PHẦN MỞ ĐẦU

1. Tính cấp thiết của luận án

Lọc thông tin là lĩnh vực nghiên cứu các quá trình phân bổ thông tin thích hợp và gỡ bỏ thông tin không thích hợp đến với mỗi người dùng. Lọc thông tin cho các hệ tư vấn được tiếp cận theo hai xu hướng chính, đó là lọc dựa vào nội dung sản phẩm và lọc dựa vào thói quen sử dụng sản phẩm của người hay còn được gọi là lọc cộng tác. So với lọc theo nội dung, lọc cộng tác cho lại kết quả tốt hơn và có thể lọc bất kỳ dạng thông tin nào. Tuy nhiên, lọc cộng tác gặp phải vấn đề dữ liệu thưa, người dùng mới và sản phẩm mới cần được tiếp tục nghiên cứu giải quyết.

Kết hợp giữa lọc cộng tác và lọc nội dung để nâng cao chất lượng dự đoán và tránh hiện trạng dữ liệu thưa của lọc cộng tác được tập trung nghiên cứu nhiều trong thời gian gần đây. Các phương pháp lọc kết hợp hiện nay vẫn hạn chế trong biểu diễn và ước lượng mức độ ảnh hưởng của mỗi đặc trưng nội dung đến thói quen sử dụng sản phẩm của người dùng.

Đề tài "*Phát triển một số phương pháp lọc thông tin cho hệ tư vấn*" được thực hiện trong khuôn khổ luận án tiến sĩ chuyên ngành khoa học máy tính nhằm góp phần giải quyết một số vấn đề còn tồn tại trong lọc cộng tác và lọc kết hợp.

2. Mục tiêu của luận án

Mục tiêu của luận án là nghiên cứu áp dụng, cải tiến các kỹ thuật học máy nhằm nâng cao độ chính xác của lọc thông tin trong các hệ tư vấn. Đặc biệt, nghiên cứu tập trung vào việc nâng cao kết quả dự đoán nhu cầu người dùng trong trường hợp dữ liệu thưa, cũng như trong trường hợp có cả dữ liệu sở thích và thông tin nội dung.

3. Các đóng góp của luận án

Luận án nghiên cứu và đề xuất được hai kết quả chính, đó là hạn chế ảnh hưởng của vấn đề dữ liệu thưa trong lọc cộng tác bằng phương pháp học đa nhiệm và phương pháp kết hợp giữa lọc cộng tác và lọc nội dung dựa vào mô hình đồ thị.

4. Bố cục của luận án

Bố cục luận án được xây dựng thành ba chương và một phụ lục, trong đó:

Chương 1 giới thiệu tổng quan về lọc thông tin.

Chương 2 trình bày phương pháp hạn chế ảnh hưởng của tình trạng dữ liệu thưa bằng phương pháp học đa nhiệm.

Chương 3 trình bày phương pháp kết hợp giữa lọc cộng tác và lọc nội dung dựa trên mô hình đồ thị.

Phần phụ lục trình bày thiết kế và xây dựng ứng dụng cho phương pháp lọc kết hợp được đề xuất trong Chương 3. Cuối cùng là một số kết luận và đề xuất các nghiên cứu tiếp theo.

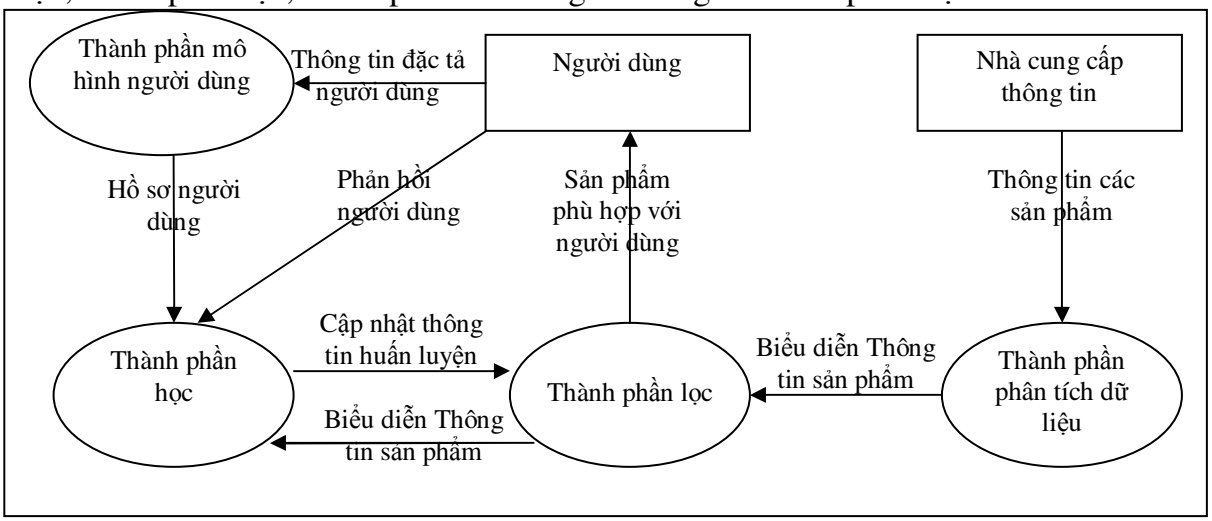
CHƯƠNG 1
TỔNG QUAN VỀ LỌC THÔNG TIN

1.1. GIỚI THIỆU CHUNG

Lọc thông tin là lĩnh vực nghiên cứu các quá trình phân bổ thông tin thích hợp, ngăn ngừa và gỡ bỏ thông tin không thích hợp cho mỗi người dùng. Thông tin được phân bổ (còn được gọi là sản phẩm) có thể là văn bản, trang web, phim ảnh, dịch vụ, phim hoặc bất kỳ dạng thông tin nào được sản sinh ra từ các phương tiện truyền thông.

1.1.1. Kiến trúc tổng quát của hệ thống lọc thông tin

Một hệ thống lọc thông tin tổng quát bao gồm bốn thành phần cơ bản: thành phần phân tích dữ liệu, thành phần lọc, thành phần mô tả người dùng và thành phần học.



Hình 1.1. *Kiến trúc tổng quát của hệ thống lọc thông tin*

1.1.2. Lọc thông tin và truy vấn thông tin

Một số thành phần của hệ thống lọc có thể được tìm thấy trong các hệ thống truy vấn thông tin. Tuy nhiên, ta có thể phân biệt sự khác biệt giữa hệ thống lọc thông tin với các hệ thống khác thông qua những đặc trưng liên quan đến người dùng, sản phẩm và phương pháp thực hiện.

1.1.3. Học máy và lọc thông tin

Thành phần lọc thông tin được xây dựng theo hai cách tiếp cận chính: *lọc dựa trên tri thức và lọc dựa trên dữ liệu*. Đối với lọc dựa trên tri thức, thông tin được lọc bằng cách sử dụng các luật. Mỗi luật biểu diễn nhu cầu thông tin người dùng hoặc một mẫu thông tin cần lọc. Mỗi quyết định lọc sẽ được thực hiện nếu những điều kiện của luật đưa ra được thỏa mãn. Khác với lọc dựa trên tri thức, trong cách tiếp cận dựa trên dữ liệu, các quy tắc cho thành phần lọc được xây dựng từ dữ liệu mà hệ thống thu thập được bằng cách sử dụng kỹ thuật thống kê hoặc các thuật toán học máy. Cách tiếp cận này cho phép tạo ra và cập nhật quy tắc lọc thông tin mà không cần tới tri thức chuyên gia, đồng thời chất lượng lọc có thể tốt hơn so với cách tiếp cận dựa trên tri thức, đặc biệt khi có lượng dữ liệu lớn và chất lượng. So với lọc dựa vào tri thức, lọc dựa vào dữ liệu được quan tâm nghiên cứu nhiều hơn.

1.1.4. Lọc thông tin và các hệ tư vấn

Hệ tư vấn (RS) đang phát triển và được sử dụng rộng rãi trong nhiều ứng dụng khác nhau của khoa học máy tính nhằm gợi ý, giới thiệu hàng hóa, dịch vụ, thông tin tiềm năng đến với người dùng. Các hệ tư vấn được phân loại dựa vào phương pháp lọc được áp dụng, bao gồm: tư vấn dựa vào phương pháp lọc nội dung, tư vấn dựa vào phương pháp lọc cộng tác và tư vấn dựa vào phương pháp lọc kết hợp.

1.2. PHƯƠNG PHÁP LỌC THEO NỘI DUNG

Lọc theo nội dung là phương pháp thực hiện dựa trên việc so sánh nội dung thông tin hay mô tả hàng hóa, để tìm ra những sản phẩm tương tự với những gì mà người dùng đã từng quan tâm để giới thiệu cho họ những sản phẩm này. Các phương pháp tiếp cận cho lọc theo nội dung được chia thành hai phương pháp chính: lọc nội dung dựa vào bộ nhớ và lọc nội dung dựa vào mô hình. Những vấn đề cần tiếp tục nghiên cứu của lọc nội dung là vấn đề trích chọn đặc trưng và người dùng mới.

1.3. PHƯƠNG PHÁP LỌC CỘNG TÁC

Lọc cộng tác khai thác những khía cạnh liên quan đến thói quen sở thích của người sử dụng sản phẩm để đưa ra dự đoán và phân bổ các sản phẩm cho người dùng này. Các phương pháp tiếp cận cho lọc cộng tác cũng được chia thành hai phương pháp chính: lọc cộng tác dựa vào bộ nhớ và lọc cộng tác dựa vào mô hình. Những vấn đề cần tiếp tục nghiên cứu của lọc cộng tác là vấn đề dữ liệu thưa, vấn đề người dùng mới và sản phẩm mới.

1.4. PHƯƠNG PHÁP LỌC KẾT HỢP

Lọc kết hợp là phương pháp kết hợp giữa lọc cộng tác và lọc nội dung, nhằm tận dụng lợi thế và tránh những hạn chế của mỗi phương pháp. Lọc kết hợp được tiếp cận theo bốn xu hướng chính: Kết hợp tuyến tính, kết hợp đặc tính của lọc nội dung vào lọc cộng tác, kết hợp đặc tính của lọc cộng tác vào lọc nội dung và xây dựng mô hình hợp nhất cho cả lọc cộng tác và lọc nội dung. Vấn đề cần tiếp tục nghiên cứu của lọc kết hợp là nâng cao hiệu quả phương pháp biểu diễn và dự đoán cho mô hình kết hợp.

1.6. KẾT LUẬN

Lọc theo nội dung thực hiện hiệu quả với các dạng thông tin được biểu diễn dưới dạng các đặc trưng nội dung nhưng lại khó lọc được các dạng thông tin đa phương tiện. Lọc cộng tác cho lại kết quả tốt hơn so với lọc nội dung và có thể lọc bất kỳ dạng thông tin nào nhưng gặp phải khó khăn trong trường hợp dữ liệu thưa, người dùng mới và sản phẩm mới. Lọc kết hợp chỉ phát huy hiệu quả nếu phương pháp kết hợp giải quyết được những mâu thuẫn trong dự đoán theo lọc nội dung và lọc cộng tác. Chính vì vậy, trọng tâm nghiên cứu của luận án là vấn dữ liệu thưa của lọc cộng tác và vấn đề kết hợp hiệu quả giữa lọc cộng tác và lọc nội dung.

CHƯƠNG 2
LỌC CỘNG TÁC BẰNG PHƯƠNG PHÁP HỌC ĐA NHIỆM

2.1. ĐẶT VẤN ĐỀ

Giả sử hệ gồm N người dùng $U = \{u_1, \ldots, u_N\}$, M sản phẩm $P = \{p_1, p_2, \ldots, p_M\}$ với ma trận đánh giá $R = (r_{ij})$. Nhiệm vụ của lọc cộng tác là xây dựng phương pháp dự đoán và phân bổ cho người dùng hiện thời u_a các sản phẩm phù hợp nhất với u_a chưa được đánh giá dựa trên ma trận đánh giá $R = (r_{ij})$. Đối với các hệ thống lọc cộng tác, số lượng người dùng $|U|$ và số lượng sản phẩm $|P|$ là rất lớn. Tuy vậy, mỗi người dùng chỉ đưa ra một số rất ít các đánh giá của mình trong tập các sản phẩm. Điều này làm cho ma trận đầu vào r_{ij} có số các đánh giá $r_{ij} \neq \varnothing$ nhỏ hơn rất nhiều lần số các đánh giá $r_{ij} = \varnothing$. Lọc cộng tác gọi vấn đề này là vấn đề dữ liệu thưa.

Vấn đề dữ liệu thưa làm cho nhiều cặp người dùng không xác định được mức độ tương tự và việc xác định tập hàng xóm cho mỗi người dùng trở nên kém tin cậy. Đặc biệt, vấn đề người dùng mới cần có những đánh giá ban đầu.

2.2. LỌC CỘNG TÁC BẰNG PHÂN LOẠI

Bài toán lọc cộng tác có thể phát biểu như bài toán phân loại tự động của học máy. Dựa trên đánh giá của người dùng về những sản phẩm khác nhau, với mỗi người dùng, một mô hình phân loại sẽ được xây dựng và huấn luyện. Mô hình này sau đó được sử dụng để phân chia sản phẩm mới thành các loại khác nhau, ví dụ như loại "*phù hợp*" và "*không phù hợp*". Tương tự như vậy, có thể thay đổi vai trò giữa người dùng và sản phẩm và xây dựng bộ phân loại cho phép dự đoán một sản phẩm cụ thể có "*phù hợp*" hay "*không phù hợp*" đối với người dùng.

2.2.1. Phát biểu bài toán lọc cộng tác bằng phân loại

Cho ma trận đánh giá người dùng $R = (r_{ij})$ như được trình bày trong mục 2.1. Các hàng của ma trận tương ứng với tập người dùng; các cột của ma trận tương ứng với tập sản phẩm; các phần tử r_{ij} của ma trận tương ứng với đánh giá của người dùng đối với sản phẩm. Thông thường, mỗi người dùng chỉ đánh giá một tập rất nhỏ các mặt hàng và do vậy đa số các giá trị r_{ij} được để trống. Nhiệm vụ của các phương pháp phân loại là điền vào hay dự đoán các giá trị thích hợp vào các ô trống cho mỗi hàng của ma trận đánh giá.

Để thực hiện dự đoán, một bộ phân loại sẽ được xây dựng riêng cho mỗi người dùng. Mỗi bộ phân loại dự đoán các giá trị rỗng cho một hàng của ma trận đánh giá. Mỗi bộ phân loại thực hiện huấn luyện trên tập các ví dụ huấn luyện; mỗi ví dụ huấn luyện được biểu diễn dưới dạng một véc tơ đặc trưng; mỗi đặc trưng tương ứng với một người dùng khác người dùng cần dự đoán. Giá trị của đặc trưng là giá trị các ô của ma trận đánh giá. Nhãn phân loại cho các ví dụ huấn luyện là các đánh giá khác $\varnothing$ của người dùng hiện thời.

2.2.2. Phân loại bằng phương pháp Boosting

Boosting là phương pháp học máy cho phép tạo ra bộ phân loại có độ chính xác cao bằng cách kết hợp nhiều bộ phân loại có độ chính xác kém hơn hay còn được gọi là bộ phân loại yếu.

Dựa trên nguyên tắc chung này, nhiều phiên bản khác nhau của kỹ thuật Boosting đã được đề xuất và sử dụng. Luận án này sử dụng phiên bản *Gentle AdaBoost* (viết tắt là GentleBoost) được Friedman đề xuất do các ưu điểm của phương pháp này là đơn giản, ổn định, và cho kết quả phân loại tốt trong nhiều ứng dụng.

Phương pháp GentleBoost cho trường hợp phân loại hai lớp có thể mô tả tóm tắt như sau. Cho tập dữ liệu huấn luyện bao gồm M ví dụ (x_1, y_1), …, (x_M, y_M) với x_i là vectơ các đặc trưng và y_i là nhãn phân loại nhận giá trị $y_i = +1$ hoặc $y_i = -1$ (tương ứng với "*thích hợp*" và "*không thích hợp*"). Bộ phân loại mạnh $F(x)$ được tạo thành bằng cách tổ hợp tuyến tính $F(x) = \sum_{k=1}^{K} f_k(x)$, trong đó $f_k(x)$ là bộ phân loại yếu có khả năng dự đoán nhãn phân loại cho vec tơ đầu vào x. Kết quả phân loại cuối cùng được tạo ra bằng cách tính sign($F(x)$). Thuật toán bao gồm K vòng lặp được thể hiện trong hình 2.1 dưới đây.

Đầu vào:

- Tập dữ liệu huấn luyện gồm M ví dụ (x_1, y_1),.., (x_M, y_M) với x_i là vectơ các đặc trưng và y_i là nhãn phân loại nhận giá trị $y_i = +1$ hoặc $y_i = -1$.

Đầu ra:

- *Trả lại* $\text{sign}[F(x)] = \text{sign}[\sum_{k=1}^{K} f_k(x)]$

Các bước thực hiện:

1. Khởi tạo các trọng số $w_i = 1/M$, $i = 1..M$, w_i là trọng số của ví dụ huấn luyện thứ i.
   Khởi tạo $F(x) = 0$
2. Lặp với $k = 1, 2, \ldots, K$
   a. Huấn luyện $f_k(x)$ sử dụng dữ liệu huấn luyện có trọng số
   b. Cập nhật $F(x) \leftarrow F(x) + f_k(x)$
   c. Cập nhật trọng số $w_i \leftarrow w_i e^{-y_i f_k(x_i)}$ và chuẩn tắc hoá trọng số
3. Trả về bộ phân loại $\text{sign}[F(x)] = \text{sign}[\sum_{k=1}^{K} f_k(x)]$

Hình 2.1. *Thuật toán GentleBoost.*

Tại bước (a) của mỗi vòng lặp, thuật toán lựa chọn $f_k(x)$ sao cho sai số phân loại dưới đây là nhỏ nhất:

$$J = \sum_{i=1}^{M} w_i (y_i - f_k(x_i))^2 \qquad (2.1)$$

Để tìm được bộ phân loại cho phép cực tiểu hoá (2.1), cần xác định bộ phân loại yếu $f_k(x)$ cho phép cực tiểu hoá bình phương lỗi phân loại có tính tới trọng số. Ở đây, bộ phân loại yếu được sử dụng là gốc quyết định. Gốc quyết định là phiên bản đơn giản của cây quyết định với một nút duy nhất. Gốc quyết định lựa chọn một đặc trưng của ví dụ huấn luyện, sau đó tuỳ thuộc vào giá trị của đặc trưng để gán cho nhãn giá trị 1 hay −1. Quá trình xác định nhãn phân loại được biểu diễn bởi công thức 2.2.

$$f_k(x) = a\delta(x^f > t) + b\delta(x^f \le t) \qquad (2.2)$$

Trong đó $\delta(e) = 1$ nếu e đúng và $\delta(e) = 0$ nếu ngược lại, t là một giá trị ngưỡng, a và b là tham số, x^f là giá trị đặc trưng thứ f của vectơ x. Trong trường hợp dữ liệu đánh giá chỉ bao gồm giá trị 1 và 0 hoặc 1 và −1, có thể chọn ngưỡng $t = 0$. Như vậy, ngoài việc phân loại, gốc quyết định còn thực hiện trích chọn đặc trưng do mỗi gốc chỉ chọn một đặc trưng duy nhất. Quá trình huấn luyện để chọn ra gốc tốt nhất được thực hiện bằng cách thử tất cả đặc trưng f để phép cực tiểu hoá (2.1). Với mỗi giá trị của f, giá trị tối ưu của a và b được tính theo kỹ thuật bình phương tối thiểu mà bản chất là tính giá trị tham số tại điểm có đạo hàm bằng 0.

$$a = \frac{\sum_i w_i y_i \delta(x^f > 0)}{\sum_i w_i \delta(x^f > 0)} \tag{2.3}$$

$$b = \frac{\sum_i w_i y_i \delta(x^f \le 0)}{\sum_i w_i \delta(x^f \le 0)} \tag{2.4}$$

Giá trị f, a và b tính được cho sai số dự đoán (2.1) nhỏ nhất sẽ được chọn để tạo ra bộ phân loại $f_k(x)$ cho vòng lặp thứ k. Bộ phân loại yếu $f_k(x)$ sau đó được thêm vào bộ phân loại chính $F(x)$ (bước b).

Tại bước (c), các ví dụ phân loại sai có $y_i f_k(x_i) < 0$ được GentleBoost tăng trọng số, các ví dụ phân loại đúng có $y_i f_k(x_i) > 0$ bị giảm trọng số. Với cách làm này, thuật toán sẽ khiến bộ phân loại ở vòng sau chú ý hơn tới những ví dụ hiện đang bị phân loại sai.

Mệnh đề 2.1. *Thuật toán GentleBoost cực tiểu hóa hàm lỗi khi phân loại thông qua các bước của phép khai triển Niutơn.*

2.3. PHÂN LOẠI VỚI CÁC ĐẶC TRƯNG CHUNG

2.3.1. Phương pháp học đa nhiệm

Phương pháp học máy thực hiện đồng thời cho nhiều nhiệm vụ liên quan để nâng cao kết quả dự đoán được gọi là phương pháp học đa nhiệm. Bằng việc suy diễn đồng thời giữa các nhiệm vụ, học đa nhiệm phát hiện ra được những tri thức từ nhiều nhiệm vụ để tăng cường vào kết quả dự đoán cho mỗi nhiệm vụ đơn lẻ. Với những bài toán có số lượng nhiệm vụ lớn nhưng có số ví dụ huấn luyện ít, học đa nhiệm nâng cao kết quả dự đoán cho mỗi nhiệm vụ bằng cách chia sẻ những thông tin chung giữa các nhiệm vụ.

Lọc cộng tác có thể được thực hiện theo phương pháp học đa nhiệm bằng kỹ thuật Boosting đã trình bày trong mục 2.2.2. Để thực hiện điều này, thuật toán GentleBoost được cải tiến bằng cách tại mỗi vòng lặp, thay vì giảm sai số cho một bài toán phân loại, thuật toán giảm sai số đồng thời cho một tập con các bài toán phân loại. Với cách làm này, tại mỗi vòng lặp thuật toán tìm được một *đặc trưng chung* cho tất cả các bài toán phân loại trong tập con các bài toán phân loại được chọn. Đặc trưng chung tìm được đóng vai trò chia sẻ, chuyển giao thông tin giữa các bài toán phân loại tăng cường thêm vào kết quả dự đoán.

2.3.2. Boosting đồng thời cho nhiều bài toán phân loại

Với tập N người dùng U; M sản phẩm U, và giá trị đánh giá r_{ij} như đã cho trong ở trên, ta có tất cả N bài toán phân loại, bài toán thứ n, $n = 1,\ldots,N$ được cho bởi M ví dụ huấn luyện ($x^n{}_1$,

y^n_1),…, (x^n_M, y^n_M), trong đó $y^n_j = r_{nj}$ là đánh giá của người dùng n cho sản phẩm j, và $x_{nj} = (r_{1j}, \ldots, r_{(n-1)j}, r_{(n+1)j}, \ldots, r_{Nj})$ là đánh giá của tất cả người dùng cho sản phẩm j trừ người dùng n. Cần lưu ý rằng, chỉ những cột có $r_{nj} \neq \varnothing$ mới được sử dụng làm ví dụ huấn luyện trong bài toán thứ k. Tuy nhiên, ta vẫn liệt kê cả những ví dụ có $r_{nj} = \varnothing$. Những ví dụ này sau đó sẽ được gán trọng số bằng 0 và do vậy không ảnh hưởng tới kết quả huấn luyện.

Mỗi ví dụ huấn luyện thứ j sẽ được làm tương ứng với n trọng số w^n_j, $n = 1,\ldots,N$. Mỗi trọng số được sử dụng khi ví dụ đó được dùng với bộ phân loại thứ n; $w^n_j = 0$ nếu $r_{nj} = 0$ tức là ví dụ j không tham gia vào huấn luyện bộ phân loại n. Sai số phân loại được tính bằng tổng sai số cho tất cả N bộ phân loại:

$$J = \sum_{n=1}^{N} \sum_{i=1}^{M} w_i^n (y_i^n - f_k^n(x_i))^2 \tag{2.13}$$

Tại mỗi vòng lặp n, gọi $S(t)$ là tập con các bài toán. Thay vì xác định đặc trưng f tốt nhất cho từng bài toán riêng lẻ như ở phần trước, thuật toán cần xác định đặc trưng chung cho tất cả bài toán thuộc $S(t)$ và chọn gốc quyết định tương ứng sao cho sai số (2.13) là nhỏ nhất. Gốc cây quyết định sẽ có dạng như sau:

$$f_k^n(x,t) = \begin{cases} a_S \delta(x^f > 0) + b_S \delta(x^f \leq 0) & \text{khi } n \in S(t) \\ c^n & \text{khi } n \notin S(t) \end{cases} \tag{2.14}$$

Ở đây, giá trị gốc cây quyết định phụ thuộc vào việc tập con $S(t)$ được chọn là tập con nào và vì vậy ta ký hiệu hàm f_k là hàm của t. Ký hiệu $f_k^n(x,t)$ được hiểu là hàm phân loại yếu tại bước thứ n cho bài toán thứ k và hàm này chung cho tập con $S(t)$ các bài toán phân loại. Do giá trị hàm lỗi (2.13) cũng phụ thuộc vào tập con $S(t)$ nên hàm lỗi (2.13) cũng cần viết lại thành hàm của tham số t như sau:

$$J(t) = \sum_{n=1}^{N} \sum_{i=1}^{M} w_i^n (y_i^n - f_k^n(x_i,t))^2 \tag{2.15}$$

Điểm khác nhau cơ bản so với gốc quyết định ở phần trước là gốc quyết định (2.15) phân biệt trường hợp bài toán n thuộc tập con $S(t)$ và trường hợp không thuộc. Trong trường hợp n không thuộc $S(t)$, hàm $f_k(x)$ sẽ được đặt bằng hằng số c^n để tránh trường hợp lựa chọn bộ phân loại một cách tình cờ do chênh lệch số lượng giữa ví dụ huấn luyện 1 và −1 (chẳng hạn trong trường hợp quá nhiều ví dụ 1 thì có thể luôn dự đoán nhãn là 1 không cần quan tâm tới đặc trưng).

Với mỗi tập con $S(t)$, giải bài toán cực tiểu hoá sai số (2.15) ta nhận được:

$$a_S(f) = \frac{\sum_{n \in S(t)} \sum_{i=1}^{M} w_i^n y_i^n \delta(x_i^f > 0)}{\sum_{n \in S(t)} \sum_{i=1}^{M} w_i^n \delta(x_i^f > 0)}, \tag{2.16}$$

$$b_S(f) = \frac{\sum_{n \in S(t)} \sum_{i=1}^{M} w_i^n y_i^n \delta(x_i^f \le 0)}{\sum_{n \in S(t)} \sum_{i=1}^{M} w_i^n \delta(x_i^f \le 0)}, \quad (2.17)$$

$$c^n = \frac{\sum_{i=1}^{M} w_i^n y_i^n}{\sum_{i=1}^{M} w_i^n}, \quad nk \notin S(t) \quad (2.18)$$

Tại mỗi bước lặp, thuật toán sẽ lựa chọn tập con $S(t)$ tốt nhất, tức là tập con cho giá trị hàm lỗi (2.15) nhỏ nhất và gốc quyết định tốt nhất cho tập con đó. Ký hiệu $F^n(x)$ là bộ phân loại chính xác cho bài toán phân loại thứ n, ta có thuật toán Boosting mới được thể hiện trên hình 2.4.

Đầu vào:

- Tập ví dụ huấn luyện của N bài toán phân loại, bài toán thứ n, n = 1,…, M được cho bởi M ví dụ huấn luyện (x^n_1, y^n_1),.., (x^n_M, y^n_M).

Đầu ra:

- Trả về bộ phân loại sign [$F^n(x)$]

Các bước thực hiện:

1. Khởi tạo $w^n_j = 1$ nếu $r_{nj} \neq \varnothing$ và $w^n_j = 0$ nếu $r_{nj} = \varnothing$, $i = 1,.., M$; n = 1,.., N

   Khởi tạo $F^n(x) = 0$
2. Lặp với $k = 1, \ldots, K$
   a. Lặp với tập con các bài toán $S(t)$
      i. Tính tham số a_S, b_S, và c^n theo (2.16), (2.17), (2.18)
      ii. Tính sai số $J(t) = \sum_{n=1}^{N} \sum_{i=1}^{M} w_i^n (y_i^n - f_k^n(x_i, t))^2$
   b. Chọn tập $S(t)$ tốt nhất $t^* = \arg\min_t J(t)$
   c. Cập nhật $F^n(x) \leftarrow F^n(x) + f_k^n(x_i, t^*)$
   d. Cập nhật trọng số $w_i^n \leftarrow w_i^n e^{-y_i^n f_k(x_i, t^*)}$
3. Trả về bộ phân loại sign [$F^n(x)$]

Hình 2.4. *Thuật toán MC-Boosting cải tiến sử dụng đặc trưng chung cho nhiều bài toán*

Mệnh đề 2.2. *Thuật toán MC-Boost cực tiểu hóa hàm lỗi khi phân loại thông qua các bước của phép khai triển Niutơn.*

Mệnh đề 2.3. *Số lượng các tập con S(t) cần duyệt của MC-Boost là $O(KN^2)$. Trong đó, K là số vòng lặp, N là số lượng người dùng.*

2.4. THỬ NGHIỆM VÀ KẾT QUẢ

2.4.1. Phương pháp thử nghiệm

Trước tiên, toàn bộ khách hàng được chia thành hai phần, một phần U_{tr} được sử dụng làm dữ liệu huấn luyện, phần còn lại U_{te} được sử dụng để kiểm tra. Dữ liệu huấn luyện được sử dụng để xây dựng mô hình theo thuật toán mô tả ở trên. Với mỗi khách hàng thuộc tập dữ liệu kiểm tra u, các đánh giá (đã có) của khách hàng được chia làm hai phần O_u và P_u. O_u được coi là đã biết, trong khi đó P_u là đánh giá cần dự đoán từ dữ liệu huấn luyện và O_u. Sai số dự đoán MAE_u với mỗi khách hàng u thuộc tập dữ liệu kiểm tra được tính bằng trung bình sai số tuyệt đối giữa giá trị dự đoán và giá trị thực đối với tất cả sản phẩm thuộc tập P_u.

2.4.2. Dữ liệu thử nghiệm

Thuật toán lọc cộng tác được thử nghiệm trên hai bộ dữ liệu EachMovie và MovieLens. Đây là hai bộ dữ liệu thường được sử dụng để đánh giá các phương pháp lọc cộng tác.

2.4.3. Kết quả thử nghiệm

Phương pháp Boosting với đặc trưng chung (ký hiệu là MC Boost) trình bày trong phần 4.2 được so sánh với những phương pháp sau: Phương pháp K hàng xóm gần nhất sử dụng độ tương quan Pearson (KPC). Phương pháp Boosting không sử dụng đặc trưng chung như trình bày trong Mục 2.2.2.

Trong trường hợp đủ dữ liệu, cụ thể là khi biết trước nhiều đánh giá của người dùng trong tập kiểm tra (N=20), phương pháp GentleBoost cho kết quả tốt hơn so với MC Boost. Có thể giải thích kết quả này là do GentBoost chọn được đặc trưng tối ưu hơn đối với từng bài toán phân loại, trong khi MC Boost chỉ chọn được đặc trưng tối ưu cho cả nhóm bài toán phân loại.

Tuy nhiên, khi dữ liệu ít đi, cụ thể là khi chỉ biết trước 5 hoặc 10 đánh giá của người dùng kiểm tra thì MC Boost cho sai số MAE nhỏ hơn so với GentleBoost trong đa số trường hợp. Lý do chủ yếu là do MC Boost cho phép kết hợp thông tin từ những người dùng tương tự với người dùng kiểm tra thông qua các đặc trưng chung và do vậy giảm được ảnh hưởng của việc thiếu nhãn phân loại.

Bảng 2.1. *Kết quả thử nghiệm với MovieLens*

Kích thước tập huấn luyện	Phương pháp	Số đánh giá cho trước của tập kiểm tra		
		5	10	20
100 người dùng	KPC	0.378	0.337	0.328
	GentleBoost	0.350	0.322	0.291
	MC Boost	0.329	0.305	0.292
200 người dùng	KPC	0.361	0.330	0.318
	GentleBoost	0.333	0.314	0.284
	MC Boost	0.314	0.299	0.289
300 người dùng	KPC	0.348	0.336	0.317
	GentleBoost	0.325	0.304	0.279
	MC Boost	0.308	0.298	0.283

Bảng 2.5. *Kết quả thử nghiệm với EachMovie*

Kích thước tập huấn luyện	Phương pháp	Số đánh giá cho trước của tập kiểm tra		
		5	10	20
1000 người dùng	KPC	0.559	0.474	0.449
	GentleBoost	0.515	**0.455**	**0.421**
	MC Boost	**0.492**	0.460	0.429
2000 người dùng	KPC	0.528	0.450	0.422
	GentleBoost	0.495	0.424	**0.393**
	MC Boost	**0.484**	**0.419**	**0.393**
6000 người dùng	KPC	0.521	0.437	0.378
	GentleBoost	0.477	0.408	**0.362**
	MC Boost	**0.452**	**0.397**	0.365

2.4.4. Phân tích kết quả

Để thấy rõ sự nổi trội của mô hình, chúng tôi lấy giá trị trung bình MAE của 10 lần kiểm nghiệm ngẫu nhiên trong tập dữ liệu kiểm tra để tiến hành một paired t-test. Giá trị nổi trội thống kê *p* (Statistical Significance) trong tất cả các bộ dữ liệu huấn luyện đều nhỏ hơn 0.05. Điều đó chứng tỏ, trên 5% giá trị MAE của phương pháp KPC lớn hơn GentleBoost và MC-Boost. Nói cách khác, GentleBoost và MC-Boost cho lại kết quả phân loại tốt hơn KPC.

2.5. KẾT LUẬN

Chương này đã trình bày một phương pháp học đa nhiệm cho lọc cộng tác. Phương pháp được phát triển dựa trên nền tảng của kỹ thuật phân loại Boosting kết hợp với trích chọn đặc trưng dựa vào gốc cây quyết định. Đây là một cải tiến của thuật toán Boosting, trong đó việc lựa chọn đặc trưng cho mỗi bộ phân loại yếu được thực hiện đồng thời trên một nhóm người dùng tương tự nhau.

Ưu điểm chủ yếu của phương pháp này là việc phân loại đồng thời từng nhóm người dùng và sử dụng thông tin từ những người dùng tương tự nhau, nhờ vậy cải thiện độ chính xác phân loại khi dữ liệu đánh giá thưa thớt(ví dụ khi người dùng cần dự đoán chỉ đánh giá rất ít sản phẩm trước đó). Kết quả thử nghiệm trên hai bộ dữ liệu MovieLens và EachMovie đã cho thấy phương pháp đề xuất cho kết quả tốt hơn những phương pháp khác trong trường hợp dữ liệu thưa.

CHƯƠNG 3
LỌC KẾT HỢP DỰA TRÊN MÔ HÌNH ĐỒ THỊ

3.1. VẤN ĐỀ LỌC KẾT HỢP

3.1.1. Bài toán lọc kết hợp

Giả sử hệ có N người dùng $U = \{u_1, u_2,\ldots, u_N\}$ và M sản phẩm $P = \{p_1, p_2,.., p_M\}$. Mỗi người dùng $u_i \in U$ đưa ra đánh giá của mình đối với sản phẩm $p_j \in P$ bằng một số r_{ij}. Mỗi đánh giá r_{ij} có thể nhận các giá trị trong một tập các giá trị rời rạc xác định. Để đơn giản, ta có thể coi r_{ij} nhận các giá trị *+1*, *-1, 0*.

$$r_{ij} = \begin{cases} +1 & \textit{Nếu người dùng } u_i \textit{ đã thích hợp phân bổ sản phẩm } p_j \\ -1 & \textit{Nếu người dùng } u_i \textit{ không thích hợp phân bổ sản phẩm } p_j \\ \phi & \textit{Nếu người dùng } u_i \textit{ chưa đánh giá sản phẩm } p_j \end{cases} \qquad (3.1)$$

Gọi $C = \{ c_1, c_2,..,c_K\}$ là K đặc trưng thể hiện nội dung các sản phẩm P. Ký hiệu ma trận $Y = (y_{ij})$ biểu thị mối quan hệ giữa sản phẩm và đặc trưng nội dung sản phẩm được xác định theo công thức (3.2).

$$y_{ij} = \begin{cases} 1 & \textit{nếu sản phẩm } p_i \textit{ có đặc trưng nội dung } c_j \\ 0 & \textit{nếu sản phẩm } p_i \textit{ không có đặc trưng nội dung } c_j \end{cases} \qquad (3.2)$$

Nhiệm vụ của lọc kết hợp là dự đoán cho người dùng hiện thời u_a những sản phẩm $p_k \in P$ chưa được u_a đánh giá dựa trên ma trận đánh giá r_{ij} và các đặc trưng nội dung $C = \{ c_1,c_2,..,c_K\}$.

3.2. LỌC CỘNG TÁC DỰA TRÊN MÔ HÌNH ĐỒ THỊ

3.2.1. Phương pháp biểu diễn đồ thị

Mô hình đồ thị cho lọc cộng tác có thể mô tả như sau. Cho ma trận đánh giá đầu vào của lọc cộng tác $R = (r_{ij})$ được xác định theo công thức (3.1). Gọi $X=(x_{ij})$ là ma trận cấp $N \times M$ có các phần tử được xác định theo công thức (3.3). Trong đó, $x_{ij} = 1$ tương ứng với trạng thái người dùng u_i đã đánh giá sản phẩm p_j, $x_{ij} = 0$ tương ứng với trạng thái người dùng chưa đánh giá sản phẩm p_j.

$$x_{ij} = \begin{cases} 1 & if \quad r_{ij} \neq \phi \\ 0 & otherwise \end{cases} \qquad (3.3)$$

Đồ thị biểu diễn đánh giá của người dùng đối với các sản phẩm (Gọi tắt là Người dùng - Sản phẩm) $G =(V, E)$ được biểu diễn theo ma trận X, trong đó tập đỉnh $V = U \cup P$ (U là tập người dùng, P là tập sản phẩm); tập cạnh E bao gồm tập các cạnh biểu diễn đánh giá của người dùng đối với sản phẩm. Cạnh nối giữa đỉnh $u_i \in U$ và đỉnh $p_j \in P$ được thiết lập nếu người dùng u_i đã đánh giá sản phẩm p_j ($x_{ij} = 1$). Trọng số của mỗi cạnh được lấy tương ứng là r_{ij}. Như vậy, trong biểu diễn này, đồ thị Người dùng- Sản phẩm có hai loại cạnh: Cạnh có trọng số dương r_{ij}=+1 biểu diễn

người dùng u_i "*thích*" sản phẩm p_j, cạnh có trọng số âm r_{ij}=-1 biểu diễn người dùng u_i "*không thích*" sản phẩm p_j.

3.2.2. dự đoán trên đồ thị Người dùng- Sản phẩm

Phương pháp dự đoán trên đồ thị Người dùng- Sản phẩm có thể được thực hiện thông qua các bước sau: Tách đồ thị Người dùng- Sản phẩm thành các đồ thị con được trình bày trong Mục 3.2.2.1, dự đoán trên đồ thị con chỉ bao gồm các cạnh có trọng số dương được trình bày trong Mục 3.2.2.2, dự đoán trên đồ thị con chỉ bao gồm các cạnh có trọng số âm được trình bày trong Mục 3.2.2.3, dự đoán trên tất cả đánh giá được trình bày trong Mục 3.2.2.4.

3.2.2.1. Tách đồ thị Người dùng- Sản phẩm thành các đồ thị con

Cho đồ thị Người dùng - Sản phẩm $G=(V, E)$ được biểu diễn theo ma trận $X = (x_{ij})$ cấp $N \times M$ như đã trình bày trong Mục 3.2.1. Ký hiệu $X^+ = \left(x_{ij}^+\right)$ là ma trận cấp $N \times M$ được xác định theo công thức (3.4). Ký hiệu $X^- = \left(x_{ij}^-\right)$ là ma trận cấp $N \times M$ được xác định theo công thức (3.5).

$$x_{ij}^+ = \begin{cases} 1 & if \quad r_{ij} > 0 \\ 0 & otherwise \end{cases} \tag{3.4}$$

$$x_{ij}^- = \begin{cases} 1 & if \quad r_{ij} < 0 \\ 0 & otherwise \end{cases} \tag{3.5}$$

Đồ thị $G^+ = (V, E^+)$ được biểu diễn theo ma trận X^+ có tập đỉnh đúng bằng tập đỉnh của G, có tập cạnh E^+ bao gồm các cạnh có trọng số dương của G.

$$E^+ = \left\{ e = (u_i, p_j) \in E \mid \quad r_{ij} = 1 \right\} \tag{3.6}$$

Đồ thị $G^- = (V, E^-)$ được biểu diễn theo ma trận X^- có tập đỉnh đúng bằng tập đỉnh của G, có tập cạnh E^- bao gồm các cạnh có trọng số âm của G.

$$E^- = \left\{ e = (u_i, p_j) \in E \mid \quad r_{ij} = -1 \right\} \tag{3.7}$$

3.2.2.2. Phương pháp dự đoán trên đồ thị G^+

Phương pháp dự đoán trên đồ thị G^+ được Huang đề xuất dựa trên việc tính toán trọng số các đường đi từ đỉnh người dùng đến đỉnh sản phẩm [113]. Những sản phẩm nào có trọng số cao nhất sẽ được dùng để tư vấn cho người dùng hiện thời.

Để ý rằng, đồ thị G, G^+, G^- đều là những đồ thị hai phía, một phía là các đỉnh người dùng, phía còn lại là các đỉnh sản phẩm. Do vậy, các đường đi từ đỉnh người dùng đến đỉnh sản phẩm luôn có độ dài lẻ.

Đối với đồ thị hai phía, số các đường đi độ dài L xuất phát từ một đỉnh bất kỳ thuộc phía người dùng đến đỉnh bất kỳ thuộc phía sản phẩm được xác định theo công thức (3.8), trong đó X là ma trận biểu diễn đồ thị hai phía, X^T là ma trận chuyển vị của X, L là độ dài đường đi.

$$X = \begin{cases} X & if \quad L = 1 \\ X.X^T X_\alpha^{L-2} & if \quad L = 3,5,7,... \end{cases} \tag{3.8}$$

Để ghi nhận trọng số của các đường đi từ đỉnh sản phẩm đến đỉnh người dùng trên đồ thị G^+ sao cho những đường đi dài có trọng số thấp, những đường đi ngắn có trọng số cao, ta sử dụng hằng khử nhiễu α ($0<\alpha\leq 1$) theo công thức (3.8), trong đó X^+ là ma trận biểu diễn đồ thị G^+, $(X^+)^T$

là ma trận chuyển vị của X^+, L là độ dài đường đi. Thuật toán dự đoán trên đồ thị G^+ được thể hiện trong Hình 3.1.

Đầu vào:

- *Ma trận X^+ là biểu diễn của đồ thị G^+*

Đầu ra:

- *K sản phẩm có trọng số cao nhất chưa được người dùng đánh giá*

Các bước thực hiện:

Bước 1. Tìm trọng số các đường đi độ dài lẻ L trên đồ thị G^+ sao cho các đường đi độ có dài nhỏ được đánh trọng số cao, các đường đi có độ dài lớn được đánh trọng số thấp.

$$(X^+)_\alpha^L = \begin{cases} \alpha.(X^+) & if \quad L = 1 \\ \alpha^2.(X^+).(X^+)^T.(X^+)_\alpha^{L-2} & if \quad L = 3,5,7... \end{cases}$$

Bước 2. Sắp xếp các sản phẩm theo thứ tự giảm dần của trọng số x_α^{+L}.

Bước 3. Chọn K sản phẩm có trọng số x_α^{+L} cao nhất chưa được đánh giá để tư vấn cho người dùng hiện thời.

Hình 3.1. Thuật toán dự đoán trên đồ thị G^+

Mệnh đề 3.1. Độ phức tạp thuật toán dự đoán trên đồ thị G^+ là $O(L.N^{2.376})$. Trong đó, L là độ dài đường đi từ đỉnh người dùng đến đỉnh sản phẩm, N là số cột của ma trận X^+.

3.2.2.3. Phương pháp dự đoán trên đồ thị G^-

Để xem xét ảnh hưởng các đánh giá "*không thích*" vào quá trình dự đoán, ta có thể ước lượng mức độ đóng góp của các đánh giá này trên đồ thị G^- bằng cách phủ định lại phương pháp dự đoán trên đồ thị G^+. Cụ thể phương pháp thay thế việc dự đoán trên đồ thị G^+ bằng đồ thị G^-. Thay việc ước lượng trọng số đường đi từ đỉnh người dùng đến đỉnh sản phẩm dài sẽ có trọng số thấp, đường đi ngắn có trọng số cao bằng việc ước lượng trọng số các đường đi dài có trọng số cao, đường đi ngắn có trọng số thấp. Thay việc sử dụng hằng số khử nhiễu $+\alpha$ bằng hằng số khử nhiễu $-\alpha$ để trọng số các đường đi luôn âm và tăng dần theo độ dài đường đi. Thay việc sắp xếp các sản phẩm theo thứ tự giảm dần của trọng số bằng việc sắp xếp các sản phẩm theo thứ tự tăng dần của trọng số. Thay quá trình phân bổ các sản phẩm có trọng số cao cho người dùng hiện thời bằng việc loại bỏ các sản phẩm có trọng số thấp. Thuật toán dự đoán trên đồ thị G^- được thể hiện trong Hình 3.2.

Đầu vào:

- *Ma trận X^- là biểu diễn của đồ thị G^-*

Đầu ra:

- *K sản phẩm có trọng số nhỏ nhất chưa được người dùng đánh giá*

Các bước thực hiện:

Bước 1. Tìm trọng số các đường đi độ dài lẻ *L* trên đồ thị G^- sao cho các đường đi có độ dài nhỏ được đánh trọng số thấp, các đường đi có độ dài lớn được đánh trọng số cao.

$$(X^-)_\alpha^L = \begin{cases} -\alpha.(X^-) & if \quad L = 1 \\ (-\alpha)^2.(X^-).(X^-)^T.(X^-)_\alpha^{L-2} & if \quad L = 3,5,7... \end{cases}$$

Bước 2. Sắp xếp các sản phẩm theo thứ tự tăng dần của trọng số x_α^{-L}.

Bước 3. Loại bỏ *K* sản phẩm có trọng số x_α^{-L} thấp nhất chưa được người dùng đánh giá ra khỏi danh sách các sản phẩm cần tư vấn cho người dùng hiện thời.

***Hình 3.2**. Thuật toán dự đoán trên đồ thị G^-*

Mệnh đề 3.2. Độ phức tạp thuật toán dự đoán trên đồ thị G^- là $O(L.N^{2.376})$. Trong đó, *L* là độ dài đường đi từ đỉnh người dùng đến đỉnh sản phẩm, *N* là số cột của ma trận X^-.

3.2.2.4. Phương pháp dự đoán theo tất cả đánh giá

Phương pháp dự đoán trên đồ thị G^+ chỉ được thực hiện trên những đánh giá "*thích*" của người dùng đối với sản phẩm, phương pháp dự đoán trên đồ thị G^- chỉ được thực hiện trên những đánh giá "*không thích*" của người dùng đối với sản phẩm. Việc bỏ qua những đánh giá "*không thích*" của người dùng đối với sản phẩm có những ảnh hưởng không nhỏ đến chất lượng dự đoán, vì đánh giá "*thích*" hay "*không thích*" đều phản ánh thói quen và sở thích sử dụng sản phẩm của người dùng. Để khắc phục mâu thuẫn này, ta có thể mở rộng phương pháp dự đoán cho tất cả các đánh giá "*thích*" và "*không thích*" của người dùng. Các bước cụ thể của phương pháp được tiến hành như Hình 3.3.

Đầu vào:

- *Ma trận X^+, X^- là biểu diễn của đồ thị G^+, G^-*

Đầu ra:

- *K sản phẩm có trọng số cao nhất chưa được người dùng đánh giá*

Các bước thực hiện:

Bước 1. Tính toán ma trận trọng số $(X^+)^L_\alpha$ của các đường đi độ dài lẻ *L* trên ma trận X^+ sao cho các đường đi có độ dài nhỏ được đánh trọng số cao, các đường đi có độ dài lớn được đánh trọng số thấp.

$$(X^+)^L_\alpha = \begin{cases} \alpha.(X^+) & if \quad L=1 \\ \alpha^2.(X^+).(X^+)^T.(X^+)^{L-2}_\alpha & if \quad L=3,5,7... \end{cases}$$

Bước 2. Tính toán ma trận trọng số $(X^-)^L_\alpha$ của các đường đi độ dài lẻ *L* trên ma trận X^- sao cho các đường đi có độ dài nhỏ được đánh trọng số thấp, các đường đi có độ dài lớn được đánh trọng số cao.

$$(X^-)^L_\alpha = \begin{cases} -\alpha.(X^-) & if \quad L=1 \\ (-\alpha)^2.(X^-).(X^-)^T.(X^-)^{L-2}_\alpha & if \quad L=3,5,7... \end{cases}$$

Bước 3. Kết hợp ma trận trọng số $X^L_\alpha = (X^+)^L_\alpha + (X^-)^L_\alpha$.

Bước 4. Sắp xếp các sản phẩm theo thứ tự tăng dần của trọng số x^L_α.

Bước 5. Chọn *K* sản phẩm có trọng số x^L_α cao nhất chưa được đánh giá để tư vấn cho người dùng hiện thời.

Hình 3.3. *Thuật toán dự đoán trên tất cả đánh giá*

Mệnh đề 3.3. Độ phức tạp thuật toán dự đoán trên tất cả đánh giá là O($L.N^{2.376}$). Trong đó, *L* là độ dài đường đi từ đỉnh người dùng đến đỉnh sản phẩm, *N* là số cột của ma trận X^+, X^-.

3.3. KẾT HỢP LỌC CỘNG TÁC VÀ LỌC NỘI DUNG

Mục này trình bày mô hình đồ thị kết hợp giữa lọc cộng tác và lọc nội dung. Đối với lọc cộng tác, mô hình quan tâm xem xét và biểu diễn cho tất cả các đánh giá "*thích hợp*" hoặc "*không thích hợp*" như đã trình bày trong Mục 3.2. Đối với các đặc trưng nội dung, mô hình đề xuất phương pháp xác định mức độ quan trọng của từng đặc trưng nội dung cụ thể đối với mỗi người dùng dựa trên ước lượng sự tương tự theo nội dung và đánh giá người dùng. Phương pháp dự đoán được thực hiện dựa trên mức độ đóng góp của các đánh giá người dùng và đặc trưng nội dung sản phẩm người dùng ưa thích.

3.3.1. Biểu diễn đồ thị kết hợp

Cho ma trận đánh giá người dùng $R = (r_{ij})$ được xác định theo công thức (3.1), ma trận nội dung sản phẩm $Y = (y_{ij})$ được xác định theo công thức (3.2), ma trận $X = (x_{ij})$ được xác định theo công thức 3.6. Khi đó, đồ thị kết hợp $G = (V, E)$ được hình thành bởi tập đỉnh $V = U \cup P \cup C$ (*U* là tập người dùng, *P* là tập sản phẩm, *C* là tập đặc trưng nội dung sản phẩm); Cạnh nối giữa đỉnh

$u_i \in U$ và đỉnh $p_j \in P$ được thiết lập nếu $x_{ij} \neq 0$ và được đánh trọng số là $r_{ij} = +1$ hoặc $r_{ij} = -1$, cạnh nối giữa $p_i \in P$ và $c_j \in C$ được thiết lập nếu $y_{ij} \neq 0$; có trọng số bằng nhau là +1. Ví dụ với ma trận đánh giá *R* được cho trong Bảng 3.1, ma trận nội dung trong bảng 3.2 thì ma trận *X* được thể hiện trong Bảng 3.3 và đồ thị kết hợp được biểu diễn như Hình 3.4.

Bảng 3.1. *Ma trận đánh giá R*

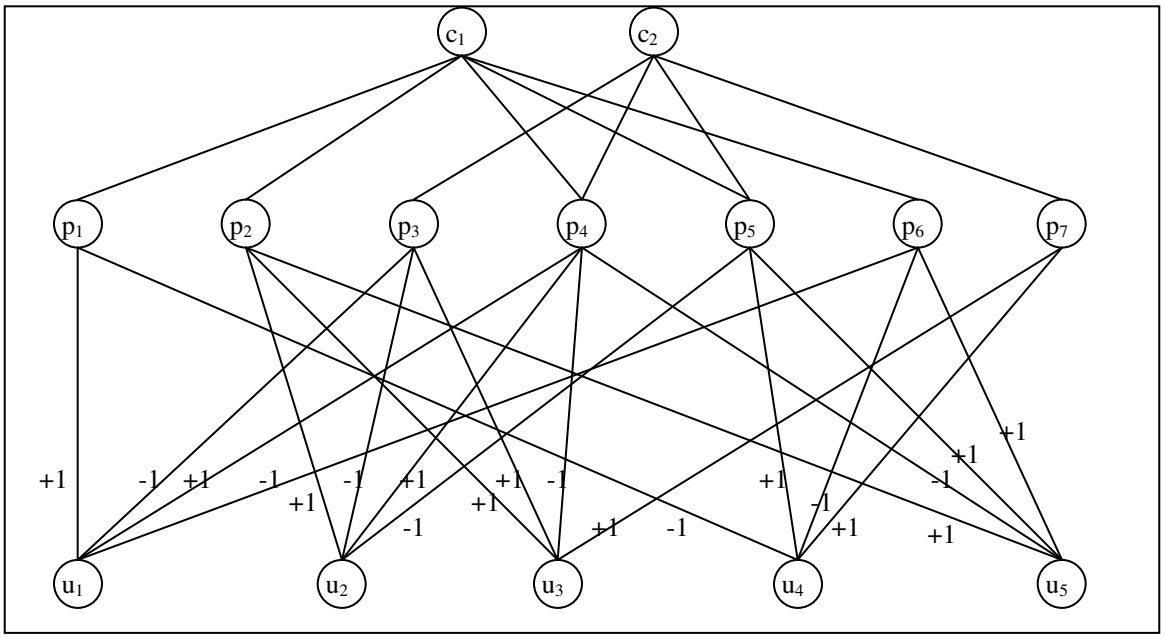
Người dùng	*Sản phẩm*						
	p_1	p_2	p_3	p_4	p_5	p_6	p_7
u_1	1	∅	-1	1	∅	-1	∅
u_2	∅	1	-1	1	-1	∅	∅
u_3	∅	1	1	-1	∅	∅	1
u_4	-1	∅	∅	∅	1	-1	1
u_5	∅	1	∅	-1	1	1	∅

Bảng 3.2. *Ma trận Sản phẩm-Nội dung*

Sản phẩm	*Nội dung*	
	c_1	c_2
p_1	1	0
p_2	1	0
p_3	0	1
p_4	1	1
p_5	1	1
p_6	1	0
p_7	0	1

Bảng 3.3. *Ma trận người dùng sản phẩm X*

Người dùng	*Sản phẩm*						
	p_1	p_2	p_3	p_4	p_5	p_6	p_7
u_1	1	0	1	1	0	1	0
u_2	0	1	1	1	1	0	0
u_3	0	1	1	1	0	0	1
u_4	1	0	0	0	1	1	1
u_5	0	1	0	1	1	1	0



Hình 3.4. *Đồ thị thiết lập liên kết giữa người dùng và nội dung sản phẩm*

3.3.2. Xây dựng liên kết người dùng và nội dung sản phẩm

Gọi s_{ik} là số các sản phẩm p_j có nội dung c_k mà người dùng u_i đã đánh giá. Giá trị s_{ik} chính là số đường đi độ dài 2 từ đỉnh người dùng u_i đến đỉnh đặc trưng nội dung c_k thông qua các đỉnh trung gian p_j.

$$s_{ik} = \sum_{j=1}^{M} x_{ij} * y_{jk} \tag{3.9}$$

Gọi w_{ik} là hiệu số giữa tập các sản phẩm p_j có nội dung c_k người dùng u_i đánh giá "*thích hợp*" và tập các sản phẩm p_j có nội dung c_k người dùng u_i đánh giá "*không thích hợp*".

$$w_{ik} = \sum_{j=1}^{M} r_{ij} * y_{jk} \tag{3.10}$$

Khi đó, mức độ quan trọng của đặc trưng nội dung c_k đối với người dùng u_i được xác định theo công thức (3.11).

$$v_{ik} = \begin{cases} \dfrac{\min(s_{ik}, \gamma)}{\gamma} * \dfrac{w_{ik}}{s_{ik}} & if \quad \dfrac{w_{ik}}{s_{ik}} > T \\ 0 & otherwise \end{cases} \tag{3.11}$$

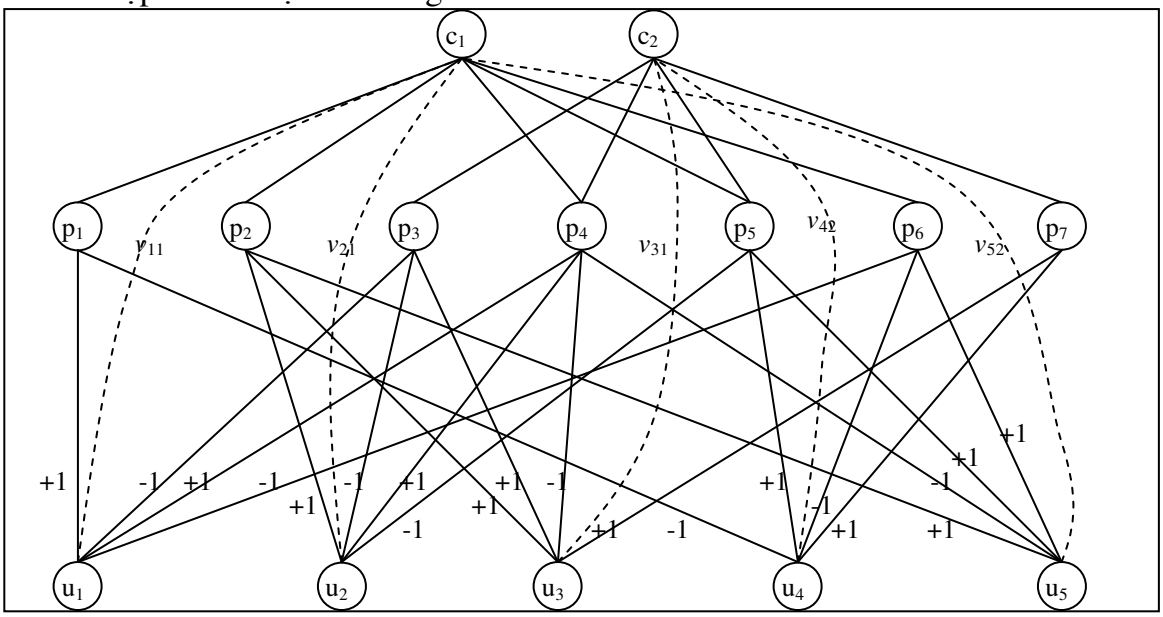
Trong công thức (3.11), nếu $s_{ik} > \gamma$ thì $\dfrac{\min(s_{ik}, \gamma)}{\gamma} = 1$, khi đó v_{ik} được xác định theo công thức (3.12). Nếu $s_{ik} \le \gamma$ thì $\dfrac{\min(s_{ik}, \gamma)}{\gamma} = \dfrac{s_{ik}}{\gamma}$, khi đó v_{ik} được xác định theo công thức (3.13).

$$v_{ik} = \begin{cases} \dfrac{w_{ik}}{s_{ik}} & if \quad \dfrac{w_{ik}}{s_{ik}} > T \\ 0 & otherwise \end{cases} \tag{3.12}$$

$$v_{ik} = \begin{cases} \dfrac{w_{ik}}{\gamma} & if \quad \dfrac{w_{ik}}{s_{ik}} > T \\ 0 & otherwise \end{cases} \tag{3.13}$$

Trong thử nghiệm, chúng tôi dùng ngưỡng $\gamma = 20$, nghĩa là nếu người dùng u_i đánh giá các sản phẩm p_j có nội dung c_k lớn hơn 20 thì v_{ik} được xác định theo 3.10, trường hợp còn lại v_{ij} được tính theo 3.11. Giá trị ngưỡng T được chọn là 0.3, nghĩa là số cạnh có trọng số dương gấp đôi số cạnh có trọng số âm đối với mỗi đặc trưng được xem là quan trọng.

Với mỗi cặp đỉnh (u_i, c_k) có $v_{ik} > 0$, chúng ta thiết lập một liên kết trực tiếp giữa người dùng u_i và đặc trưng c_k với trọng số v_{ik}. Ví dụ với các ma trận *R, Y, X* được cho trong Bảng 3.1, 3.2, 3.3 và đồ thị biểu diễn trong Hình 3.4 thì đồ thị kết hợp được xác định theo Hình 3.5, trong đó các cạnh mới thiết lập thêm được nối bằng các nét đứt.



Hình 3.5. *Đồ thị thiết lập liên kết giữa người dùng và nội dung sản phẩm*

3.3.3. Phương pháp dự đoán

Các phương pháp lọc cộng tác thuần túy, lọc nội dung thuần túy, lọc kết hợp đơn giản, lọc kết hợp dựa vào ước lượng mức độ quan trọng của các đặc trưng nội dung (Combinate-Graph) có thể xem như một bài toán tìm kiếm trên đồ thị kết hợp.

3.3.3.1. Lọc cộng tác dựa trên mô hình đồ thị kết hợp

Phương pháp lọc cộng tác có thể dễ dàng cài đặt bằng mô hình đồ thị thông qua việc tính toán các đuờng đi độ dài 3 từ đỉnh người dùng đến đỉnh sản phẩm thông qua các cạnh đánh giá. Những sản phẩm nào có số đường đi độ dài 3 nhiều nhất đến nó sẽ được phân bổ cho người dùng hiện thời.

3.3.3.2. Lọc nội dung dựa trên mô hình đồ thị kết hợp

Phương pháp dự đoán theo nội dung nguyên thủy cũng dễ dàng cài đặt dựa trên mô hình đồ thị bằng cách xem xét tất cả các đường đi thông qua đỉnh đặc trưng nội dung. Những sản phẩm nào có nhiều đường đi nhất thông qua đỉnh đặc trưng nội dung sẽ được phân bổ cho người dùng hiện thời.

3.3.3.3. Phương pháp lọc kết hợp đơn giản

Phương pháp lọc kết hợp đơn giản (Ký hiệu là SimpleHybrid) được thực hiện bằng cách kết hợp phương pháp lọc cộng tác như đã trình bày trong Mục 3.3.3.1 và lọc nội dung trong Mục 3.3.3.2. Những sản phẩm nào có số đường đi nhiều nhất đến nó sẽ được dùng để phân bổ cho người dùng hiện thời.

3.3.3.4. Phương pháp kết hợp đề xuất

Như đã trình bày ở trên, phương pháp dự đoán đề xuất dựa trên việc ước lượng mức độ quan trọng các đặc trưng nội dung cho mỗi người dùng. Để thực hiện điều này trên đồ thị kết hợp, ta xem xét và thực hiện tính toán mức độ đóng góp vào kết quả dự đoán cho hai loại đường đi: *đường đi thông qua đỉnh nội dung(đường đi loại 1)* và *đường đi thông qua đỉnh sản phẩm (đường đi loại 2)*.

Đường đi loại 1 luôn có độ dài 2 đi từ đỉnh người dùng $u_i \in U$ thông qua các cạnh nối đỉnh nội dung $c_k \in C$ đến đỉnh sản phẩm $p_j \in P$. Những đường đi này phản ánh sự tương tự của người dùng sản phẩm đối với các đặc trưng nội dung. Những đường đi loại này được tính toán dựa trên đồ thị kết hợp sau khi loại bỏ đi các cạnh đánh giá.

Đường đi loại 2 bao gồm các đường đi từ đỉnh người dùng đến đỉnh sản phẩm chưa được người dùng đánh giá thông qua các đỉnh sản phẩm và đỉnh người dùng trung gian. Độ dài những đường đi này luôn lẻ và không vượt quá L. Những đường đi độ dài lẻ có thể thông qua các cạnh có trọng số âm hoặc các cạnh có trọng số dương đều được xem xét đến trong quá trình dự đoán. Các đường đi loại này bao gồm:

- *Tất cả các đường đi từ đỉnh người dùng đến đỉnh sản phẩm thông qua các cạnh trung gian đều có trọng số dương*. Trọng số các đường đi này được tính toán trên đồ thị có biểu diễn dương như đã trình bày trong Mục 3.3.1.
- *Tất cả các đường đi từ đỉnh người dùng đến đỉnh sản phẩm thông qua các cạnh trung gian đều có trọng số âm*. Trọng số các đường đi này được tính toán trên đồ thị có biểu diễn âm như đã trình bày trong Mục 3.3.2.
- *Những đường đi qua hai đỉnh trung gian và kết thúc tại cùng một đỉnh nhưng trái dấu*, điều đó có nghĩa cả hai người dùng có đánh khác nhau về sản phẩm này. Đối với những đường đi này, chúng ta không cần xem xét đến vì hai người dùng không tương đồng với nhau về sở thích.
- *Những đường thông qua hai đỉnh liên tục nhau đều có trọng số âm*. Điều này có nghĩa hai người dùng đều tương tự với p_6 (đều là không thích hợp). Tuy nhiên, trong thử nghiệm các đường đi loại này cho lại kết quả dự đoán không cao. Do vậy, ta không cần xem xét đến những đường đi này.

Để xác định mức độ đóng góp của mỗi loại đường đi vào kết quả dự đoán, ta sử dụng tham số λ *(0* $\leq \lambda \leq$ *1)* điều chỉnh mức độ ưu tiên cho từng loại. Gọi Y_r^i là số đường đi loại 1 có độ dài 2 từ đỉnh đầu u_i đến đỉnh cuối p_r thông các đỉnh nội dung có dạng u_i-c_j-p_r, X_r^i là trọng số đường đi độ dài lẻ không nhỏ hơn L từ đỉnh đầu u_i đến đỉnh cuối p_r thông qua các đỉnh sản phẩm có dạng u_i-p_j-u_k-p_r. Khi đó, khả năng tư vấn p_r cho người dùng u_i là W_r^i được xác định theo công thức (3.14).

$$W_r^i = \lambda . X_r^i + (1-\lambda) Y_r^i \qquad (3.14)$$

Trong công thức 3.12, nếu ta ưu tiên cho lọc cộng tác thì λ được lấy gần với 0; Nếu ưu tiên cho lọc nội dung thì thì λ được lấy gần với 1; Nếu λ =0 thì phương pháp dự đoán trở lại đúng mô hình lọc cộng tác dựa trên tất cả đánh giá; Nếu λ =1 thì phương pháp dự đoán hoàn toàn dựa trên nội dung. Nếu lấy λ = 0.5 thì mức độ ưu tiên cho lọc cộng tác và lọc nội dung là như nhau. Thuật toán dự đoán trên đồ thị kết hợp được thể hiện trong Hình 3.6.

Đầu vào:

- Ma trận biểu diễn các cạnh Người dùng - Nội dung.
- Ma trận X^+, X^- biểu diễn đồ thị G^+, G^- .

Đầu ra:

- K sản phẩm có trọng số cao nhất chưa được người dùng đánh giá

Các bước thực hiện:

Bước 1: *Xác định trọng số các đường đi loại 1 là* Y_1^2.

Bước 2: *Xác định trọng số các đường đi loại 2 là* X_α^L*:*

a. Tìm $(X^+)_\alpha^L$ là trọng số các đường đi trên đồ thị G^+ theo thuật toán được trình bày trong Mục 3.2.2.2.

b. Tìm $(X^-)_\alpha^L$ là trọng số các đường đi trên đồ thị G^- theo thuật toán được trình bày trong Mục 3.2.2.3.

c. Kết hợp trọng số $(X^+)_\alpha^L$ và $(X^-)_\alpha^L$:

$$X_\alpha^L = (X^+)_\alpha^L + (X^-)_\alpha^L.$$

Bước 3: *Hợp nhất trọng số của hai loại đường đi theo công thức (3.14) ta nhận được:* $T = \lambda.X_\alpha^L + (1-\lambda)Y_1^2$

Bước 4: *Sắp xếp các sản phẩm theo thứ tự giảm dần của trọng số.*

Bước 5. *Chọn K sản phẩm có trọng số cao nhất tư vấn cho người dùng hiện thời.*

Hình 3.6. *Thuật toán dự đoán Combined-Graph trên đồ thị kết hợp.*

Mệnh đề 3.4. Độ phức tạp thuật toán trên đồ thị kết hợp là $O(L.|U|^{2.376} + (|U|+|P|+|C|)^{2.376})$. Trong đó, $|U|$ là số lượng người dùng, $|P|$ là số lượng sản phẩm, $|C|$ là số lượng các đặc trưng nội dung.

3.4. THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.4.1. Dữ liệu thử nghiệm

Mô hình đề xuất được tiến hành thử nghiệm trên hai tập dữ liệu của bộ dữ liệu MovieLens. Tập dữ liệu MovieLens thứ nhất (MovieLens1) gồm 1682 người dùng, 942 phim với trên 100000 đánh giá. Tập dữ liệu MovieLens thứ hai (MovieLens2) gồm 6040 người dùng, 3900 phim với trên 1000000 đánh giá như đã được mô tả trong Chương 2 (www.grouplens.org/node/12).

Chọn ngẫu nhiên trong tập MovieLan1 500 người dùng làm dữ liệu huấn luyện, chọn ngẫu nhiên trong số còn lại 150 người dùng làm dữ liệu kiểm tra. Chọn ngẫu nhiên trong tập MovieLan2 1000 người dùng làm dữ liệu huấn luyện, chọn ngẫu nhiên trong số còn lại 320 người dùng làm dữ liệu kiểm tra. Hai mức đánh giá cao nhất (4, 5) được biến đổi thành "*thích*" (+1), các mức còn lại biến đổi thành "*không thích*" (-1). Các đặc trưng nội dung của phim được chọn là thể loại và đạo diễn. Các tập dữ liệu này cũng được Grouplens cung cấp kèm theo các tập dữ liệu tương ứng.

3.4.2. Phương pháp thử nghiệm

Phương pháp đánh giá sai số phân loại dựa trên độ chính xác *P* (*Precision*) và độ nhạy *R* (*Recall*). Trước tiên toàn bộ sản phẩm trong tập dữ liệu kiểm tra được chia thành hai lớp: Lớp các sản phẩm phân bổ thích hợp và lớp các sản phẩm phân bổ không thích hợp. Gọi *N* là tổng số các đánh giá người dùng trong tập dữ liệu kiểm tra, trong đó N_r là số các sản phẩm người dùng đã đánh giá thích hợp, N_{rs} là số các sản phẩm phương pháp lọc dự đoán chính xác, khi đó độ chính xác *P* được tính theo công thức (3.15), độ nhạy *R* được tính toán theo công thức (3.16), và độ đo *F* (*F-Measure*) được tính theo công thức (3.17). Giá trị *P, R, F_Measure* càng lớn độ, chính xác của phương pháp càng cao.

$$P = \frac{N_{rs}}{N_r} \tag{3.15}$$

$$R = \frac{N_{rs}}{N} \tag{3.16}$$

$$F\_Measure = \frac{2 \times P \times R}{(P + R)} \tag{3.17}$$

3.4.3. So sánh và đánh giá dựa vào Precision, Recall và F-measure

Mô hình lọc cộng tác kết hợp với lọc nội dung dựa trên đồ thị (ký hiệu là CombinedGraph). Độ chính xác, độ nhạy và *F-Measure* được tính toán dựa trên danh sách đầu tiên của 10, 20 và 50 sản phẩm dùng để tư vấn. Các giá trị ngưỡng lần lượt được chọn là: $\gamma = 20$ và $\alpha = 0.5$, $\lambda = 0.8$.

Bảng 3.4. Giá trị Precision, Recall, F-Measure kiểm nghiệm trên tập MovieLens1

Phương pháp	Độ đo	Số sản phẩm dùng để tư vấn		
		10	20	50
UserBased	Độ nhạy	0.001	0.031	0.078
	Độ chính xác	0.003	0.041	0.054
	F-Measure	0.123	0.028	0.054
ContentBased	Độ nhạy	0.018	0.026	0.046
	Độ chính xác	0.038	0.032	0.026
	F-Measure	0.020	0.024	0.028
3Hop	Độ nhạy	0.138	0.207	0.361
	Độ chính xác	0.331	0.286	0.214
	F-Measure	0.152	0.190	0.222
SimpleHybrid	Độ nhạy	0.098	0.144	0.259
	Độ chính xác	0.211	0.174	0.144
	F-Measure	0.105	0.123	0.152
CombinedGraph	Độ nhạy	0.142	0.215	0.366
	Độ chính xác	0.339	0.291	0.215
	F-Measure	0.157	0.195	0.224

Bảng 3.5. Giá trị Precision, Recall, F-Measure kiểm nghiệm trên tập MovieLens2

Phương pháp	Độ đo	Số sản phẩm dùng để tư vấn		
		10	20	50
UserBased	Độ nhạy	0.007	0.021	0.069
	Độ chính xác	0.015	0.025	0.034
	F-Measure	0.009	0.023	0.045
ContentBased	Độ nhạy	0.009	0.017	0.037
	Độ chính xác	0.022	0.020	0.018
	F-Measure	0.013	0.018	0.024
3Hop	Độ nhạy	0.155	0.222	0.377
	Độ chính xác	0.284	0.225	0.164
	F-Measure	0.200	0.223	0.228
SimpleHybrid	Độ nhạy	0.117	0.162	0.279
	Độ chính xác	0.186	0.148	0.118
	F-Measure	0.144	0.155	0.166
CombinedGraph	Độ nhạy	0.165	0.234	0.381
	Độ chính xác	0.292	0.240	0.175
	F-Measure	0.211	0.237	0.240

Kết quả kiểm nghiệm của mô hình đề xuất được lấy trung bình từ 10 lần kiểm nghiệm ngẫu nhiên cùng với kết quả của của các phương pháp: Phương pháp lọc cộng tác dựa trên người dùng sử dụng thuật toán *KNN* và độ tương quan Pearson (ký hiệu là UserBased) [47], phương pháp lọc cộng tác trên đồ thị G^+ (Ký hiệu là 3Hop), phương pháp lọc theo nội dung (ký hiệu là ContentBased) dựa trên mô hình đồ thị, phương pháp lọc kết hợp đơn giản (Ký hiệu là SimpleHybrid).

Kết quả kiểm nghiệm cho thấy mô hình đề xuất cho lại kết quả độ chính xác, độ nhạy và F-Measure đều lớn hơn so với các phương pháp còn lại. Điều đó chứng tỏ việc xác định mức độ ưa thích của người dùng đối với những đặc trưng nội dung sản phẩm có ý nghĩa đặc biệt quan trọng để nâng cao chất lượng dự đoán cho các hệ thống tư vấn.

3.4.4. Phân tích kết quả

Để thấy rõ sự nổi trội của mô hình, chúng tôi lấy giá trị trung bình F-Measure sau 10 lần kiểm nghiệm ngẫu nhiên của 150 người dùng trong tập dữ liệu kiểm tra của MovieLens1 và 320 người dùng trong tập dữ liệu kiểm tra của MovieLens2 để tiến hành một paired t-test. Tham số so sánh mức độ nổi trội thống kê giữa CombinedGraph và các phương pháp còn lại đều cho giá trị $p<0.05$ chứng tỏ phương pháp CombinedGraph cho lại giá trị F-Measure lớn hơn phương pháp so sánh ít nhất 5% trên tổng số lần lần quan sát.

3.5. KẾT LUẬN

Chương này trình bày một mô hình trực quan và hiệu quả cho lọc kết hợp. Mô hình cho phép biểu diễn tất cả các đánh giá người dùng trên đồ thị kết hợp. Trên cơ sở đó, xây dựng mô hình kết hợp giữa lọc cộng tác và lọc nội dung bằng cách xác định mức độ quan trọng của mỗi đặc trưng nội dung đối với từng người dùng riêng biệt để thực hiện dự đoán. Mô hình được biểu diễn bằng đồ thị các mối quan hệ giữa người dùng, sản phẩm và nội dung không chỉ kế thừa được các thuật toán tìm kiếm hiệu quả trên đồ thị mà còn dễ dàng mở rộng biểu diễn cho các lớp thông

tin khác nhau. Kết quả kiểm nghiệm trên bộ dữ liệu MovieLens cho thấy, mô hình cho lại kết quả tốt hơn các phương pháp lọc cộng tác dựa trên độ tương quan và lọc theo nội dung thuần túy. Đặc biệt, mô hình thực hiện hiệu quả trong trường hợp có ít dữ liệu đánh giá.

KẾT LUẬN

Lọc cộng tác và lọc nội dung là hai phương pháp tiếp cận chính được áp dụng cho các hệ thống lọc thông tin. Lọc nội dung thực hiện tốt trên các đối tượng dữ liệu văn bản nhưng lại khó thực hiện trên các dạng thông tin đa phương tiện. Lọc cộng tác có thể lọc được mọi loại thông tin nhưng gặp phải vấn đề dữ liệu thưa, một người dùng mới chưa có đánh giá nào về sản phẩm, một sản phẩm mới chưa được người dùng nào đánh giá. Kết hợp cả hai phương pháp lọc cơ bản trên sẽ góp phần nâng cao chất lượng dự đoán cho các hệ thống tư vấn.

Luận án đã giải quyết được hai vấn đề còn tồn tại trong lọc thông tin, đó là vấn đề dữ liệu thưa của lọc cộng tác và vấn đề kết hợp hiệu quả giữa lọc cộng tác và lọc nội dung.

Để hạn chế ảnh hưởng của vấn đề dữ liệu thưa, luận án đề xuất sử dụng phương pháp học đa nhiệm vào lọc cộng tác nhằm sử dụng tập đặc trưng chung của tập người dùng khác nhau vào quá trình huấn luyện. Những đặc trưng chung tìm được đóng vai trò chia sẻ thông tin trong tập người dùng tương ứng không chỉ nâng cao được kết quả dự đoán mà còn hạn chế được ảnh hưởng của vấn đề dữ liệu thưa.

Để giải quyết vấn đề kết hợp giữa lọc cộng tác và lọc nội dung, các tác giả thường cài đặt hai cơ chế lọc cộng tác và lọc nội dung độc lập nhau sau đó tổng hợp kết quả dự đoán hai phương pháp cho toàn bộ mô hình. Rõ ràng, giữa hai cách tiếp cận lọc cộng tác và lọc nội dung dựa trên quan điểm khác nhau để tìm ra những sản phẩm tương tự đối với người dùng. Hai sản phẩm tương tự nhau về nội dung không thể suy ra được hai người dùng tương tự nhau về sở thích, ngược lại hai người dùng tương tự nhau về đánh giá không suy ra được họ sử dụng các sản phẩm giống nhau về nội dung. Để giải quyết mâu thuẫn này, ta có thể dựa vào việc quan sát tất cả đánh giá người dùng đối với mỗi đặc trưng nội dung cụ thể từ đó tìm ra mức độ quan trọng của các đặc trưng nội dung cho người dùng. Kết quả kiểm nghiệm trên bộ dữ liệu thực về phim cho thấy, cách tiếp cận này cho lại kết quả tốt ngay cả trong trường hợp dữ liệu đánh giá thưa thớt.

Phương pháp lọc kết hợp đề xuất được sử dụng để xây dựng hệ tư vấn lựa chọn phim. Hệ thống phản ánh đầy đủ các chức năng cơ bản của một hệ thống lọc thông tin, bao gồm thành phần phân tích thông tin, thành phần người dùng, thành phần học và thành phần lọc. Hệ thống cho lại kết quả tư vấn tốt trên bộ dữ liệu MovieLens gồm 39000 phim và 6040 người dùng.

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

[1] Nguyen Duy Phuong, Le Quang Thang, Tu Minh Phuong (2008), "A Graph-Based for Combining Collaborative and Content-Based Filtering", *PRICAI 2008*: 859-869.

[2] Nguyen Duy Phuong, Tu Minh Phuong (2008), "Collaborative Filtering by Multi-Task Learning", *RIVF 2008*: 227-232.

[3] Nguyễn Duy Phương, Từ Minh Phương (2009), "Lọc cộng tác và lọc theo nội dung dựa trên mô hình đồ thị", *Chuyên san các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông*, Tập V-1 số 1, trang: 4-12.

[4] Nguyễn Duy Phương, Từ Minh Phương (2008), "Một thuật toán lọc cộng tác cho trường hợp ít dữ liệu", *Tạp chí Tin học và Điều khiển học*, tập 24, trang: 62-74.

[5] Nguyễn Duy Phương, Phạm Văn Cường, Từ Minh Phương (2008), "Một số giải pháp lọc thư rác tiếng Việt", *Chuyên san các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông*, số 19, trang: 102-112.