộng tác, ở kĩ thuật này, hệ thống sẽ dựa vào sự tương tác của khách hàng đối với sản phẩm để đưa ra gợi ý.

Trong cuộc sống hằng ngày, mọi người thường tin vào những lời giới thiệu từ những người khác thông qua lời nói, thư từ văn bản, các nguồn tin thu được trên các phương tiện thông tin đại chúng. Về bản chất, lọc cộng tác cũng chính là hình thức tư vấn như trên, tuy nhiên, việc này được thực hiện tự động bởi các máy tính. Các nhà phát triển hệ thống gợi ý đầu tiên vào năm 1992 (D. Goldberg et al., 1992) đã đưa ra thuật ngữ “collabrative filtering” (lọc cộng tác). Giả thuyết của lọc cộng tác là: “Nếu người dùng u và u' đánh giá cho n sản phẩm tương tựnhau, hoặc có hành vi tương tựnhau (như: xem, mua, nghe…) thì họ sẽ có các đánh giá tương tự nhau đối với các sản phẩm khác”. Trong hệ thống gợi ý, người ta thường biểu diễn các đánh giá của người dùng cho các sản phẩm qua ma trận gồm một tập người dùng U và tập sản phẩm I như hình dưới:



Chúng tôi dùng các ký tự để phân biệt người dùng và các mục tin: u,u' (đại diện cho người dùng) - i, i' (đại diện cho các sản phẩm). Ký hiệu rui để chỉ mức độ thích của người dùng u cho một sản phẩm i nào đó, chẳng hạn giá trị này trong khoảng từ 1 (không thích) đến 5 (thích nhất) đối với dữ liệu bán hàng, uirˆ là một hàm dùng để dự đoán đánh giá của người dùng u cho sản phẩm i (có thể xem như hàm tiện ích). Theo mô hình láng giềng, việc dự đoán đánh giá của một người dùng trên một mục tin (item) được dựa trên các đánh giá của những người dùng “lân cận” trên mục tin đó. Vì vậy, một độ đo tương tự (similarity) giữa những người dùng cần được định nghĩa trước khi một tập những người dùng láng giềng gần nhất được chọn ra. Hệ tư vấn dựa trên lọc cộng tác (hệ lọc cộng tác) sẽ dự đoán hàm tiện ích của những sản phẩm cho những người dùng cụ thể dựa trên trọng số ban đầu của sản phẩm được cung cấp bởi người dùng khác. Giống như trước, hàm tiện ích uirˆ của sản phẩm i cho người dùng u được đánh giá dựa trên những hàm tiện ích rˆ(uj, i) được gán bởi sản phẩm i và người dùng ujЄ U, đây là những người mà có cùng sở thích và thị hiếu giống u. Chẳng hạn, trong ứng dụng về tư vấn sách, để tư vấn những sách mới cho người dùng u, hệ tư vấn lọc cộng tác sẽ tìm những người tương tự với người dùng u; nghĩa là những người dùng khác nhau nhưng có cùng thị hiếu về sách (trọng số về các sách giống nhau là như nhau). Sau đó, chỉ những quyển sách có độ quan tâm nhất sẽ được tư vấn cho người dùng. Mở rộng của vấn đề này là tìm tập N sản phẩm mà rất có thể người dùng sẽ quan tâm – công việc chính trong các hệ tư vấn hiện nay. Giả sử rằng chúng ta có một hệ thống dự đoán các đánh giá của người dùng lên các sản phẩm mới, và dựa vào đó ta có thể lấy ra N sản phẩm được dự đoán có trọng số cao nhất. Cuối cùng, hai vấn đề quan trọng nhất đó là làm thế nào để tìm được tập K người dùng tương tự với người dùng u nhất dựa trên thị hiếu về sản phẩm hay làm thế nào để tìm ra tập K sản phẩm tương tự với sản phẩm i nhất để có thể đưa ra các dự đoán trọng số đánh giá của người dùng và sản phẩm tương ứng.

Ở phương pháp này, chúng ta chấp nhận giả thiết rằng: ***khách hàng này sẽ có hành vi tương tự hành vi của các khách hàng lân cận.***

Trong phương pháp này, có ưu và nhược điểm riêng:

* Ưu điểm:

Xác suất gợi ý đầu ra cao vì sử dụng các yếu tố có tính phản ánh hành vi mạnh. Chúng ta dễ hàng tìm được sản phẩm mà người dùng có thể muốn mua, ví dụ nếu một người mua kem đánh răng thì thường mua cái bàn chải, vậy nếu một người khác mua kem đánh răng thì hệ thống gợi ý bàn chải thì xác suất sẽ cao hơn việc gợi ý ra cái bàn chải cùng màu sắc…

* Tuy nhiên nó cũng có những nhược điểm sau:

Chậm chạp là nhược điểm đầu tiên mà chúng ta cần nhắc đến, bởi vì hệ thống phải tính toán quá nhiều thứ để đưa ra được gợi ý cho người dùng chứ không dễ dành như phương pháp 1.

Thứ 2, vấn đề “cold start”, hay hiểu một cách cụ thể lả một khách hàng mới đăng kí vào hệ thống thì chưa có lịch sử tương tác, vậy làm cách nào để gợi ý ra cho loại khách hàng này?

Thứ 3, vấn đề thưa thớt dữ liệu. Đã có rất nhiều cố gắng giải quyết vấn đề thưa thớt dữ liệu. Một phương pháp được đưa ra với một số thành công đó là phương pháp nhằm làm giảm chiều ma trận số người dùng – sản phẩm. Chiến lược đơn giản để giảm số chiều là hình thành tập các cụm sản phẩm hoặc người dùng, sau đó sử dụng các cụm này như một thành phần cơ bản trong dự đoán. Trong khuôn khổ báo cáo, chúng tôi sử dụng thuật toán phân cụm K-means để phân cụm người dùng.

**1.3 Thuật toán phân cụm K-means**

Thuật toán K-means clustering do MacQueen giới thiệu trong tài liệu “J. Some Methods for Classification and Analysis of Multivariate Observations” năm 1967.

K-means Clustering là một thuật toán dùng trong các bài toán phân loại/nhóm n đối tượng thành k nhóm dựa trên đặc tính/thuộc tính của đối tượng (k ≤n nguyên, dương).

Về nguyên lý, có n đối tượng, mỗi đối tượng có m thuộc tính, ta phân chia được các đối tượng thành k nhóm dựa trên các thuộc tính của đối tượng bằng việc áp dụng thuật toán này.

Coi mỗi thuộc tính của đối tượng (đối tượng có m thuộc tính) như một toạ độ của không gian m chiều và biểu diễn đối tượng như một điểm của không gian m chiều.

ai =( xi1, xi2, ... xim)

ai (i=1..n) - đối tượng thứ i

xij (i=1..n, j=1..m) - thuộc tính thứ j của đối tượng i

Phương thức phân loại/nhóm dữ liệu thực hiện dựa trên khoảng cách Euclidean nhỏ nhất giữa đối tượng đến phần tử  trung tâm của các nhóm.

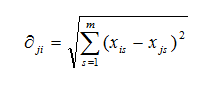
Phần tử trung tâm của nhóm được xác định bằng giá trị trung bình các phần tử trong nhóm.

***Khoảng cách Euclidean.***

   ai=(xi1, xi2,... xim) i=1..n - đối tượng thứ i cần phân phân loại

cj=(xj1, xj2,... xjm) j=1..k - phần tử trung tâm nhóm j

Khoảng cách Euclidean từ đối tượng ai đến phần tử trung tâm nhóm j cj được tính toán dựa trên công thức:

[](http://2.bp.blogspot.com/-_84Xvpqzt0Y/UcuLPf0jL1I/AAAAAAAAA2c/-2kC46N1rOQ/s1600/2.png)

ji - khoảng cách Euclidean từ ai đến cj

xis - thuộc tính thứ s của đối tượng ai

xjs - thuộc tính thứ s của phần tử trung tâm cj

***Phần tử trung tâm.***

k phần tử trung tâm (k nhóm) ban đầu được chọn ngẫu nhiên, sau mỗi lần nhóm các đối tượng vào các nhóm, phần tử trung tâm được tính toán lại.

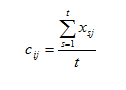
Clusteri = {a1, a2 .... at} – Nhóm thứ i

i=1..k,  k số cluster

j= 1..m,  m số thuộc tính

t - số phần tử hiện có của nhóm thứ i

xsj - thuộc tính thứ j của phần tử s   s=1..t

[](http://2.bp.blogspot.com/-zT5td0Mcnxo/UcuLtDDNPhI/AAAAAAAAA20/kqNAl5AryCI/s1600/33.png)

**Thuật toán k-means bao gồm các bước cơ bản sau :**

**Input:** Số cụm k và các trọng tâm cụm {mj}kj=1.

**Output:** Các cụm C[i] (1 ≤  i ≤  k) và hàm tiêu chuẩn E đạt giá trị tối thiểu.

**Begin**

**Bước 1**: Khởi tạo

Chọn k trọng tâm {mj}kj=1 ban đầu trong không gian Rd (d là số chiều của dữ liệu). Việc lựa chọn này có thể là ngẫu nhiên hoặc theo kinh nghiệm.

**Bước 2:** Tính toán khoảng cách

Đối với mỗi điểm Xi  *(1* ≤ *i* ≤ *n),* tính toán khoảng cách của nó tới mỗi trọng tâm mj *(1* ≤ *j* ≤  *k).* Sau đó tìm trọng tâm gần nhất đối với mỗi điểm.

**Bước 3:** Cập nhật lại trọng tâm

Đối với mỗi *1* ≤ *j* ≤ *k*, cập nhật trọng tâm cụm mj  bằng cách xác định trung bình cộng các vectơ đối tượng dữ liệu.

**Điều kiện dừng:**

Lặp lại các bước 2 và 3 cho đến khi các trọng tâm của cụm không thay đổi.

**End.**

Thuật toán k-means trên được chứng minh là hội tụ và có độ phức tạp tính toán là:

[Description: http://2.bp.blogspot.com/-XSpaa518AYo/UcuMGNBTjeI/AAAAAAAAA28/JocNRjPtBLM/s1600/4.png](http://2.bp.blogspot.com/-XSpaa518AYo/UcuMGNBTjeI/AAAAAAAAA28/JocNRjPtBLM/s1600/4.png)

Trong đó, n là số đối tượng dữ liệu, k là số cụm dữ liệu, d là số chiều, τ là số vòng lặp, Tflop là thời gian để thực hiện một phép tính cơ sở như phép tính nhân, chia,... Như vậy, do k-means phân tích phân cụm đơn giản nên có thể áp dụng đối với tập dữ liệu lớn.Tuy nhiên, nhược điểm của k-means là chỉ áp dụng với dữ liệu có thuộc tính số và khám phá ra các cụm có dạng hình cầu, k-means còn rất nhạy cảm với nhiễu và các phần tử ngoại lai trong dữ liệu. Hơn nữa, chất lượng phân cụm dữ liêuk của thuật toán k-means phụ thuộc nhiều vào các tham số đầu vào như: số cụm k và k trọng tâm khởi tạo ban đầu. Trong trường hợp các trọng tâm khởi tạo ban đầu mà quá lệch so với các trọng tâm cụm tự nhiên thì kết quả phân cụm của k-means là rất thấp, nghĩa là các cụm dữ liệu được khám phá rất lệch so với các cụm trong thực tế. Trên thực tế chưa có một giải pháp tối ưu nào để chọn các tham số đầu vào, giải pháp thường được sử dụng nhất là thử nghiệm với các giá trị đầu vào k khác nhau rồi sau đó chọn giải pháp tốt nhất.

Giống như các thuật toán khác, k- mean cũng có một số hạn chế nhất định:

-         Việc khởi tạo phần tử trung tâm của nhóm ban đầu ảnh hưởng đến sự phân chia đối tượng vào nhóm trong trường hợp dữ liệu không lớn.

-         Số nhóm k luôn phải được xác định trước.

-         Không xác định được rõ ràng vùng của nhóm, cùng 1 đối tượng, nó có thể được đưa vào nhóm này hoặc nhóm khác khi dung lượng dữ liệu thay đổi.

-         Điều kiện khởi tạo có ảnh hưởng lớn đến kết quả. Điều kiện khởi tạo khác nhau có thể cho ra kết quả phân vùng nhóm khác nhau.

-         Không xác định được mức độ ảnh hưởng của thuộc tính đến quá trình tạo nhóm.

Như vậy, với dữ liệu nhỏ, thuật toán có thể có những hạn chế. Để khắc phục những hạn chế này, nên sử dụng thuật toán kmean trong trường hợp dữ liệu lớn.

Về vấn đề hạn chế phân nhóm, có thể dùng phương pháp xác định trung tuyến thay vì xác định mean.

Một quan niệm cho rằng k-means không thể dùng cho dữ liệu có kiểu là định lượng. Điều này không đúng, k-means có thể được dùng giải quyết các bài toán dữ liệu đa biến, thậm chí cho các bài toán có nhiều dạng dữ liệu. Chìa khoá cho việc giải bài toán này của k-means là sử dụng ma trận khoảng cách

**III. XÂY DỰNG HỆ THỐNG**

### 1. Giả thiết

Một website X:

- Chuyên bán hàng

- Nhiều chi nhánh, nhiều địa điểm bán hàng

- Sản phẩm đa dạng, số lượng rất lớn, lên đến hàng trăm triệu sản phẩm.

- Đáp ứng được nhu cầu của một lượng khách hàng lớn, lên đến hàng trăm triệu khách hàng ở mọi độ tuổi khác nhau

Vậy trong một lượng khách hàng lớn với số lượng sản phẩm bán ra nhiều và đa dạng như vậy, làm thế nào để gợi ý cho khách hàng nên mua sản phẩm gì là phù hợp?

Quy trình:

- Website hiển thị danh mục các mặt hàng, các mặt hàng hiện có trong hệ thống.

- Dựa vào sự tương tác, lịch sử giao dịch của người dùng mà hệ thống sẽ gợi ý ra cho khách hàng sản phẩm phù hợp ở thời điểm hiện tại.

- Sau đó, cập nhật lại lịch sử giao dịch của người dùng và tiến hành huấn luyện lại dữ liệu.

Chúng tôi chia người dùng thành 2 trường hợp chính:

User đăng nhập vào hệ thống ngay khi đến hệ thống hoặc user chưa đăng nhập hoặc chưa đăng ký tài khoản hệ thống. Nếu người dùng đăng nhập vào hệ thống ngay từ đầu thì sau mỗi lần xem sản phẩm hệ thống sẽ tự cập nhật lại số lần xem trong cơ sở dữ liệu. Nếu người dùng không đăng nhập hệ thống ngay khi mới đến hệ thống, lúc đó nó sẽ tự động tạo cho người dùng này một account ghi nhận lại số lần xem sản phẩm của họ, cũng như sẽ cập nhật số lần mua cho người dùng này vào tập dữ liệu feedback theo 3 trường hợp:

o Trường hợp 1: nếu user xem xong có đăng nhập vào hệ thống thì số lần xem đó được cập nhật chuyển sang user vừa được đăng nhập đồng thời xóa bỏ account đã tạo tạm.

o Trường hợp 2: nếu user xem xong và đăng ký tài khoản mới thì cập nhật thông tin sang cho user vừa đăng ký, xóa bỏ account đã tạo tạm.

o Trường hợp 3: Nếu user xem và rời khỏi hệ thống thì hệ thống sẽ lưu giữ lại thông tin đó cho một user với thông tin user được lấy từ thông tin địa chỉ IP dùng truy cập vào hệ thống. Trường hợp này, được ghi nhận để lấy thông tin dự đoán toàn cục cho các items. Giá trịdự đoán toàn cục cho item bằng tổng giá trị trung bình dự đoán của mỗi item trên các users. Hệ thống sử dụng giá trị dự đoán toàn cục này để giải quyết vấn đề gợi ý cho người dùng mới (new user).

### 2. Phân tích thiết kế



Yêu cầu của khách hàng được phân tích ra sao:

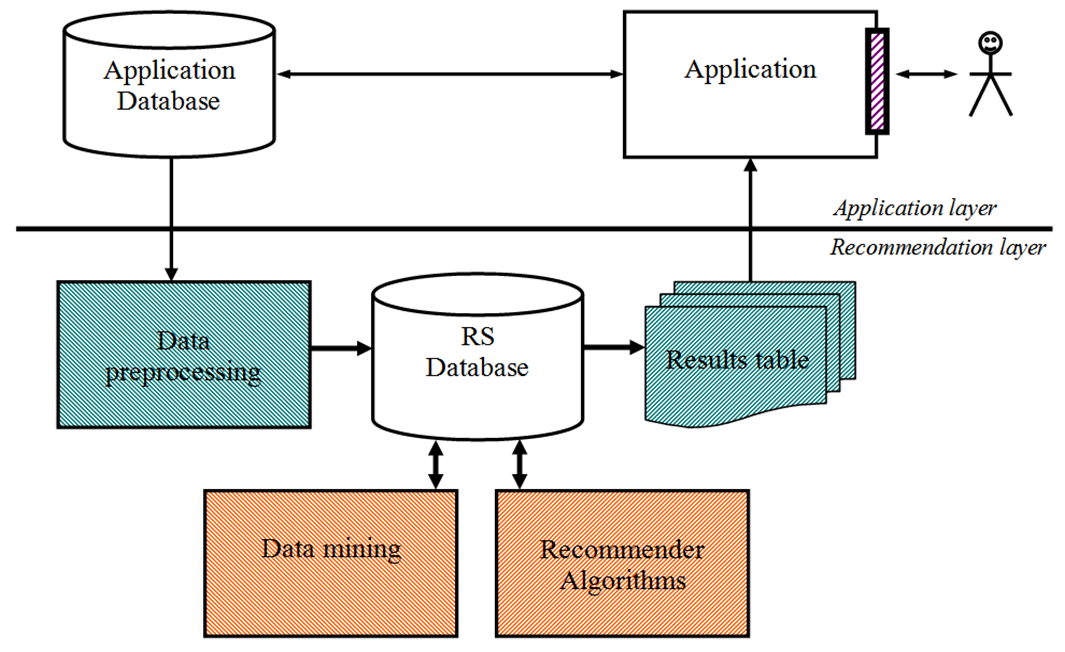
Nếu như các website bán hàng hiện tại chỉ gợi ý cho khách hàng sản phẩm bán chạy, sản phẩm mới nhất của hệ thống thì ở đây chúng tôi chia làm 4 loại gợi ý của 2 loại khách hàng khác nhau.

- R1 & R4: đó là hàng bán chạy và hàng mới trong hệ thống hoặc trong một cụm khách hàng cụ thể.

- R2: Gợi ý theo sản phẩm đã mua

- R3: Gợi ý theo sản phẩm chưa mua.

Từ những phân tích và nhận định trên chúng tôi xây dựng một hệ thống có mô hình như sau:



Tầng App layer, đó là website hiện tại mà chúng tôi giả thiết bên trên, chúng tôi tiến hành phân tích database của website để lấy về database của hệ thống gợi ý các yếu tố mà chúng tôi cần như thông tin của khách hàng, thông tin của sản phẩm và quan trọng nhất là thông tin giao dịch của khách hàng. Tầng RS layer đó là hạ tầng xử lý dữ liệu để đưa ra gợi ý cho khách hàng. Chúng tôi thiết kế một mô hình hoàn toàn độc lập với dữ liệu của website hiện tại, với mục đích không làm ảnh hưởng tới website khi chạy các thuật toán gợi ý. Sau khi xử lý và đưa ra kết quả, chúng tôi đẩy kết quả lên website bằng cách thiết kế một vùng giao diện để chứa kết quả gợi ý.

#### 2.1 Miêu tả tác nhân và các ca sử dụng

##### 2.1.1 Biểu đồ UC

**Biểu đồ UC cho sys**



**Biểu đồ UC cho customer**

****

##### 2.1.2 Biểu đồ lớp tham gia UC

**Biểu đồ cho clustering user**

****

**Biểu đồ recommend product**

****

**Biểu đồ cho retrieve dynamic data**

****

##### 2.1.3 Biểu đồ trình tự

**Biểu đồ trình tự cho clustering user**

****

**Biểu đồ trình tự cho recommend product**

****

**Biểu đồ trình tự cho retrieve dynamic data**

****

**Biểu đồ trình tự cho recommend product**

****

##### 2.2.2 Mô tả tác nhân

|  |  |  |
| --- | --- | --- |
| **#** | **Actor Name** | **Definition & Interests** |
| 1 | Admin | Tương tác với hệ thống sau |
| 2 | Customer | Khách hàng tương tác với trang web |

##### 2.2.3 Mô tả UC

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Name** | **Brief Description** |
| 1 | UC01 | Login | Cho phép actor đăng nhập vào hệ thống |
| 2 | UC02 | Clustering user | Cho phép actor thực hiện việc phân cụm người dùng |
| 3 | UC03 | Retrieve dynamic data | Cho phép lấy dữ liệu |
| 4 | UC04 | Recommend product | Cho phép gợi ý sản phẩm |
| 5 | UC05 | Recommend ads | Cho phép gợi ý quảng cáo |
| 6 | UC06 | Analyse interactive user | Cho phép phân tích tương tác người dùng |
| 7 | UC07 | Analyse transaction | Cho phép phân tích được lịc sử giao dịch người dùng |
| 8 | UC08 | Analyse comment | Cho phép phân tích bình luận |
| 9 | UC09 | Like product | Cho phép actor like sản phẩm |
| 10 | UC10 | Purchase product | Cho phép actor mua hàng |
| 11 | UC11 | Shipping product | Cho phép actor đặt hàng |
| 12 | UC12 | Rating product | Cho phép actor bình chọn sản phẩm |
| 13 | UC13 | View product | Cho phép actor xem sản phẩm |
| 14 | UC14 | Comment product | Cho phép actor bình luận sản phẩm |
| 15 | UC15 | Sign in | Cho phép actor đăng nhập |

##### 2.2.4 Biểu đồ UC và Actor

|  |  |  |
| --- | --- | --- |
| **Actor**  **Use Case** | **Customer** | **Admin** |
| UC01 : Login | **x** |  |
| UC02 : Clustering user |  | **x** |
| UC03 : Retrieve dynamic data |  | **x** |
| UC04 : Recommend product |  | **x** |
| UC05: Recommend ads |  | **x** |
| UC06 : Analyse interactive user |  | **x** |
| UC07 : Analyse transaction |  | **x** |
| UC08 : Analyse comment |  | **x** |
| UC09 : Like product | **x** |  |
| UC10 : Purchase product | **x** |  |
| UC11 : Shipping product | **x** |  |
| UC12 : Rating product | **x** |  |
| UC13 : View product | **x** |  |
| UC14 : Comment product | **x** |  |
| UC15 : Sign in | **x** |  |

#### 2.3 Môi trường hệ điều hành

Trong tài liệu này chúng tôi đê xuất thông tin cần thiết sau:

OS: yêu cầu HDH window XP trở lên. Đề xuất window 7

Kĩ thuật: RAM 1Gb trở lên, HDD: 10Gb trở lên, chip : Pentium trở lên, Xung: 1Ghz trở lên.

Công cụ: Visual studio, Microsoft SQL server, xmind, MS project, MS visio, Rational Rose. Đề xuất: visual studio 2012 , MS SQL server 2012,

#### 2.4 Giả thiết và các điều kiện lệ thuộc

Các công cụ làm việc tốt, máy móc không bị hỏng hóc.

### 3. Yêu cầu chức năng

#### 3.1 Login

Mô tả UC

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Login | **Code** | UC01 |
| **Description** | Cho phép actor đăng nhập vào hệ thống | | |
| **Actor** | Admin/customer | **Trigger** | Actor bấm nút login |
| **Pre-condition** |  | | |
| **Post condition** | Chuyển tới trang default với role tương ứng | | |

Hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập tên đăng nhập/mật khẩu và click vào button Login trên trang đăng nhập . |  |  |
|  |  | 2 | Kiểm tra tên đăng nhập/mật khẩu là chính xác sau đó chuyển tới trang default role tương ứng |

Thông điệp

|  |  |
| --- | --- |
| MS01 | “Tên Đăng Nhập và/hoặc Mật Khẩu của bạn không chính xác. Vui lòng kiểm tra và thử lại.”  Message thông báo khi actor nhập sai tên đăng nhập/mật khẩu |

#### 3.2 Clustering customer

Mô tả UC

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Clusteringcustomers | **Code** | UC13 |
| **Description** | Cho phép actor thực hiện công việc thao tác vào hệ thống phân loại khách hàng | | |
| **Actor** | Admin | **Trigger** |  |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Quá trình hệ thống làm việc đúng trả lại thành công | | |

Hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Clustering customers successfully** | | | |
| 1 | Actor sẽ tiến hành đăng nhập vào hệ thống |  |  |
|  |  | 2 | Hiển thị thông báo log thành công |
| 3 | Tiến hành chạy service |  |  |
|  |  | 4 | Hệ thống đang xử lí |
|  |  | 5 | Tiến hành chạy ghi log file và kết thúc là hiển thị thành công |

Thông điệp

|  |  |
| --- | --- |
|  |  |

#### 3.3 Recommend product

Miêu tả UC

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Recommend product | **Code** | UC14 |
| **Description** | Cho phép gợi ý sản phẩm cho khách hàng | | |
| **Actor** | Admin | **Trigger** | Chạy service |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Hệ thống chạy thành công. | | |

Hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Recommend product** | | | |
| 1 | Chuyển tiếp sang recommend product |  |  |
|  |  | 2 | Chuyển hướng lựa chọn theo user |
|  |  | 3 | Tiến hành xử lí, ghi log nội dung output và thông báo cho người dùng biết đầu ra khi hoàn tất |

**3.4 Retrieve dynamic data**

* Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Retrieve dynamic data | **Code** | UC15 |
| **Description** | Cho phép lấy dữ liệu từ các db của các trang web khác nhau | | |
| **Actor** | Admin | **Trigger** | Tiến hành thay đổi 1 số câu lệnh theo mặc định cho phép |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Trả về kết quả thành công | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: retrieve dynamic data successfully** | | | |
| 1 | Chuyển tiếp sang recommend product |  |  |
|  |  | 2 | Chuyển hướng lựa chọn theo user |
|  |  | 3 | Tiến hành xử lí, ghi log nội dung output và thông báo cho người dùng biết đầu ra khi hoàn tất |

### 4.Thiết kế

#### 4.1 Thiết kế cơ sở dữ liệu

Lập thành các bảng để lưu trữ bảng trong CSDL. Phân theo module ta được như sau:

Module lấy dữ liệu động

* RS\_Item

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ItemID | Varchar(50) not null | Primary key |
| 2 | ItemName | Varchar(50) not null |  |
| 3 | MetaItemID | Varchar(50) not null |  |
| 4 | DataID | Varchar(50) not null | Primary key |
| 5 | CreatedDate | datetime |  |
| 6 | ItemCategoryCode | Varchar(50) not null |  |
| 7 | ItemFamillyCode | Varchar(50) not null |  |

* RS\_Users

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | UserID | Varchar(50) not null | Primary key |
| 2 | UserName | Varchar(50) not null |  |
| 3 | U\_SubCategoryID | Varchar(50) not null |  |
| 4 | U\_CategoryID | Varchar(50) not null |  |
| 5 | CreateDate | datetime |  |
| 6 | CA\_EUROS | float |  |
| 7 | DataID | Varchar(50) not null |  |

* RS\_Transaction

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | UserID | Varchar(50) not null | Primary key |
| 2 | ItemID | Varchar(50) not null | Primary key |
| 3 | Quantity | int |  |
| 4 | Date | datetime | Primary key |

* DI.DATA\_SOURCE

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | DataID | Varchar(50) not null | Primary key |
| 2 | CreationDate | datetime |  |
| 3 | ConnectionString | Varchar(MAX) not null |  |
| 4 | Type | Varchar(50) not null |  |

* DI.QUERY

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | QueryID | Varchar(50) not null | Primary key |
| 2 | DescTable | Varchar(50) not null |  |
| 3 | OrderNo | int |  |
| 4 | String | Varchar(MAX) not null |  |
| 5 | Description | Varchar(MAX) not null |  |
| 6 | DataID | Varchar(50) not null |  |

* DI.PARAMETER

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ParameterID | Varchar(50) not null | Primary key |
| 2 | QueryID | Varchar(50) not null | Primary key |
| 3 | Value | Varchar(MAX) not null |  |
| 4 | Type | Varchar(10) not null |  |

Module phân cụm user

* RS.Rating\_Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ClusterID | Varchar(50) not null |  |
| 2 | UserID | Varchar(50) not null | Primary key |
| 3 | MetaItemID | Varchar(50) not null | Primary key |
| 4 | Value | float |  |

* RS.Clustering\_User

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ClusterID | Varchar(50) not null | Primary key |
| 2 | U\_SubCategoryID | Varchar(50) not null |  |

* RS.Cluster\_Setting

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | Keys | Varchar(50) not null | Primary key |
| 2 | DataType | Varchar(50) not null |  |
| 3 | Description | Varchar(50) not null |  |

* RS.Cluster\_Setting\_Value

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ValueID | int | Primary key |
| 2 | Keys | Varchar(50) not null |  |
| 3 | Value | int |  |
| 4 | isDefault | bit |  |

Module gợi ý

* RS.Rating\_Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ClusterID | Varchar(50) not null |  |
| 2 | UserID | Varchar(50) not null | Primary key |
| 3 | MetaItemID | Varchar(50) not null | Primary key |
| 4 | Value | float |  |

* RS.Recommend\_Setting\_TBL

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | [Key] | Varchar(50) not null | Primary key |
| 2 | DataType | Varchar(50) not null |  |
| 3 | Description | Varchar(MAX) not null |  |

* RECOMMEND\_SETTING\_VALUE\_TBL

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ValueID | int | Primary key |
| 2 | Keys | Varchar(50) not null |  |
| 3 | Value | int |  |
| 4 | isDefault | bit |  |

* RECOMMENDATION\_SCHEDULE\_TBL

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ScheduleID | int | Primary key |
| 2 | StartTime | datetime |  |
| 3 | StopTime | datetime |  |
| 4 | [Log] | varchar(MAX) |  |
| 5 | LoginID | varchar(10) |  |

* RECOMMENDATION\_TBL

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | RecomID | int | Primary key |
| 2 | ScheduleID | int |  |
| 3 | MetaItemID | varchar(50) |  |
| 4 | UserID | varchar(50) |  |
| 5 | Quantity | float |  |
| 6 | Score | float |  |
| 7 | RecommendType | varchar(50) |  |
| 8 | ItemFamillyCode | varchar(50) |  |

CONFIDENT\_MATRIX

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ClusterID | varchar(50) |  |
| 2 | MetaItemSource | varchar(50) |  |
| 3 | MetaItemDestination | varchar(50) |  |
| 4 | Confident | float |  |

* DISTANCE\_MATRIX

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ClusterID | varchar(50) |  |
| 2 | MetaItemSource | varchar(50) |  |
| 3 | MetaItemDestination | varchar(50) |  |
| 4 | Distance | float |  |

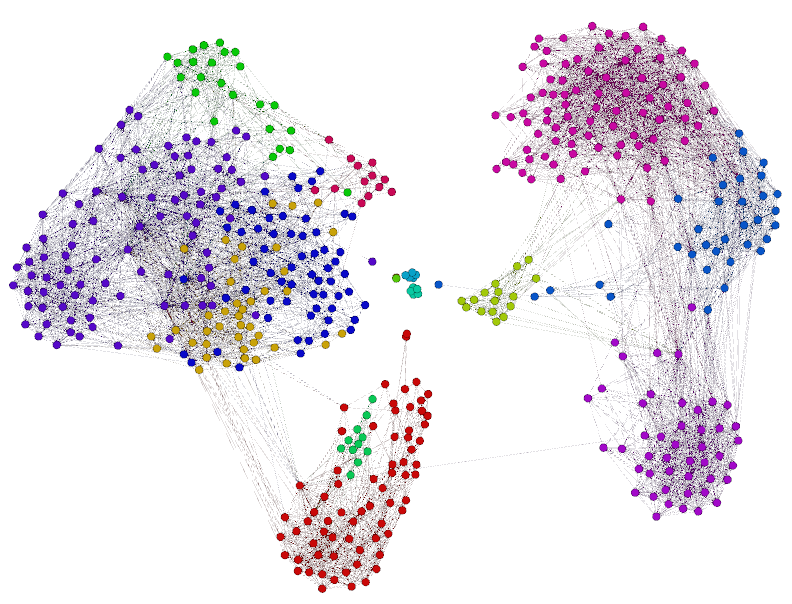
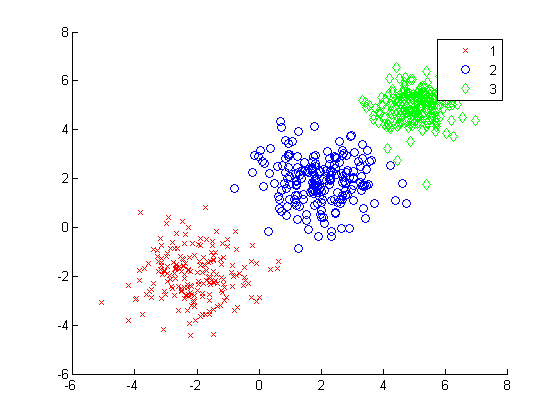
#### 4.2 Thiết kế chương trình và đưa ra giải pháp khắc phục các nhược điểm.

1. Vấn đề lấy dữ liệu của khách hàng.

Vì đặc thì của hệ thống gợi ý là xử lý dữ liệu hiện có của khách hàng, và thiết kế 1 database chạy độc lập với website hiện tại, vì vậy chúng tôi đã nghĩ ra một module lấy dữ liệu ở mức động, có nghĩa là khi chạy hệ thống cho một website khác, số thao tác cần sửa là ít nhất. Phương pháp của chúng tôi rất đơn giản, chúng tôi sẽ tạo ra một database mẫu với đầy đủ các thuộc tính cần thiết nhất, sau đó các trường của db này sẽ được mapping với các trường của db website, chúng tôi chỉ việc sửa câu lệnh select thông tin cần khai thác chứ không cần vào sửa code của chương trình.

2. Vấn đề chậm chạp và thưa thớt dữ liệu.

Nhược điểm của phương pháp lọc cộng tác là chậm chạp vì phải khai thác nhiều thông tin, hơn nữa vấn đề thưa thớt dữ liệu là điều không tránh khỏi vì vậy chúng tôi áp dụng thuật toán phân cụm dữ liệu K-Means để lọc ra các cụm người dùng có sở thích giống nhau, từ đó chúng tôi sẽ chạy tiến trình gợi ý trên mỗi cụm nhỏ, vì vậy sẽ giải quyết luôn được tình trạng chậm chạp vì phải xử lý quá nhiều thông tin.



**Step 1.** Khởi tạo một phân hoạch ngẫu nhiên

**Step 2.** Tính tâm của các cụm Ci, i = 1,2,…k từ phân hoạch hiện có

**Step 3.** Tính lại phân hoạch:

Với mỗi điểm dữ liệu t

- Tính khoảng cách từ t tới tất cả các tâm

- Điểm dữ liệu t được phân vào nhóm i nếu nó gần tâm Ci nhất

**Step 4.** Nếu thỏa mãn điều kiện hội tụ 🡪 STOP

Ngược lại: lặp lại Step 2.

**Điều kiện dừng:** sử dụng một hoặc một số điều kiện sau

**- Số lần lặp lớn hơn 1 giá trị cho trước (Max Num Iterations)**

**- Tâm mới không thay đổi giữa các lần lặp**

**- Giá trị giảm của hàm mục tiêu F nhỏ hơn một ngưỡng cho trước**

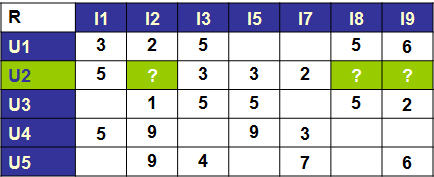


### 

3. Một công thức hiệu quả cho ma trận rate

Sau khi phân cụm dữ liệu xong, vấn đề quan trọng là phải thiết lập được một ma trận rate để từ đây xếp hạng và đưa ra gợi ý cho người dùng, vậy vấn đề tính các phần tử của ma trận đó như thế nào?

Sau đây chúng tôi đề xuất một công thức tổng quát như sau:



Trong đó: a, b, c là các hệ số thể hiện mức độ quan trọng của từng yếu tố

x, y, z tương ứng là số lượt mua, số lượt xem, số lượng mua sản phẩm

Hệ số k là hệ số trung hòa của cả ma trận rate

### 5. Lập trình

Để thuận lợi cho việc phát triển có hiệu quả.Nhóm đã phân chia công việc lập trình như sau:

|  |  |
| --- | --- |
| Mô tả công việc | Thành viên |
| Lấy dữ liệu động | Vũ Tuấn Khắc –Vũ Tiến Dũng |
| Xây dựng công thức | Dương Lê Phúc Nguyện –Vũ Tiến Dũng |
| Phân cụm người dùng | Dương Lê Phúc Nguyện –Vũ Tiến Dũng |
| Gợi ý sản phẩm | Vũ Tuấn Khắc - Dương Lê Phúc Nguyện |

### 6. Kiểm thử

Các công việc kiểm thử sẽ tiến hành song song với công việc lập trình. Do tính chất công việc nên nhóm quyết định đê đội coder lập trình và kiểm thử đơn giản và sau đó sẽ thuê 1 nhân viên kiểm thử chuyên để kiểm tra trong quá trình viết mã.

### 7. Vận hành bảo trì

Quá trình vận hành sẽ được tiến hành ngay sau khi hoàn tất và công việc bảo trì là diễn ra theo định kì tháng/lần.

**KẾT LUẬN**

Bài viết này đã đềxuất một giải pháp xây dựng hệ thống gợi ý sản phẩm trong bán hàng trực tuyến dựa trên phản hồi tiềm ẩn của người dùng. Trước hết chúng tôi đề xuất phương pháp thu thập thông tin phản hồi tiềm ẩn, sau đó đề xuất phương pháp sử dụng các thông tin này, đồng thời cài đặt các mô hình, điều chỉnh và tích hợp vào hệ thống nhằm gợi ý các sản phẩm phù hợp với sở thích của người dùng.

Để đánh giá phương pháp đã được đề xuất, chúng tôi đã xây dựng hệ thống và thu thập phản hồi từ người dùng thực. Kết quả thực nghiệm cho thấy giải pháp tích hợp các thông tin phản hồi tiềm ẩn cho độ lỗi thấp hơn chỉ sử dụng một thông tin đơn lẻ như trong các hệ thống gợi ý khác, đồng thời khả năng mà hệ thống gợi ý phù hợp với sở thích của từng đối tượng người dùng là khá tốt, vì vậy giải pháp được đề xuất hoàn toàn có thể ứng dụng cho các trang web bán hàng trực tuyến hiện nay.