

# PHƯƠNG PHÁP XÂY DỰNG HỆ THỐNG GỢI Ý SẢN PHẨM SỬ DỤNG PHẢN HỒI TIỀM ẨN

Lưu Nguyễn Anh Thư, Nguyễn Thái Nghe

Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ

lnathu@cit.ctu.edu.vn, ntinghe@cit.ctu.edu.vn

**Tóm tắt:** Hệ thống gợi ý (Recommender Systems) đã và đang được ứng dụng trong rất nhiều lĩnh vực như giải trí, giáo dục, khoa học, và đặc biệt là thương mại điện tử. Việc tích hợp kỹ thuật gợi ý vào các hệ thống trực tuyến nhằm tự động phân tích các hành vi trong quá khứ của người dùng để dự đoán nhu cầu/sở thích của họ trong tương lai, từ đó có những đề xuất hợp lý cho người dùng là rất cần thiết trong thực tế.

Bài viết này đề xuất một giải pháp xây dựng hệ thống gợi ý dành cho bán hàng trực tuyến sử dụng phản hồi tiềm ẩn (implicit feedbacks) từ người dùng. Trước hết chúng tôi đề xuất phương pháp thu thập thông tin phản hồi tiềm ẩn, sau đó tìm hiểu các phương pháp gợi ý phù hợp từ đó đề xuất sử dụng phương pháp tập hợp mô hình để kết hợp các mô hình dự đoán nhằm tăng độ chính xác. Kế đến là việc cài đặt, điều chỉnh, kiểm thử và tích hợp các mô hình đã đề xuất vào hệ thống nhằm gợi ý các sản phẩm phù hợp với sở thích của người dùng. Sau cùng, chúng tôi thu thập phản hồi từ người dùng thực nhằm đánh giá hiệu quả của phương pháp đã đề xuất. Kết quả cho thấy mô hình đề xuất có khả năng gợi ý tốt cho người dùng và hoàn toàn có thể tích hợp vào các hệ thống bán hàng trực tuyến.

**Từ khóa:** Hệ thống gợi ý, bán hàng trực tuyến, phản hồi tiềm ẩn, kỹ thuật phân rã ma trận

## I. GIỚI THIỆU

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, công video clip YouTube (www.youtube.com), gợi ý phim của MovieLens (www.movielens.org),... Việc gợi ý sản phẩm phù hợp sẽ 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 trên sản phẩm – trong trường hợp này gọi là dự đoán xếp hạng (rating prediction) [4] hoặc các phản hồi có thể được xác định một cách không tường minh hay còn gọi là tiềm ẩn (implicit feedbacks) 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 sẽ trực tiếp đánh giá trên sản phẩm, như bình chọn từ ★ (không thích) đến ★★★★★ (rất thích); hay Youtube thu thập thông tin qua like(👍)/ dislike(👎), và các hệ thống khác [3]. Thông qua việc thu thập phản hồi tường minh, hệ thống dễ dàng xác định 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 sẵn sàng/vui lòng 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. Kết quả cho thấy khả năng mà hệ thống gợi ý phù hợp với sở thích của từng người dùng là khá tốt.

## II. HỆ THỐNG GỢI Ý (Recommender Systems - RS)

### A. Hệ thống gợi ý

Mục đích của hệ thống gợi ý 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)** - 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 phản hồi của người dùng trên sản phẩm, thường là các xếp hạng (**rating**).

Thông thường người ta gọi  $U$  là tập tất cả người dùng (users) và  $u$  là một người dùng cụ thể nào đó ( $u \in U$ ).  $I$  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 ( $i \in I$ ).  $I$  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 người dùng.  $R$  là một tập hợp các giá trị dùng để ước lượng ‘sở thích’ (preference) của người dùng, và  $r_{ui} \in R$  ( $R \subseteq \mathfrak{R}$ ) là xếp hạng của người dùng  $u$  trên sản phẩm  $i$ . Giá trị  $r_{ui}$  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)/ dislike (0),...) mà  $u$  đã bình chọn cho  $i$  – trong trường hợp này gọi là dự đoán xếp hạng (rating prediction) hoặc  $r_{ui}$  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  $i$ ,... [2][7]. Bài viết này quan tâm nhiều đến cách xác định  $r_{ui}$  không tường minh.

Các thông tin về user, item, feedback thường được biểu diễn thông qua một ma trận như trong Hình 1. Trong đó mỗi dòng là một người dùng  $u$ , mỗi cột là một sản phẩm  $i$ , và giao giữa dòng và cột là các phản hồi 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 hoặc chọn mua trong quá khứ. Những ô trống là những item chưa được xem (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).

		Items					
		1	2	...	$i$	...	$m$
Users	1	5	3		1	2	
	2		2				4
	:			5			
	$u$	3	4	?	2	1	
	:					4	
$n$			3	2			

**Hình 1.** 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.

Một cách hình thức, gọi  $D^{train} \subseteq U \times I \times R$  là tập dữ liệu huấn luyện,  $D^{test} \subseteq U \times I \times R$  là tập dữ liệu kiểm thử, và một ánh xạ  $r: U \times I \rightarrow R$  ( $(u, i) \mapsto r_{ui}$ )

Mục tiêu của RS là tìm một hàm  $\hat{r}: U \times I \rightarrow \mathfrak{R}$  sao cho  $\xi(r, \hat{r})$  thỏa mãn một điều kiện nào đó. Ví dụ, nếu  $\xi$  là một hàm ước lượng lỗi như Mean Absolute Error (MAE) hay Root Mean Squared Error (RMSE) thì nó cần phải được tối thiểu.

$$MAE = \frac{1}{|D^{test}|} \sum_{(u, i, r) \in D^{test}} |r_{ui} - \hat{r}_{ui}| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{|D^{test}|} \sum_{(u, i, r) \in D^{test}} (r_{ui} - \hat{r}_{(u,i)})^2} \tag{2}$$

Hiện nay, có rất nhiều giải thuật đã được đề xuất cho hệ thống gợi ý, chúng có thể được gom lại theo 3 nhóm sau [1][2][7]:

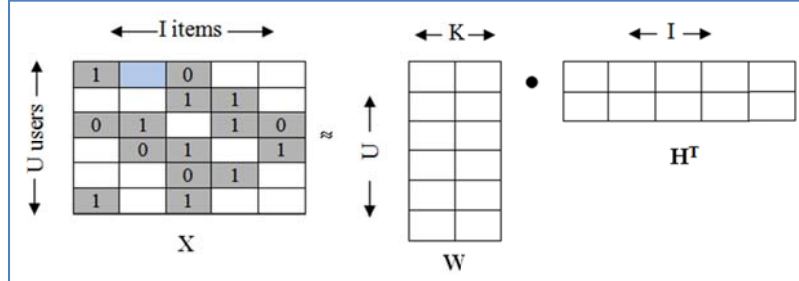
- Gợi ý dựa trên lọc cộng tác (collaborative filtering): người dùng sẽ nhận gợi ý những sản phẩm được ưa thích xuất phát từ những người có cùng thị hiếu và sở thích với mình. Nhóm này dựa vào các phương pháp chủ yếu:
  - o Phương pháp láng giềng (Neighborhood-based, còn gọi là Memory-based), trong đó 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).
  - o 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 dựa trên nội dung (như các thuộc tính) của sản phẩm
- Gợi ý dựa trên cách tiếp cận kết hợp: kết hợp hai phương pháp tiếp cận dựa trên nội dung và lọc 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.

### B. Kỹ thuật phân rã ma trận (matrix factorization – MF)

Kỹ thuật phân rã ma trận (MF) là một trong những phương pháp dựa trên mô hình thành công nhất hiện nay (state-of-the-art) trong RS [1][2]. MF là việc chia một ma trận lớn  $\mathbf{X}$  thành hai ma trận có kích thước nhỏ hơn  $\mathbf{W}$  và  $\mathbf{H}$ , sao cho ta có thể xây dựng lại  $\mathbf{X}$  từ hai ma trận nhỏ hơn này càng chính xác càng tốt [5].

$\mathbf{X} \sim \mathbf{W}\mathbf{H}^T$  như minh họa như trong Hình 2. Trong đó,  $\mathbf{W} \in \mathfrak{R}^{|U| \times K}$  là một ma trận mà mỗi dòng  $u$  là một véc tơ bao gồm  $K$  nhân tố tiềm ẩn (latent factors) mô tả người dùng  $u$ ; và  $\mathbf{H} \in \mathfrak{R}^{|I| \times K}$  là một ma trận mà mỗi dòng  $i$  là một véc tơ bao gồm  $K$  nhân tố tiềm ẩn mô tả cho item  $i$  (lưu ý:  $K \ll |U|$  và  $K \ll |I|$ ).



Hình 2. Minh họa kỹ thuật phân rã ma trận

Gọi  $w_{uk}$  và  $h_{ik}$  là các phần tử tương ứng của hai ma trận  $\mathbf{W}$  và  $\mathbf{H}$ , khi đó xếp hạng của người dùng  $u$  trên mục tin  $i$  được dự đoán bởi công thức:

$$\hat{r}_{ui} = \sum_{k=1}^K w_{uk} h_{ik} = \mathbf{w} \cdot \mathbf{h}^T \quad (3)$$

Như vậy, vấn đề chủ chốt của kỹ thuật MF là làm thế nào để tìm được giá trị của hai tham số  $\mathbf{W}$  và  $\mathbf{H}$ . Hai tham số này có được bằng cách tối ưu hóa hàm mục tiêu (objective function) như RMSE ở công thức (2). Trong RS, hàm mục tiêu của MF hay được sử dụng như sau:

$$O^{MF} = \sum_{u,i \in D^{train}} (r_{ui} - \hat{r}_{ui})^2 = \sum_{u,i \in D^{train}} \left( r_{ui} - \sum_{k=1}^K w_{uk} h_{ik} \right)^2 \quad (4)$$

Một trong những kỹ thuật có thể dùng để tối ưu hóa hàm mục tiêu là dùng SGD (Stochastic Gradient Descent) [5]. Để tối ưu hóa hàm mục tiêu (4), trước tiên ta khởi tạo các giá trị ngẫu nhiên cho  $\mathbf{W}$  và  $\mathbf{H}$ , sau đó từng bước cập nhật giá trị của chúng cho đến khi hàm mục tiêu hội tụ về giá trị nhỏ nhất (convergence). Để làm được điều đó, ta cần phải xác định là nên tăng hay nên giảm các giá trị của  $\mathbf{W}$  và  $\mathbf{H}$  qua mỗi lần cập nhật, do vậy cần phải tìm đạo hàm từng phần của chúng:

$$\frac{\partial}{\partial w_{uk}} O^{MF} = -2(r_{ui} - \hat{r}_{ui})h_{ik} \quad (5)$$

$$\frac{\partial}{\partial h_{ik}} O^{MF} = -2(r_{ui} - \hat{r}_{ui})w_{uk} \quad (6)$$

Sau khi tìm đạo hàm, các phần tử của  $\mathbf{W}$  và  $\mathbf{H}$  sẽ được cập nhật ngược hướng với giá trị của đạo hàm, qua công thức:

$$w_{uk}^{new} = w_{uk}^{old} - \beta \cdot \frac{\partial}{\partial w_{uk}} O^{MF} = w_{uk}^{old} + 2\beta \cdot (r_{ui} - \hat{r}_{ui})h_{ik} \quad (7)$$

$$h_{ik}^{new} = h_{ik}^{old} - \beta \cdot \frac{\partial}{\partial h_{ik}} O^{MF} = h_{ik}^{old} + 2\beta \cdot (r_{ui} - \hat{r}_{ui})w_{uk} \quad (8)$$

Trong đó  $\beta$  là tốc độ học (learning rate,  $0 < \beta < 1$ ). Quá trình cập nhật sẽ được thực hiện đến khi nào hàm mục tiêu đạt được giá trị nhỏ nhất.

**Chính tắc hóa (Regularization):** Để ngăn ngừa sự quá khớp hay còn gọi là học vẹt (overfitting – xảy ra khi mô hình dự đoán cho kết quả tốt trên dữ liệu huấn luyện, nhưng cho kết quả kém trên dữ liệu thử nghiệm) người ta thay đổi hàm mục tiêu (4) bằng cách thêm vào một đại lượng gọi là chính tắc hóa (regularization) để điều khiển độ lớn của các giá trị trong  $\mathbf{W}$  và  $\mathbf{H}$ . Hàm mục tiêu (4) bây giờ trở thành:

$$O^{MF} = \sum_{u, j \in D^{train}} \left( r_{uj} - \sum_{k=1}^K w_{uk} h_{jk} \right)^2 + \lambda \cdot (\|W\|_F^2 + \|H\|_F^2) \quad (9)$$

Trong đó  $\lambda$  là hệ số chính tắc hóa ( $0 \leq \lambda < 1$ ) và  $\|\cdot\|_F^2$  là chuẩn Frobenius. Với hàm mục tiêu mới này, giá trị của  $w_{uk}$  và  $h_{jk}$  được cập nhật qua công thức:

$$w_{uk}^{new} = w_{uk}^{old} + \beta \cdot (2(r_{uj} - \hat{r}_{uj})h_{jk} - \lambda \cdot w_{uk}^{old}) \quad (10)$$

$$h_{jk}^{new} = h_{jk}^{old} + \beta \cdot (2(r_{uj} - \hat{r}_{uj})w_{uk} - \lambda \cdot h_{jk}^{old}) \quad (11)$$

Quá trình cập nhật giá trị của  $\mathbf{W}$  và  $\mathbf{H}$  được lặp lại cho đến khi đạt độ lỗi chấp nhận hoặc lặp lại đến số lần quy định trước.

*Quá trình dự đoán:* Sau quá trình huấn luyện ta được 2 ma trận  $\mathbf{W}$  và  $\mathbf{H}$  đã tối ưu thì quá trình dự đoán được thực hiện theo công thức (3). Ví dụ: Dự đoán xếp hạng của user 2 cho item 2 trong Hình 3:

$$\hat{r}_{ui} = 0.5 \cdot 0.8 + 0.6 \cdot 0.1 = 0.46$$



Hình 3. Minh họa cách dự đoán của MF

Một biến thể của MF dành cho dữ liệu phản hồi tiềm ẩn là kỹ thuật SVD++ [3] sẽ được trình bày trong phần tiếp theo.

### C. Kỹ thuật SVD++

Kỹ thuật SVD++ (Singular Value Decomposition for Implicit Feedback) [3] là một biến thể của MF, đặc trưng của SVD++ là giải quyết khá tốt cho RS với các thông tin phản hồi tiềm ẩn (implicit feedback). Trong thực tế việc thu thập các thông tin phản hồi tường minh (explicit feedback) từ người dùng đôi khi gặp khó khăn vì không phải người dùng nào cũng thích để lại đánh giá của mình trên sản phẩm. SVD++ hướng đến việc xác định mối quan tâm của người dùng  $u$  đến một đối tượng  $i$  mà người dùng  $u$  đã đánh giá nhưng lại không quan tâm đến các giá trị đánh giá đó, những giá trị đánh giá này sẽ được ghi nhận tự động bởi hệ thống, ví dụ số lần chọn mua sản phẩm, hay số lần duyệt xem sản phẩm của người dùng, thời gian xem sản phẩm, ... SVD++ được xây dựng dựa trên cơ sở của Matrix Factorization (MF) chuẩn kết hợp thêm các giá trị lệch (biases) và các nhân tố phản hồi tiềm ẩn [3]. Kết quả dự đoán của SVD++ đã cho thấy có độ chính xác cao, tỉ lệ lỗi thấp hơn MF do nó sử dụng nhiều yếu tố phản hồi để phân tích dự đoán.

Quá trình dự đoán xếp hạng của người dùng  $u$  trên sản phẩm  $i$  trong SVD++ được tính theo công thức:

$$\hat{r}_{ui} = \mu + b_u + b_i + (\mathbf{w}_u + |N(u)|^{-1/2} \sum_{j \in N(u)} \mathbf{y}_j)^T \mathbf{h}_i \quad (12)$$

Trong công thức này  $\mu$  là giá trị trung bình toàn cục,  $b_u$ ,  $b_i$  tương ứng là độ lệch (thiên vị - bias) của người dùng  $u$  và item  $i$  (user bias, item bias),  $\mathbf{w}_u$  và  $\mathbf{h}_i$  là 2 véc tơ nhân tố tiềm ẩn của người dùng  $u$  và item  $i$  như trong kỹ thuật MF.  $N(u)$  là tập các thông tin phản hồi tiềm ẩn của người dùng  $u$ . Giá trị  $\mathbf{y}_j$  là vector chứa các thông tin phản hồi tiềm ẩn của tập người dùng  $N(u)$  trên các items được đánh giá. Vector  $\mathbf{y}_j$  là một ma trận có kích thước bằng số sản phẩm mà người dùng  $u$  đã đánh giá và số  $K$  nhân tố tiềm ẩn. Bạn đọc có quan tâm xin xem chi tiết trong tài liệu [3].

## III. ĐỀ XUẤT MÔ HÌNH GỢI Ý CHO BÁN HÀNG TRỰC TUYẾN SỬ DỤNG PHẢN HỒI TIỀM ẨN

Trước tiên chúng tôi đề xuất phương pháp thu thập thông tin phản hồi tiềm ẩn từ người dùng, sau đó đề xuất giải pháp tích hợp và cài đặt các mô hình dựa trên phương pháp tập hợp mô hình (ensemble models [12]), đề xuất cách xử lý vấn đề người dùng mới (cold-start problem [2]).

### A. Phương pháp thu thập thông tin phản hồi tiềm ẩn

Đa phần các hệ thống bán hàng hiện tại thu thập thông tin phản hồi từ người dùng bằng cách thiết kế sẵn các khung đánh giá cụ thể (thu thập tường minh) cho người dùng chọn mức độ yêu thích sản phẩm, như từ 1 đến 5 sao, hoặc nút

chọn thích/ không thích, khung đánh giá bình luận sản phẩm,... Việc ghi nhận này có thể dẫn đến sự bất tiện cho người dùng và đa phần người dùng ít khi để lại thông tin phản hồi một cách tường minh như thế. Vì thế, **chúng tôi đề xuất cách thu thập thông tin phản hồi tiềm ẩn bằng cách cho hệ thống tự ghi nhận lại các đánh giá của người dùng trên các sản phẩm**, như dựa trên số lần mua và số lần xem sản phẩm của người dùng hệ thống. Ở đó, **thông tin số lần mua sản phẩm** của người dùng sẽ được ghi nhận và cập nhật dựa trên hóa đơn đặt hàng của họ. Nếu một hóa đơn đặt hàng được xét duyệt thì hệ thống cập nhật lại lần mua của khách hàng với các sản phẩm. Còn **thông tin số lần click xem sản phẩm** cho người dùng được chia ra 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:

- 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.
- 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.
- 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 trị dự 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).

### B. Sử dụng thông tin phản hồi tiềm ẩn và tập hợp mô hình dự đoán

Trong các giải thuật chuẩn của hệ thống gợi ý, thường người ta chỉ chọn một giá trị phản hồi của người dùng (như giá trị rating). Tuy nhiên để tận dụng tối đa các thông tin phản hồi nhằm làm tăng độ tin cậy của mô hình dự đoán, trong nghiên cứu này, chúng tôi sử dụng 2 thông tin phản hồi, đặc biệt là phản hồi tiềm ẩn, đó là **số lần mua sản phẩm** và **số lần xem sản phẩm** của khách hàng để tạo tập dữ liệu feedback. Hệ thống sẽ tự động xác định các phản hồi này và dự đoán đưa ra danh sách các item cần gợi ý mà không cần các đánh giá/xếp hạng tường minh từ người dùng. Việc tích hợp 2 thông tin phản hồi này được thực hiện theo phương pháp tập hợp mô hình (ensemble learning [12]) vì vậy nó có thể cho kết quả dự đoán chính xác hơn.

**Mô hình 1:**  $r_{1ui}$  là số lần mua của người dùng  $u$  trên sản phẩm  $i$ , được dự đoán qua công thức

$$\hat{r}_{1ui} = \mu + b_u + b_i + (w_u + |N_1(u)|^{-1/2} \sum_{j \in N(u)} y_j)^T h_i \quad (13)$$

**Mô hình 2:**  $r_{2ui}$  là số lần xem (view/click) của người dùng  $u$  trên sản phẩm  $i$ , được dự đoán qua công thức

$$\hat{r}_{2ui} = \mu + b_u + b_i + (w_u + |N_2(u)|^{-1/2} \sum_{j \in N(u)} y_j)^T h_i \quad (14)$$

Thực tế thấy rằng, một khi người dùng đã mua sản phẩm, ngầm định rằng họ đã thích/cần sản phẩm đó, do vậy trọng số mức độ yêu thích sản phẩm của người dùng trên số lần mua sản phẩm sẽ cao hơn số lần xem. Hơn nữa ở bất kỳ 1 website thương mại điện tử nào, số lần người dùng click/xem sản phẩm là rất lớn trong khi số lần mua nhỏ hơn. Để giảm bớt sự chênh lệch này, chúng tôi đề xuất thông tin về số lần mua có trọng số cao hơn số lần xem là  $\theta$  (trong bài viết này, qua thử nghiệm trên tập dữ liệu ban đầu thu thập được,  $\theta$  được chọn là 4. Lưu ý rằng đây chỉ là 1 heuristic, và có thể xem  $\theta$  như là 1 siêu tham số (hyperparameter) cần phải được xác định trước.

Khi đó, giá trị dự đoán sau cùng được xác định qua công thức:

$$\hat{r}_{ui} = \frac{r_{1ui} + \theta r_{2ui}}{2} \quad (15)$$

Chi tiết về giải thuật SVD++ được cài đặt như trong thủ tục SVDPlusPlusSGD. Ở đây chúng tôi sử dụng Stochastic gradient descent và các hệ số regularization cũng như tốc độ học được xác định riêng cho từng tham số.

```

1: procedure SVDPlusPlusSGD( $D^{Train}$ , Iter, K,  $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ )
// $\lambda_i$ : Regularization;  $\beta_i$ : Tốc độ học (LearnRate); Iter: Số lần lặp (NumIter); K: Số nhân tố tiềm ẩn (NumFactors);
//  $W[|U|][K]$  và  $H[|I|][K]$  là 2 ma trận nhân tố tiềm ẩn cần tìm

2:  $\mu = \frac{\sum_{r \in D^{train}} r}{|D^{train}|}$  // compute the global average

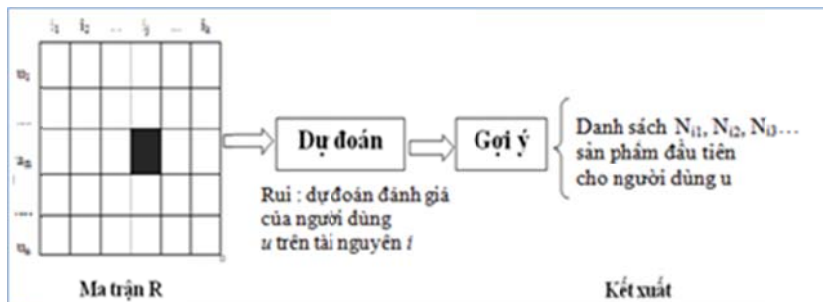
3: for each user u do
4:    $b_u \leftarrow \frac{\sum_u (r_{ui} - \mu)}{|D_u^{train}|}$  // compute user bias

5: end for
6: for each item i do
7:    $b_i \leftarrow \frac{\sum_i (r_{ui} - \mu)}{|D_i^{train}|}$  //compute item bias

8: end for
9:  $W := N(0, \sigma^2)$  // Initialize with normal distribution
10:  $H := N(0, \sigma^2)$ 
11: for (iter:=1; iter <= Iter * |DTrain|; iter++)
12:   Chọn ngẫu nhiên một dòng (u, i,  $r_{ui}$ ) từ  $D^{Train}$ 
13:    $\hat{r}_{ui} := 0$ 
14:   for (k:=1; k<=K; k++)
15:      $\hat{r}_{ui} = \mu + b_u + b_i + (w_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j)^T h_i$ 
16:   end for
17:    $e_{ui} = r_{ui} - \hat{r}_{ui}$ 
18:   for (k:=1; k<=K; k++)
19:      $b_u := b_u + \beta_1 (e_{ui} - \lambda_1 \cdot b_u)$ 
20:      $b_i := b_i + \beta_2 (e_{ui} - \lambda_2 \cdot b_i)$ 
21:      $w_u := w_u + \beta_3 (e_{ui} \cdot h_i - \lambda_3 \cdot w_u)$ 
22:      $h_i := h_i + \beta_4 (e_{ui} \cdot (w_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j) - \lambda_4 \cdot h_i)$ 
23:      $\forall j \in N(u) \ y_j := y_j + \beta_5 (e_{ui} \cdot |N(u)|^{-1/2} \cdot h_i - \lambda_5 \cdot y_j)$ 
24:   end for
25:   Break nếu đã convergence
26: end for
27: return {W, H,  $\mu$ ,  $b_u$ ,  $b_i$ }
28: end procedure

```

Sau khi dự đoán cho tất cả các item, ta chỉ cần sắp xếp lại kết quả giảm dần theo giá trị dự đoán và chọn top-N sản phẩm cần gợi ý như minh họa trong **Hình 4**.



Hình 4. Quy trình gợi ý

### C. Xử lý vấn đề người dùng mới – sản phẩm mới (cold-start problem)

Bất kỳ hệ thống gợi ý nào cũng đều gặp phải khó khăn khi người dùng mới và sản phẩm mới xuất hiện trong hệ thống do ta chưa có thông tin phản hồi trong quá khứ. Để khắc phục vấn đề về người dùng mới và sản phẩm mới trong bài viết này, chúng tôi dùng phương pháp gợi ý toàn cục.

Đối với người dùng mới khi truy cập vào hệ thống (chưa tạo tài khoản hoặc chưa đăng nhập) chúng tôi sẽ gợi ý toàn cục Top-N sản phẩm có giá trị dự đoán trung bình cao nhất cho họ. Đồng thời sẽ tự thêm cho người dùng này một tài khoản để ghi nhận lại thông tin xem sản phẩm của họ, khi người dùng đã click xem đến sản phẩm thứ  $m$  ( $m=3$ ) thì hệ thống sẽ tự động huấn luyện lại mô hình và gợi ý lại các sản phẩm phù hợp với cá nhân người này, quá trình huấn luyện này sẽ được thực hiện lặp lại mỗi 3 lần duyệt xem sản phẩm của người dùng để đáp ứng tính kịp thời của các gợi ý. Chúng tôi không chọn huấn luyện lại mô hình gợi ý sau mỗi lần tác động xem hoặc mua sản phẩm vì việc làm này sẽ làm hiệu năng của hệ thống bị giảm xúc; việc chọn 3 hay nhiều hoặc ít hơn số lần tác động này để huấn luyện lại mô hình gợi ý phụ thuộc vào mục đích của từng hệ thống.

Đối với sản phẩm mới nhập vào, chúng được hiển thị đầu tiên trên trang web và có biểu tượng 'New' để nhận biết đây là sản phẩm mới của hệ thống. Ngoài ra, khi hiển thị chi tiết mỗi sản phẩm, trang web có một không gian để hiển thị các sản phẩm tương tự với sản phẩm mà người dùng đang xem dựa vào một số thuộc tính tương tự. Vì vậy, những sản phẩm mới nhập cũng có thể được gợi ý cho người dùng.

## IV. XÂY DỰNG HỆ THỐNG

Tương tự như những hệ thống thông tin quản lý khác, hệ thống này cũng phải được phân tích, thiết kế các mô hình và sau đó là tích hợp các giải thuật gợi ý vào hệ thống. Do giới hạn về số trang của bài viết, chúng tôi chỉ giới thiệu một vài mô hình cơ bản như bên dưới.

Bên cạnh chức năng gợi ý sản phẩm, hệ thống mong đợi sẽ có các chức năng khác như cho khách hàng đặt hàng trực tiếp, ngoài ra hệ thống còn cung cấp các công cụ quản trị như: quản trị sản phẩm, quản trị hóa đơn đặt hàng; quản trị người dùng; quản trị banner cho phép người quản trị thực hiện thêm, sửa và xóa: banner hệ thống chung, slider quảng cáo, banner tin tức, thông tin liên khuyến mãi,... theo từng vị trí; quản trị huấn luyện dữ liệu, kiểm tra mô hình SVD++. Khách hàng muốn thực hiện các chức năng trên chỉ khi khách hàng là thành viên của hệ thống. Muốn trở thành thành viên của hệ thống, khách hàng phải đăng ký tài khoản thông qua hệ thống.

Một ví dụ về sơ đồ use case của khách hàng được trình bày trong **Hình 5** và lược đồ cơ sở dữ liệu của hệ thống được trình bày trong **Hình 6**.

Hệ thống được thiết kế theo mô hình Client –Server. Khi người dùng gửi yêu cầu đến Web server thông qua Web Browser được cài đặt tại máy tính người dùng. Web server gửi yêu cầu xử lý của người dùng đến Application Server. Application Server xử lý yêu cầu và có thể truy xuất dữ liệu từ SQL Database nếu cần. Sau khi xử lý xong, Application Server sẽ trả kết quả về cho Web server để chuyển cho người dùng.

## V. KẾT QUẢ MINH HỌA

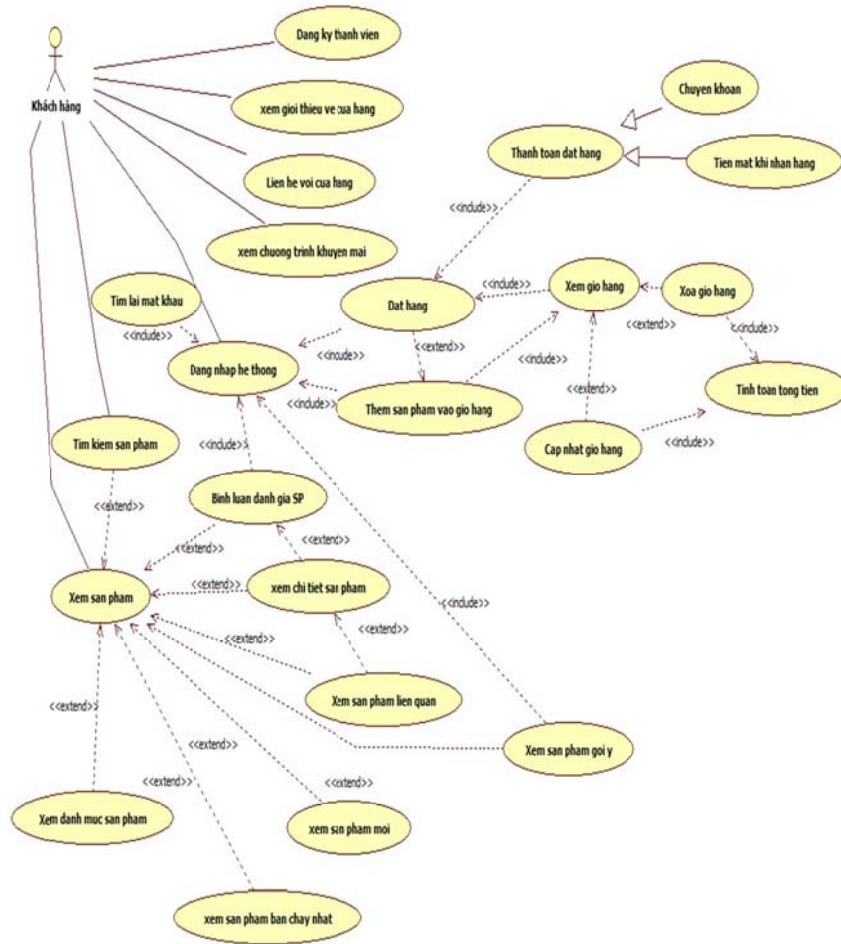
Để minh họa, mô hình đã đề xuất được áp dụng vào bài toán bán hàng trực tuyến, liên quan đến sản phẩm tin học và linh kiện máy tính. Kết quả minh họa cho các giao diện chính của hệ thống và kết quả đánh giá độ chính xác được trình bày trong các phần dưới đây.

### A. Một số giao diện minh họa

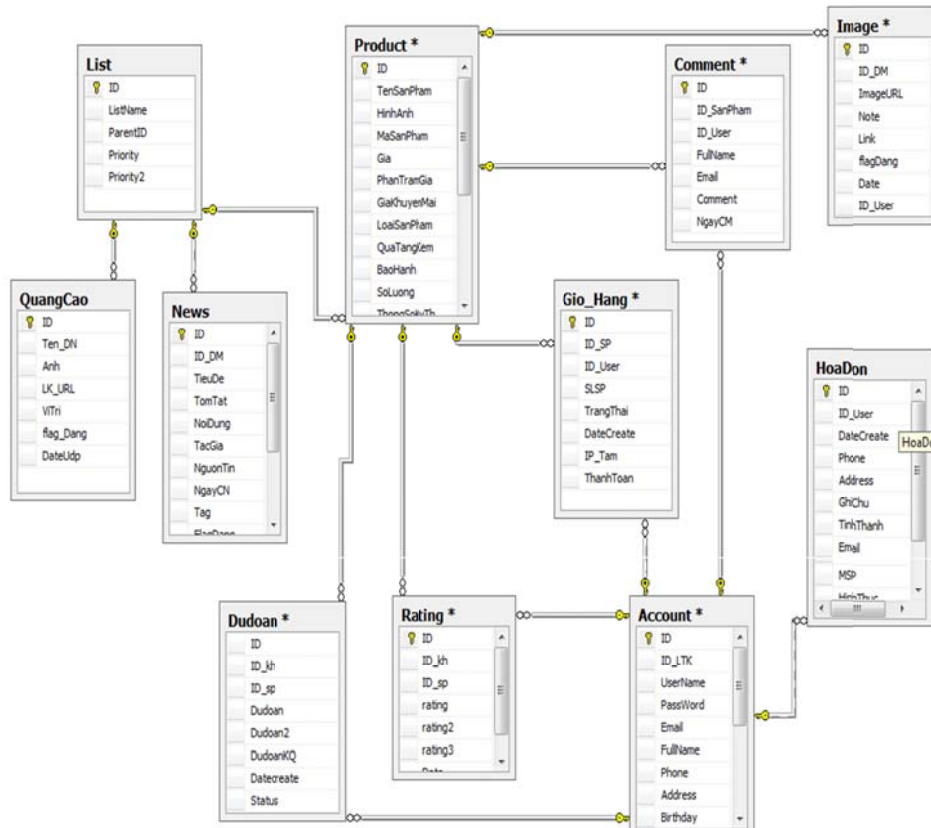
Hệ thống gợi ý bán hàng trực tuyến được cài đặt theo mô hình ứng dụng web. Hệ thống thực hiện ghi nhận thông tin phản hồi từ người dùng (số lần click xem sản phẩm, số lần mua sản phẩm) để làm dữ liệu huấn luyện cho xây dựng mô hình gợi ý. Khi mô hình huấn luyện được xây dựng hoàn tất, hệ thống sẽ thực hiện dự đoán mức độ yêu thích của người dùng trên các sản phẩm, đưa ra gợi ý TOP-N các sản phẩm mới. Hệ thống gợi ý sản phẩm này, còn gợi ý cho người dùng mới theo giá trị dự đoán trung bình toàn cục của các sản phẩm, đồng thời liệt kê những sản phẩm tương tự với sản phẩm đang xem, bên cạnh đó còn thể hiện các sản phẩm bán chạy nhất, sản phẩm mới nhất.

*Gợi ý sản phẩm cho khách hàng:* Sau khi người dùng đăng nhập vào hệ thống, sẽ nhận được các sản phẩm gợi ý dựa trên các phản hồi mà hệ thống ghi nhận được từ người dùng như minh họa trong **hình 7**.

*Huấn luyện lại mô hình gợi ý:* Chúng tôi cũng xây dựng công cụ hỗ trợ admin huấn luyện lại toàn bộ mô hình gợi ý sau một thời gian sử dụng như minh họa trong **Hình 8**. Ngoài ra còn rất nhiều tính năng khác như quản lý khách hàng, giao hàng, thanh toán,... tương tự như bất kỳ một hệ thống thương mại điện tử nào khác.



Hình 5. Sơ đồ use case của khách hàng



Hình 6. Lược đồ cơ sở dữ liệu của hệ thống



### B. Đánh giá kết quả của các mô hình dự đoán

Chúng tôi thu thập thông tin phản hồi từ người dùng thực, tập dữ liệu thu được gồm 186 người dùng, 174 sản phẩm và 637 đánh giá (số lần mua lớn nhất là 4, số lần xem sản phẩm lớn nhất là 12.25).

#### a. So sánh kết quả sử dụng thông tin phản hồi tiềm ẩn

Kết quả kiểm tra dùng độ đo MAE cho các mô hình ở công thức (13), (14), (15) được trình bày trong **Bảng 1**. Thực nghiệm cho thấy rõ ràng rằng khi sử dụng kết hợp cả 2 thông tin phản hồi từ người dùng, độ lỗi MAE giảm đi đáng kể (chỉ có 0.1878 trong khi chỉ sử dụng một thông tin phản hồi độ lỗi lần lượt là 0.4087 và 0.3464)

**Bảng 1.** So sánh tỉ lệ lỗi MAE của các mô hình

Stt lần kiểm tra	$\hat{r}_{1ui}$	$\hat{r}_{2ui}$	$\hat{r}_{ui}$
1	0.4306	0.3533	0.1848
2	0.4399	0.3365	0.1922
3	0.4373	0.3466	0.1873
4	0.4284	0.3416	0.1917
5	0.4349	0.3424	0.1942
6	0.4386	0.3431	0.1883
7	0.4317	0.3553	0.1864
8	0.4317	0.3519	0.1837
9	0.4365	0.3486	0.1829
10	0.4470	0.3453	0.1869
<b>Trung bình</b>	<b>0.4087</b>	<b>0.3464</b>	<b>0.1878</b>



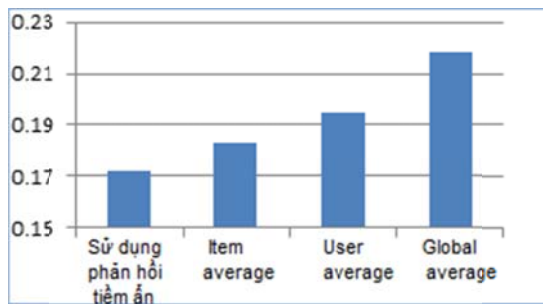
**Hình 7.** Gợi ý sản phẩm cho khách hàng



**Hình 8.** Giao diện hỗ trợ huấn luyện lại các mô hình

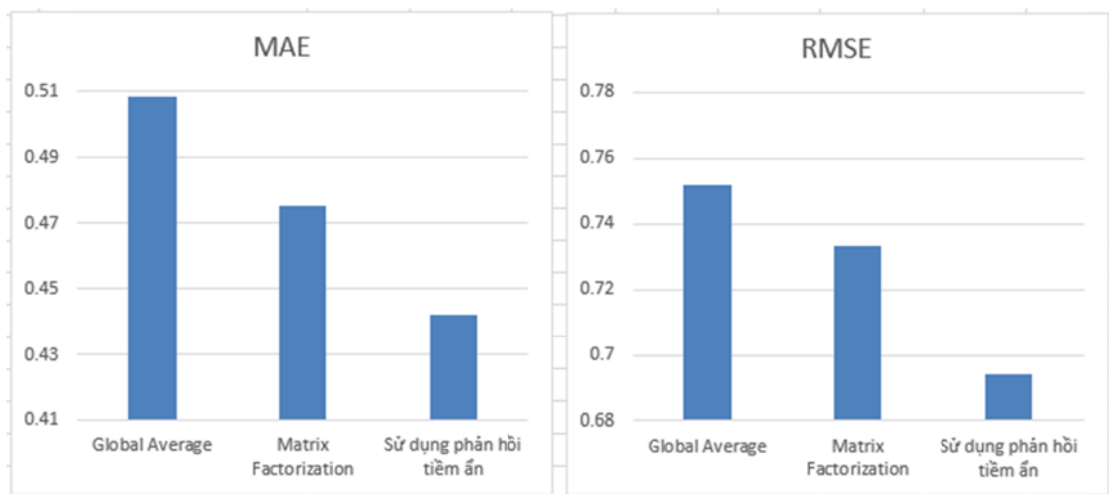
*b. So sánh kết quả với các giải thuật nền (baseline) khác*

Kết quả so sánh trên độ đo chuẩn Mean Absolute Error (MAE) cho khi sử dụng phương pháp dành cho thông tin phân hồi tiềm ẩn, tỷ lệ lỗi cũng thấp hơn các phương pháp nền khác như: Global Average, User Average, Item Average [2][7] như minh họa trong **Hình 9**.



**Hình 9.** Độ lỗi MAE của các giải thuật dự đoán trên tập dữ liệu thu thập từ người dùng thực của hệ thống

Bên cạnh đó, chúng tôi cũng so sánh phương pháp sử dụng phân hồi tiềm ẩn với các phương pháp khác trên tập dữ liệu benchmark cung cấp bởi cộng đồng người dùng nghiên cứu về hệ thống gợi ý, các tập dữ liệu này có tại địa chỉ [http://www.recsyswiki.com/wiki/Grocery\\_shopping\\_datasets](http://www.recsyswiki.com/wiki/Grocery_shopping_datasets). Chúng tôi sử dụng tập dữ liệu Ta-Feng<sup>1</sup> có tên là D12, đã được thu thập dữ liệu bán hàng trong tháng 12 năm 2000, bao gồm 15447 users, 1780 items, và 178216 ratings. Kết quả trên độ đo lỗi MAE và RMSE được trình bày trong **Hình 10**. Kết quả này cũng cho thấy sử dụng phân hồi tiềm ẩn cho độ lỗi thấp hơn các phương pháp baseline khác.



**Hình 10.** Độ lỗi MAE và RMSE của các giải thuật trên tập dữ liệu Ta-Feng

<sup>1</sup> <http://stackoverflow.com/questions/25014904/download-link-for-ta-feng-grocery-dataset>

### c. Đánh giá hiệu quả của hệ thống gợi ý sản phẩm cho người dùng

Ngoài việc đánh giá độ chính xác của các dự đoán theo độ đo lỗi như trên, chúng tôi cũng đánh giá hiệu quả của việc gợi ý xem nó có phù hợp cho mỗi người dùng hay không. Gợi ý được xem là phù hợp khi người dùng có chọn sản phẩm từ danh sách những sản phẩm đã được gợi ý cho họ, dựa theo mô tả trong tài liệu [9]. Để thực hiện điều đó, chúng tôi tiến hành kiểm tra kết quả dự đoán Top-K sản phẩm cho người dùng trên tập dữ liệu thu được. Quá trình được thực hiện theo các bước:

- Tạo tập dữ liệu train và test theo từng user. Với mỗi user chọn 70% dữ liệu cho train, 30% còn lại dùng để test.
- Tiến hành huấn luyện mô hình trên tập dữ liệu train đã tạo.
- Dự đoán cho từng user trên tất cả các item không có trong tập train. Lấy Top-K (K=15) sản phẩm có giá trị dự đoán cao nhất để kiểm tra, so sánh các giá trị này với tập dữ liệu test. Với mỗi lần gợi ý Top-K như thế, nếu các item này có trong tập test của user tương ứng, xem như lần gợi ý đó là phù hợp.
- Lặp lại cho tất cả các user được chọn thử nghiệm.

Do hệ thống mới chỉ cài đặt, lượng user thực chưa nhiều nên chúng tôi chọn ngẫu nhiên 5 user để thử nghiệm, dữ liệu minh họa trong **Bảng 2** sau:

**Bảng 2.** Thống kê số lượng đánh giá của các user dùng kiểm tra độ chính xác gợi ý hệ thống

User	Tổng số đánh giá	Số đánh giá dùng để train	Số đánh giá dùng để test
User 21	43	32	11
User 22	34	24	10
User 46	11	8	3
User 48	7	5	2
User 56	6	4	2

**Bảng 3.** Thống kê số sản phẩm dự đoán của các user trong Top-15 có xuất hiện trong tập test qua các lần dự đoán

STT lần test	User 21		User 22		User 46		User 48		User 56	
	Số sp	Mã sp	Số sp	Mã sp	Số sp	Mã sp	Số sp	Mã sp	Số sp	Mã sp
1	2	134, 164	1	38	1	130	0		0	
2	2	134, 144	2	38, 70	1	130	1	33	0	
3	3	134, 144, 164	22	38	1	30	0		0	
4	2	134, 164	2	35, 38	1	130	0		0	
5	1	164	0		1	130	0		0	
6	1	164	1	38	1	130	0		0	
7	2	134, 164	1	158	2	130, 105	1	33	0	
8	1	134	0		1	130	1	33	0	
9	2	134, 164	1	38	1	130	0		0	
10	0		1	38	1	130	0		0	
<b>TB</b>	<b>90%</b>		<b>80%</b>		<b>100%</b>		<b>30%</b>		<b>0%</b>	

Thử nghiệm trên 10 lần chạy, với mỗi lần lấy Top-15 sản phẩm dự đoán để kiểm tra trong tập test, kết quả trình bày như trong **Bảng 3**. Trong bảng này, mỗi cột là một người dùng, mỗi hàng là kết quả thống kê số lượng sản phẩm gợi ý trong Top-15 có xuất hiện trong tập test với các mã sản phẩm cụ thể qua một lần chạy kiểm tra của từng người dùng. Ví dụ: ở lần kiểm tra thứ nhất, các sản phẩm được gợi ý cho user 21 có xuất hiện trong tập test là 2 sản phẩm với mã sản phẩm cụ thể 134, 164. Như vậy, trong lần gợi ý này, user 21 nhận có sản phẩm phù hợp (chính xác) với sở thích của mình. Lặp lại tương tự cho các user khác.

Từ kết quả này, chúng tôi thấy rằng độ chính xác của kết quả gợi ý qua mỗi lần kiểm tra khá cao đối với các người dùng là thành viên thường xuyên của hệ thống và số lượng đánh giá sản phẩm nhiều. Trong bảng thống kê này có 5 người dùng trong đó có 3 người dùng (user 21, user 22, user 46) là khách hàng thường xuyên đến hệ thống có đăng ký tài khoản và số lượng đánh số sản phẩm nhiều nên độ chính xác cao. Có 2 người dùng (user 48, user 56) là khách hàng vắng lại được ghi nhận bằng địa chỉ IP khi truy cập, họ chỉ đến hệ thống một lần do vậy chưa đủ thông tin để mô hình cho kết quả dự đoán tốt.

## VI. KẾT LUẬN

Bài viết này đã đề xuất 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.

## VII. TÀI LIỆU THAM KHẢO

- [1] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (June 2015), 99-154. DOI=10.1007/s11257-015-9155-5
- [2] Ricci, F., Rokach, L., Shapira, B. & Kantor, P.B., eds. (2011). *Recommender Systems Handbook*. Springer.
- [3] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08)*. ACM, New York, NY, USA, 426-434. DOI=10.1145/1401890.1401944
- [4] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining (ICDM 2008)*, pages 263-272, 2008.
- [5] Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *IEEE Computer Society Press*, 42(8), pp 30-37, 2009.
- [6] Nguyễn Hùng Dũng và Nguyễn Thái Nghe. 2014. Hệ thống gợi ý sản phẩm trong bán hàng trực tuyến sử dụng kỹ thuật lọc cộng tác. *Tạp chí Khoa học Trường Đại học Cần Thơ*, số 31a (2014), trang 36-51. ISSN: 1859-2333.
- [7] Nguyen Thai-Nghe. 2013. An introduction to factorization technique for building recommendation systems. Vol. 6/2013, pp. 44-53, *Journal of Science - University of Da Lat*, ISSN 0866-787X
- [8] Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., and Schmidt-Thieme, L.(2010a). Recommender system for predicting student performance. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2811–2819. Elsevier's *Procedia CS*.
- [9] Guy Shani and Asela Gunawardana. *Evaluating Recommendation Systems*. November 2009.
- [10] Rendle, S., Freudenthaler, C., Gantner, Z., Lars, S. T.: Bpr: Bayesian personalized ranking from implicit feedback. In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, AUAI Press (2009).
- [11] Takacs, G., Pillaszy, I., Nemeth, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems (special topic on mining and learning with graphs and relations). *Journal of Machine Learning Research*, 10, 623-656.
- [12] Thomas G. Dietterich, *Ensemble Methods in Machine Learning*. Lecture Notes in Computer Science Volume 1857, 2000, pp 1-15. Springer.
- [13] Su, X. & Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, (4) 1-19, 2009.
- [14] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. MyMediaLite: a free recommender system library. In *Proceedings of the fifth ACM conference on Recommender systems (RecSys '11)*. ACM, New York, NY, USA, 305-308. DOI=10.1145/2043932.2043989

## A METHOD FOR BUILDING A PRODUCT RECOMMENDATION SYSTEM USING IMPLICIT FEEDBACKS

Luu Nguyễn Anh Thư, Nguyễn Thái Nghe

**Abstract:** *Recommender Systems are widely used in many areas such as entertainment, education, science, especially e-commerce. Integrating recommender system techniques to online shopping systems to recommend suitable products to users is really useful and necessary. In this work, we propose an approach for building an online shopping system with integrating recommender system technique, especially using implicit feedbacks from the users.*

*For building the system, first we propose a method to collect the users' implicit feedbacks. Then, we propose an ensemble method which combine several extended matrix factorization models which are specialized for those implicit feedbacks. Next, we analyze, design, and implement an online system to integrate the aforementioned recommendation techniques. After having the system, we collect the feedbacks from the real users to validate the proposed approach. Results show that this approach is feasible and can be applied for the real systems.*

**Keywords:** *Recommender systems, product recommendation, implicit feedback, matrix factorization*