

XÂY DỰNG HỆ THỐNG GỢI Ý BÀI HÁT DỰA TRÊN PHẢN HỒI TIỀM ẨN

Nguyễn Thái Nghe¹ và Nguyễn Tấn Phong²

¹ Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

² Cơ quan thường trú Đài Tiếng Nói Việt Nam, Khu vực Đồng bằng sông Cửu Long

Thông tin chung:

Ngày nhận: 22/06/2014

Ngày chấp nhận: 30/10/2014

Title:

Building a music recommendation based on implicit feedbacks

Từ khóa:

Hệ thống gợi ý, gợi ý bài hát, phản hồi tiềm ẩn, giải thuật BPR-MF

Keywords:

Recommender systems, music recommendation, implicit feedback, BPR-MF

ABSTRACT

Recommender systems is widely used in predicting user's preferences based on their feedbacks to recommended new items that the users may like. Recommender systems are applied in many different fields such as e-commerce (online shopping), entertainment (movies, music, etc.), and education (recommend learning resources such as books, newspapers, etc.).

In this paper, we propose a solution for building a music recommender system using implicit feedbacks so that the system can recommend suitable songs to the users. We develop a new system and integrate recommendation algorithm to this system. We collect the feedbacks from the real users and evaluate the proposed solution based on their feedbacks. Results show that our solution can be used for many current online music systems.

TÓM TẮT

Hệ thống gợi ý (recommender systems – RS) thường được sử dụng để dự đoán sở thích của người dùng dựa vào những phản hồi (feedbacks) của họ nhằm gợi ý các sản phẩm (item) mà người dùng có thể thích. RS hiện đang được ứng dụng ở rất nhiều lĩnh vực khác nhau như: trong thương mại điện tử (bán hàng trực tuyến), trong giải trí (âm nhạc, phim ảnh...), trong giáo dục đào tạo (gợi ý nguồn tài nguyên học tập như: sách, báo,...).

Trong bài viết này, chúng tôi giới thiệu một giải pháp xây dựng Hệ thống gợi ý bài hát dựa vào phản hồi tiềm ẩn (implicit feedback) từ người dùng để gợi ý những bài hát mà họ có thể thích nghe. Chúng tôi lựa chọn phương pháp biểu diễn dữ liệu, cài đặt và tích hợp giải thuật gợi ý vào hệ thống, thu thập phản hồi từ người dùng và đánh giá hiệu quả của hệ thống dựa trên những phản hồi đó. Thử nghiệm cho thấy giải pháp này hoàn toàn có thể tích hợp vào các hệ thống gợi ý âm nhạc hiện có trên thị trường.

1 GIỚI THIỆU

Hệ thống gợi ý (Recommender systems – RS) đang từng bước trở thành một lĩnh vực nghiên cứu quan trọng và được ứng dụng khá thành công trong thực tiễn, giúp người dùng đối phó với vấn đề quá tải thông tin (Goldberg *et al.*, 1992, Ricci *et al.*, 2011). Hiện nay, RS được nghiên cứu và ứng dụng trong nhiều lĩnh vực khác nhau như: thương mại

điện tử (bán hàng trực tuyến), giải trí (phim ảnh, âm nhạc,...), giáo dục đào tạo (gợi ý nguồn tài nguyên học tập như sách, báo,...),...

Trên thế giới, đã có nhiều công ty, tổ chức đã áp dụng thành công hệ thống gợi ý nhằm gợi ý các dịch vụ, sản phẩm và các thông tin cần thiết đến người dùng như: website mua sắm trực tuyến Amazon.com gợi ý cho mỗi khách hàng những sản

phẩm mà họ có thể quan tâm, YouTube.com giới thiệu các video clip cho người xem, gợi ý phim ảnh của Netflix.com, MovieLens.org và gợi ý nhạc của Last.fm,... Điều này 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 mong muốn để dàng hơn.

Ở nước ta, hiện nay đã có nhiều hệ thống (website) có chức năng gợi ý như hệ thống gợi ý sản phẩm, bài hát,... tuy nhiên đa phần các hệ thống này chỉ gợi ý dựa trên những thông tin cơ bản như: gợi ý dựa trên nội dung giống nhau (như gợi ý các bài hát của cùng ca sĩ, cùng thể loại,... ở các website nhaccuatui.vn, nhacso.net,...) hay cùng loại sản phẩm, cùng nhà cung cấp,... (www.vatgia.com, www.enbac.com,...). Qua tìm hiểu, vẫn chưa thấy có nhiều hệ thống có tích hợp các kỹ thuật state-of-the-art, như Matrix Factorization (Koren, 2009) trong hệ thống gợi ý. Thương mại điện tử ở Việt Nam đang bước đầu phát triển, do vậy hệ thống gợi ý sẽ mở ra nhiều tiềm năng trong tương lai, kể cả trong nghiên cứu và ứng dụng.

Cùng với sự phát triển mạnh mẽ của các loại hình truyền thông đa phương tiện thì âm nhạc là một trong những nội dung khá phổ biến và được xem như là một nhu cầu không thể thiếu trong cuộc sống, có thể chia sẻ bởi nhiều người từ nhiều quốc gia có ngôn ngữ và nền văn hóa khác nhau. Tuy nhiên, số lượng bài nhạc đang ngày càng tăng lên, đa dạng và phong phú cả về nội dung lẫn thể loại. Vì vậy, vấn đề đặt ra là khi một người sử dụng muốn tìm nghe những bài nhạc mà mình yêu thích, người sử dụng sẽ cần đến công cụ tìm kiếm Google và/hoặc vào một website về âm nhạc để tìm nghe. Mặc dù vậy, ở đó có nhiều bản nhạc mà người sử dụng sẽ không thể nghe thử hết để tìm ra những bài mà họ thích (điều này tốn thời gian mà lại không hiệu quả). Do đó, nhu cầu cần có một hệ thống gợi ý có khả năng dự đoán mức độ ưa thích của người sử dụng với từng bản nhạc và gợi ý cho họ các bản nhạc mới mà hệ thống cho là phù hợp.

Các hệ thống gợi ý thường sử dụng kỹ thuật lọc cộng tác (collaborative filtering) để đưa ra các dự đoán về sở thích của người dùng (user) đối với các mục tin (items – như sản phẩm, sách, báo, phim,...) mà hệ thống cho là phù hợp nhất thông qua việc sử dụng những xếp hạng (rating/feedback) trong quá khứ của người dùng và/hoặc những xếp hạng của những người dùng khác đã có trong cơ sở dữ liệu. Lọc cộng tác thường được tiếp cận theo 2 dạng: lọc cộng tác dựa vào bộ nhớ, cụ thể là tiếp cận dựa trên

người dùng (user-based) và dựa trên các mục tin (item-based). Dạng thứ 2 là tiếp cận theo mô hình (như mô hình Bayes,...) và gần đây là các mô hình nhân tố tiềm ẩn (latent factor models) như mô hình phân rã ma trận (Koren *et al.*, 2009) (matrix factorization - đã đạt được những thành công đáng kể).

Trong bài viết này, chúng tôi giới thiệu một giải pháp trong xây dựng Hệ thống gợi ý bài hát dựa vào phản hồi tiềm ẩn (implicit feedback) từ người dùng. Đây là một dạng trong bài toán gợi ý mục tin (item recommendation) (Ricci *et al.*, 2011). Mặc dù, có khá nhiều phương pháp đã được sử dụng cho vấn đề gợi ý mục tin dựa vào phản hồi tiềm ẩn, phổ biến nhất kỹ thuật lọc cộng tác k-láng giềng (kNN), thường dùng tương quan pearson hoặc cosine để tính toán độ tương tự giữa những người dùng cùng sở thích hoặc các mục tin được xếp hạng bởi cùng người dùng. Gần đây, kỹ thuật phân rã ma trận được sử dụng phổ biến và ứng dụng rất thành công. Tuy nhiên, các giải thuật này thích hợp hơn cho những hệ thống sử dụng phản hồi tường minh (explicit feedback) thông qua những xếp hạng/đánh giá cụ thể.

Ở đây, chúng tôi đề xuất sử dụng phản hồi tiềm ẩn từ người dùng (như tỷ lệ thời lượng mà người dùng đã nghe trên tổng thời lượng bài hát) và từ đó đề xuất sử dụng giải thuật xếp hạng cá nhân Bayes (Rendle *et al.*, 2009) để xây dựng “Hệ thống gợi ý bài hát” do giải thuật này hoạt động tốt trên dữ liệu phản hồi tiềm ẩn.

2 HỆ THỐNG GỢI Ý (RECOMMENDER SYSTEMS - RS)

Hệ thống gợi ý thường dựa vào ba khái niệm cơ bản, đó là tập người dùng U (user), tập mục tin I (item – sản phẩm, bài hát, video clip,...) và phản hồi r_{ui} (feedback/ xếp hạng - rating) của người dùng u trên mục tin i .

		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 mục tin (user-item-rating matrix)

Có 2 dạng bài toán chính trong RS là dự đoán xếp hạng (rating prediction) dựa vào các giá trị xếp hạng trước đó (là những phản hồi tường minh của người dùng, như xếp hạng từ 1 đến 5, hoặc thích/không thích, hay 0/1,...) để dự đoán các giá trị xếp hạng trong tương lai, và gợi ý mục tin (item recommendation) dựa vào các phản hồi tiềm ẩn từ người dùng (như số lần click chuột, thời gian xem sản phẩm,... nghĩa là những thông tin mà người dùng không trực tiếp đánh giá/xếp hạng).

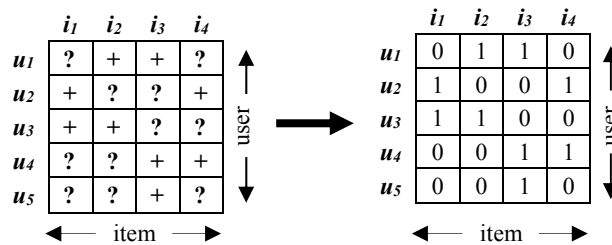
Với bài toán dự đoán xếp hạng, thông tin thường được biểu diễn trong một ma trận như ở Hình 1. Ở đó, mỗi dòng là một user, mỗi cột là một item, và mỗi ô là một giá trị xếp hạng biểu diễn phản hồi (như “mức độ thích - preference”) của user trên item tương ứng. Các ô có giá trị là những item mà các user đã xếp hạng trong quá khứ. Những ô trống là những item chưa được xếp hạng (điều đáng lưu ý là mỗi user chỉ xếp hạng cho một vài item trong quá khứ, do vậy có rất nhiều ô trống trong ma trận này – còn gọi là ma trận cực thưa – sparse matrix).

Như vậy, nhiệm vụ chính của RS là dựa vào các ô đã có giá trị trong ma trận (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 cho người dùng.

3 PHƯƠNG PHÁP BIỂU DIỄN DỮ LIỆU TIỀM ẨN CHO HỆ THỐNG GỢI Ý BÀI HÁT

Với bài toán gợi ý mục tin, RS sử dụng phản hồi tiềm ẩn từ người dùng, do đó nó thường chỉ chứa dữ liệu tương tác/quan sát dương (positive observations) hay gọi là sự phản hồi tích cực/dương (positive feedback) $S \subseteq U \times I$. Còn những mục tin mà người dùng chưa quan sát/tương tác (none-observed) là sự trộn lẫn giữa giá trị phản hồi âm (negative feedback – người dùng không thích mục này) và những giá trị thiếu (missing values – người dùng có thể thích mục này trong tương lai do họ chưa thấy/tương tác với chúng) (Rendle *et al.*, 2009) như biểu diễn bên tay trái trong Hình 2. Do đó, cách thông thường để tạo ra tập dữ liệu huấn luyện là với những cặp $(u, i) \in S$ thuộc lớp dương (positive class) sẽ được gán giá trị 1, phần còn lại thuộc về lớp âm (negative class) được gán giá trị 0, như bên tay phải của Hình 2 (Hu *et al.*, 2008).



Hình 2: Biểu diễn dữ liệu trong RS dưới dạng nhị phân

Tuy nhiên, bất lợi lớn của phương pháp này là trong suốt quá trình huấn luyện, các mô hình sẽ không phân biệt được đâu là phản hồi âm (negative feedbacks – tức người dùng không thích) và đâu là giá trị cần dự đoán do cả missing values và negative values đều được xem là giá trị 0.

Trong hệ thống gợi ý bài hát được đề xuất, chúng tôi sử dụng phương pháp như trình bày trong tài liệu (Rendle *et al.*, 2009), là sử dụng phương pháp so sánh từng đôi (pair-wise ranking) thay vì chỉ đưa ra dự đoán xếp hạng cho từng mục tin hoặc chỉ thay thế những giá trị cần xếp hạng bằng một giá trị âm.

Cụ thể, từ dữ liệu đã có S tiến hành xây dựng lại tập dữ liệu huấn luyện D_S dựa vào mối quan hệ

“thích hơn – prefer” giữa các bài hát cho mỗi người dùng u ($>_u$). Giả sử rằng nếu một bài hát i được nghe bởi người dùng u ($(u, i) \in S$) thì người dùng thích bài này hơn tất cả các bài chưa nghe khác. Ví dụ trong Hình 3 bên trái, biểu diễn quan hệ “thích hơn” cho các bài hát của người dùng u_1 như:

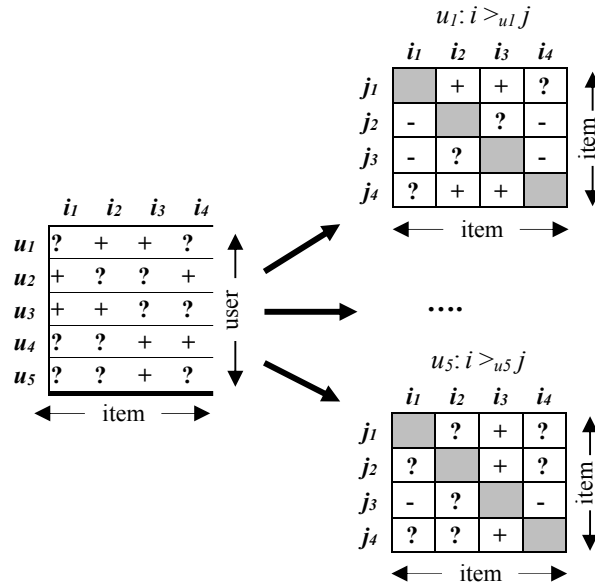
$$i_2 >_{u_1} i_1 ; i_2 >_{u_1} i_4 ; i_3 >_{u_1} i_1 ; i_3 >_{u_1} i_4$$

Với những bài hát mà người dùng đã nghe (như bài i_2 và i_3 ứng với người dùng u_1 trong Hình 3) thì hệ thống sẽ không sử dụng so sánh thích hơn như mô tả ở trên do hệ thống ngầm hiểu là người dùng có cùng mức độ thích trên những bài này. Tương tự, với những bài hát mà người dùng chưa nghe (ví dụ, bài i_1 và i_4 ứng với người dùng u_1) hệ thống cũng không sử dụng quan hệ “thích hơn – prefer” do chưa có thông tin.

Bên phải trong Hình 3 là cách biểu diễn dữ liệu được sử dụng trong hệ thống. Dấu + thể hiện người dùng thích bài i hơn bài j (lưu ý: gọi j để dễ theo dõi, nhưng bản chất i_1 và j_1 là như nhau), dấu - thể hiện người dùng thích bài j hơn bài i và dấu ? là

những cặp các bài hát cần được xếp hạng sau này.

Như vậy, hệ thống gợi ý bài hát cần dự đoán cho các giá trị ? trong các ma trận bên phải của Hình 3, sau đó sắp xếp chúng từ cao xuống thấp và chọn ra N bài cần gợi ý.



Hình 3: Biểu diễn dữ liệu D_S dạng so sánh từng đôi (Rendle et al., 2009)

Tập D_S được biểu diễn một cách hình thức như sau (I_u^+ là các bài hát mà u đã nghe và có phản hồi dương - thích):

$$D_S := \{(u, i, j) \mid i \in I_u^+ \wedge j \in I \setminus I_u^+\}$$

Trong hệ thống gợi ý bài hát mà chúng tôi xây dựng, thông tin phản hồi tiềm ẩn từ người dùng sẽ được hệ thống ghi nhận lại một cách tự động. Cụ thể, thay vì người dùng phải xếp hạng hay đánh giá trên mỗi bài hát, ở đây hệ thống sẽ ghi nhận lại *thời lượng* mà họ đã nghe bài hát đó (thời gian càng lâu đồng nghĩa với người dùng càng thích, dĩ nhiên cũng có ngoại lệ nhưng không đáng kể) và do vậy xem thời gian nghe là thông tin phản hồi từ người dùng.

Do mỗi bài hát có thời lượng ngắn/dài khác nhau nên chúng tôi chuyển đổi thời lượng mà người dùng đã nghe thành dạng *tỷ lệ* thời lượng. Đây cũng chính là thông tin phản hồi từ người dùng u cho bài hát i :

$$r_{ui} = \frac{\text{thời lượng mà } u \text{ đã nghe bài hát } i}{\text{thời lượng bài hát } i} \quad (1)$$

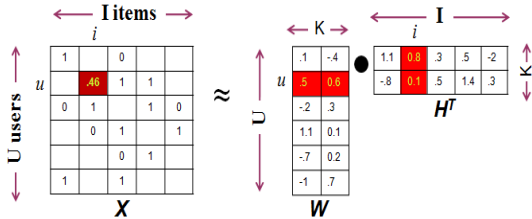
Ví dụ, với bài hát người dùng đã nghe 5^{phút}/5^{phút} (tỷ lệ là 1.0) được ngầm định rằng họ thích nó hơn là bài hát mà họ chỉ 1^{phút}/5^{phút} (tỷ lệ = 0.2).

4 GIẢI THUẬT SỬ DỤNG CHO HỆ THỐNG GỢI Ý BÀI HÁT VỚI PHẢN HỒI TIỀM ẨN

Do tính chất của dữ liệu như đã trình bày, chúng tôi đề xuất sử dụng giải thuật xếp hạng cá nhân Bayes (Bayesian Personalize Ranking – BPR) cho hệ thống gợi ý bài hát có sử dụng dữ liệu tiềm ẩn. BPR là một tiêu chuẩn dùng để tối ưu hóa (optimization criteria) hàm mục tiêu, do đó nó có thể áp dụng cho nhiều kỹ thuật khác nhau trong RS. Ở đây chúng tôi sử dụng BPR-MF (dùng tiêu chuẩn tối ưu BPR cho kỹ thuật phân rã ma trận Matrix Factorization – MF) như trình bày trong (Rendle et al., 2009) để cài đặt giải thuật gợi ý tích hợp vào hệ thống. Bài viết sẽ giới thiệu những khái niệm cơ bản về MF và BPR-MF nhằm ứng dụng chúng vào xây dựng hệ thống đề xuất, bạn đọc quan tâm có thể tham khảo chi tiết về các kỹ thuật này trong (Rendle et al., 2009; Koren et al., 2009; Thai-Nghe et al., 2012).

Kỹ thuật phân rã ma trận (Matrix Factorization - MF)

Kỹ thuật phân rã ma trận 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, nghĩa là $\mathbf{X} \sim \mathbf{W}\mathbf{H}^T$, được minh họa như trong Hình 4.



Hình 4: Minh họa kỹ thuật phân rã ma trận

Trong đó, $\mathbf{W} \in \mathbb{R}^{U \times K}$ là một ma trận mà mỗi dòng là một véc tơ bao gồm K nhân tố tiềm ẩn (latent factors) mô tả người dùng u ; và $\mathbf{H} \in \mathbb{R}^{I \times K}$ là một ma trận mà mỗi dòng là một véc tơ bao gồm K nhân tố tiềm ẩn mô tả cho item i (thông thường $K \ll U$ và $K \ll I$). Lưu ý, trong Hình 4, \mathbf{H}^T là ma trận chuyển vị (transpose) của \mathbf{H} .

Vấn đề then chốt của kỹ thuật MF là làm sao để tìm được giá trị của hai tham số (ma trận) \mathbf{W} và \mathbf{H} . Hai tham số này được xác định bằng cách tối ưu hóa hàm mục tiêu (objective function). Trong vấn đề dự đoán xếp hạng (rating prediction) thì hàm mục tiêu thông dụng nhất là Root Mean Squared Error (RMSE), được biểu diễn như sau:

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

Trong đó, D^{train} là tập dữ liệu huấn luyện, λ là

hệ số chính tắc hóa (regularization) nhằm làm giảm học vẹt (over-fitting). Giả sử sau quá trình tối ưu, ta nhận được giá trị của \mathbf{W} và \mathbf{H} , khi đó, xếp hạng của user u cho item i được dự đoán bằng công thức:

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

Ứng dụng kỹ thuật BPR-MF vào hệ thống gợi ý bài hát

Trong hệ thống gợi ý bài hát, vì dữ liệu phản hồi từ người dùng là tiềm ẩn và thuộc về vấn đề gợi ý mục tin (item recommendation) nên chúng tôi đề xuất sử dụng giải thuật BPR-MF do tính ưu việt của nó đối với loại tính chất của dữ liệu này. Hàm mục tiêu của BPR-MF như sau:

$$O^{BPR-MF} = \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{r}_{uij}) - \lambda_\theta \|\theta\|^2 \quad (4)$$

Với $\sigma(x) = \frac{1}{1 + e^{-x}}$ là một hàm sigmoid;

$\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ và \hat{r}_{ui} , \hat{r}_{uj} là giá trị dự đoán của người u cho bài hát i và bài hát j ;

θ đại diện cho các tham số \mathbf{W} và \mathbf{H} ;

và $\lambda_\theta \|\theta\|^2$ là thành phần chuẩn hóa để ngăn chặn vấn đề học vẹt (overfitting).

Sử dụng stochastic gradient descent (Bottou, 2004) để triển khai chi tiết giải thuật học trong (Rendle et al., 2009), ta được giải thuật như bên dưới:

- 1: procedure LearnBPR-MF ($D_S, \lambda_w, \lambda_{H^+}, \lambda_{H^-}, \alpha$) {
- 2: $W \leftarrow N(0, \sigma^2)$ // khởi tạo giá trị cho tham số
- 3: $H \leftarrow N(0, \sigma^2)$ // khởi tạo giá trị cho tham số
- 4: repeat // cập nhật lại giá trị tham số đến khi tối ưu
- 5: draw randomly (u, i, j) from D_S ($j \in I \setminus I_u^+$)
- 6: $\hat{r}_{uij} \leftarrow \langle w_{uk}, h_{ik} - h_{jk} \rangle$
- 7: $v \leftarrow \frac{e^{-\hat{r}_{uij}}}{1 + e^{-\hat{r}_{uij}}}$
- 8: $w_{uk} \leftarrow w_{uk} + \alpha(v(h_{ik} - h_{jk}) - \lambda_w w_{uk})$
- $h_{ik} \leftarrow h_{ik} + \alpha(v w_{uk} - \lambda_{H^+} h_{ik})$

9:

$$10: h_{jk} \leftarrow h_{jk} + \alpha(v(-w_{uk}) - \lambda_H - h_{jk})$$

11: until convergence

12: return (W, H)

13: end procedure

Trong giải thuật trên, đầu vào là các siêu tham số (hyper-parameter, cách xác định các siêu tham số này sẽ được mô tả trong phần V.B.) và tập dữ liệu huấn luyện. Tương tự như những hệ thống gợi ý khác, tập dữ liệu đầu vào có 3 thông tin là: id của người dùng, id của bài hát, và tỷ lệ thời lượng nghe như trình bày trong công thức (1). Mục tiêu chính của giải thuật này là để tìm ra hai tham số W và H sao cho tối ưu nhất (tương tự giải thuật phân rã ma trận (matrix factorization (Koren *et al.*, 2009) minh họa trong Hình 4). Trước hết, chúng được khởi tạo bằng các giá trị ngẫu nhiên theo phân phối chuẩn (normal distribution) $N(0, \sigma^2)$ như ở dòng 2, 3; Sau đó, qua từng bước lặp, giá trị của 2 tham số này được cập nhật lại đến khi hàm mục tiêu (trình bày trong công thức 4) đạt giá trị tối ưu. Ở mỗi bước lặp, cặp bài hát i, j được chọn (với $j \in I \setminus I_u^+$) để xây dựng quan hệ “thích hơn” như biểu diễn trong Hình 3 (bạn đọc quan tâm xin tham khảo chi tiết trong tài liệu (Steffen *et al.*, 2009)). Sau quá trình tối ưu ta nhận được hai tham số W và H . Khi đó, tỷ lệ thời gian nghe của người dùng u trên bài hát i (mà u chưa nghe) được dự đoán bằng công thức (3).

5 XÂY DỰNG HỆ THỐNG

Giới thiệu hệ thống

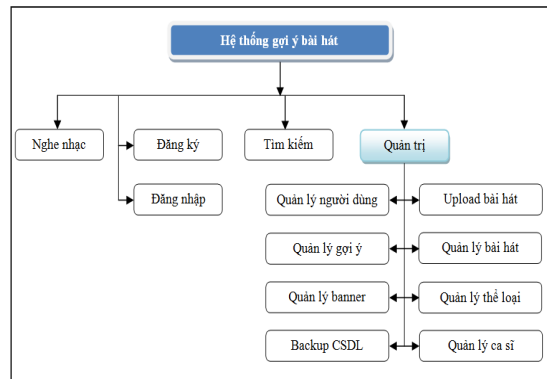
Hệ thống gợi ý bài hát được xây dựng dưới dạng một website cung cấp thông tin về âm nhạc giúp người dùng có thể chọn nghe các bài nhạc mà mình yêu thích. Khi người dùng truy cập vào website, có thể tìm kiếm, nghe và tải những bài nhạc về máy của mình. Đồng thời, hệ thống còn phân loại các bài nhạc theo từng thể loại, nhằm mang đến sự tiện lợi cho người sử dụng và cung cấp thông tin chi tiết về các bài nhạc như: tên bài hát, tên ca sĩ thể hiện,...

Với bất kỳ hệ thống gợi ý nào, nó phải có chức năng thu thập những phản hồi từ người dùng. Thông thường hệ thống ghi nhận sự phản hồi của người dùng dưới hình thức ghi nhận một giá trị xếp hạng cụ thể (thích (1) / không thích (0), hay từ ★ đến ★★★★★) gọi là phản hồi tường minh (explicit feedback). Tuy nhiên, với cách này thì hệ

thống thường khó có thể ghi nhận được nhiều phản hồi từ người dùng do đa phần người dùng ngại khi phải cung cấp thông tin phản hồi như thế và đôi khi làm cho người dùng không hài lòng (đồng nghĩa với khả năng mất khách hàng). Do đó, để tạo sự tiện lợi cho người dùng và hệ thống có thể thu thập được phản hồi một cách dễ dàng, trong hệ thống gợi ý bài hát, chúng tôi sẽ ghi nhận các phản hồi của người dùng dưới dạng phản hồi tiềm ẩn (implicit feedback). Hệ thống sẽ tự động ghi nhận lại giá trị phản hồi được tính bằng *thời gian nghe/tổng thời lượng bài nhạc*. Giá trị này trong khoảng từ 0 đến 1 sẽ được hệ thống tự động lưu vào cơ sở dữ liệu (CSDL) dùng để huấn luyện mô hình và thực hiện chức năng gợi ý.

Sơ đồ chức năng của hệ thống

Sơ đồ chức năng chính của hệ thống được trình bày trong Hình 5.



Hình 5: Sơ đồ chức năng chính của hệ thống

Nghe nhạc: Chức năng này sẽ được kích hoạt khi người dùng nhấn vào 1 bài nhạc nào đó, một player (JW Player) sẽ thực hiện play bài nhạc. Ngoài ra, trên player có hỗ trợ thêm các chức năng khác như: dừng (stop), tạm dừng (pause),...

Tìm kiếm: Chức năng cho phép người dùng tìm kiếm các bài nhạc theo từ khóa là tên các bài nhạc.

Đăng ký: Cho phép người dùng thực hiện đăng ký tạo 1 tài khoản trên hệ thống. Người dùng phải cung cấp các thông tin như: tên truy cập, mật khẩu, họ và tên, email,... và hệ thống sẽ lưu vào CSDL.

Đăng nhập: Người dùng thực hiện việc đăng nhập khi đã tạo tài khoản trên hệ thống. Khi đăng nhập, người dùng có thể thay đổi thông tin tài khoản, đổi mật khẩu. Ngoài ra, khi người dùng đã đăng nhập và thực hiện nghe các bài nhạc thì hệ thống sẽ tự động ghi nhận lại giá trị phản hồi (trọng số xếp hạng) vào CSDL. Đồng thời thực hiện chức năng gợi ý các bài nhạc mới phù hợp đến người dùng.

Upload bài hát: Cho phép người quản trị thực hiện thêm các bài nhạc mới vào hệ thống.

Quản lý bài hát: Cho phép người quản trị thực hiện thêm, sửa, xóa các bài nhạc.

Quản lý thể loại: Cho phép người quản trị thực hiện thêm, sửa, xóa các thể loại nhạc.

Quản lý ca sĩ: Cho phép người quản trị thực hiện thêm, sửa, xóa các ca sĩ thể hiện.

Quản lý người dùng: Cho phép người quản trị thực hiện xem và xóa các tài khoản người dùng.

Quản lý gợi ý: Người quản trị thực hiện việc xây dựng lại mô hình để thực hiện chức năng gợi ý.

Quản lý banner: Cho phép người quản trị thực hiện thêm, sửa và xóa banner quảng cáo (.gif, .jpg, .png, .swf) theo từng vị trí.

Backup CSDL: Cho phép người quản trị thực hiện lưu lại CSDL hiện thời của hệ thống

Chức năng quan trọng nhất của hệ thống là chức năng gợi ý bài hát bằng giải thuật xếp hạng cá

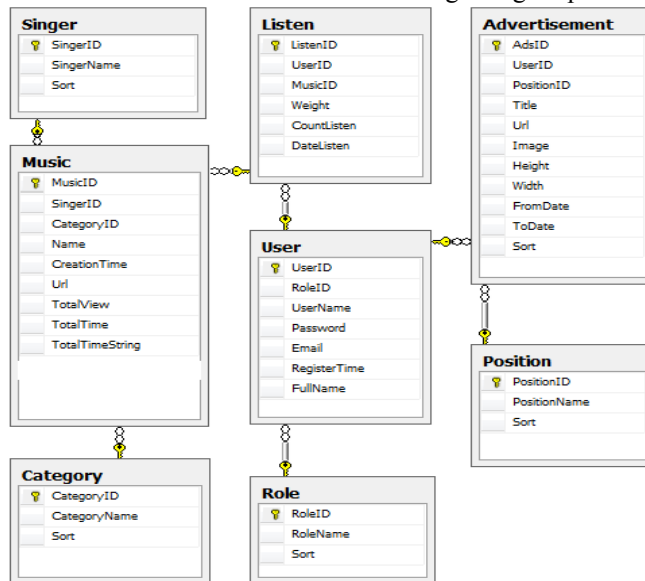
nhân BPR-MF để gợi ý các bài nhạc mới phù hợp cho từng người nghe. Những bài nhạc gợi ý này được thể hiện dưới dạng một danh sách xếp theo thứ tự ưu tiên giảm dần theo giá trị dự đoán nhằm giúp cho người dùng có thể tìm nghe những bài nhạc mà mình yêu thích một cách nhanh nhất.

Ngoài ra, hệ thống cũng có thêm các mục chức năng khác như: gợi ý TOP các bài nhạc được nhiều người nghe, các bài nhạc cùng thể loại với bài nhạc đang nghe và TOP các bài nhạc người dùng đã nghe.

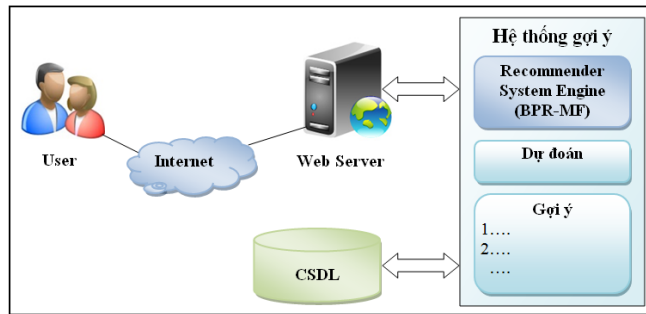
Lược đồ CSDL quan hệ

Lược đồ CSDL để lưu trữ thông tin trong hệ thống được minh họa như ở Hình 6, trong đó quan trọng nhất là người dùng (thông tin về user), bài hát (thông tin về item), và tỷ lệ thời lượng nghe (đại diện cho rating).

Hệ thống gợi ý bài hát được phát triển trên môi trường .NET sử dụng ngôn ngữ ASP.NET, C# và Hệ quản trị cơ sở dữ liệu SQL Server. Kiến trúc tổng thể của hệ thống được trình bày trong Hình 7. Ở đó, hệ thống gồm 2 thành phần chính: thành phần tương tác với người dùng và thành phần gợi ý. Thành phần tương tác với người dùng qua giao diện Web, nó có nhiệm vụ thể hiện các bài nhạc, cung cấp các chức năng đến người dùng như: đăng ký, nghe, tìm kiếm, phân loại,... đồng thời, ghi nhận lại thông tin phản hồi từ người dùng và gợi ý những bài nhạc mới thích hợp đến họ. Thành phần gợi ý có nhiệm vụ dự đoán sở thích của người dùng từ những thông tin phản hồi trong quá khứ của họ.



Hình 6: Lược đồ CSDL



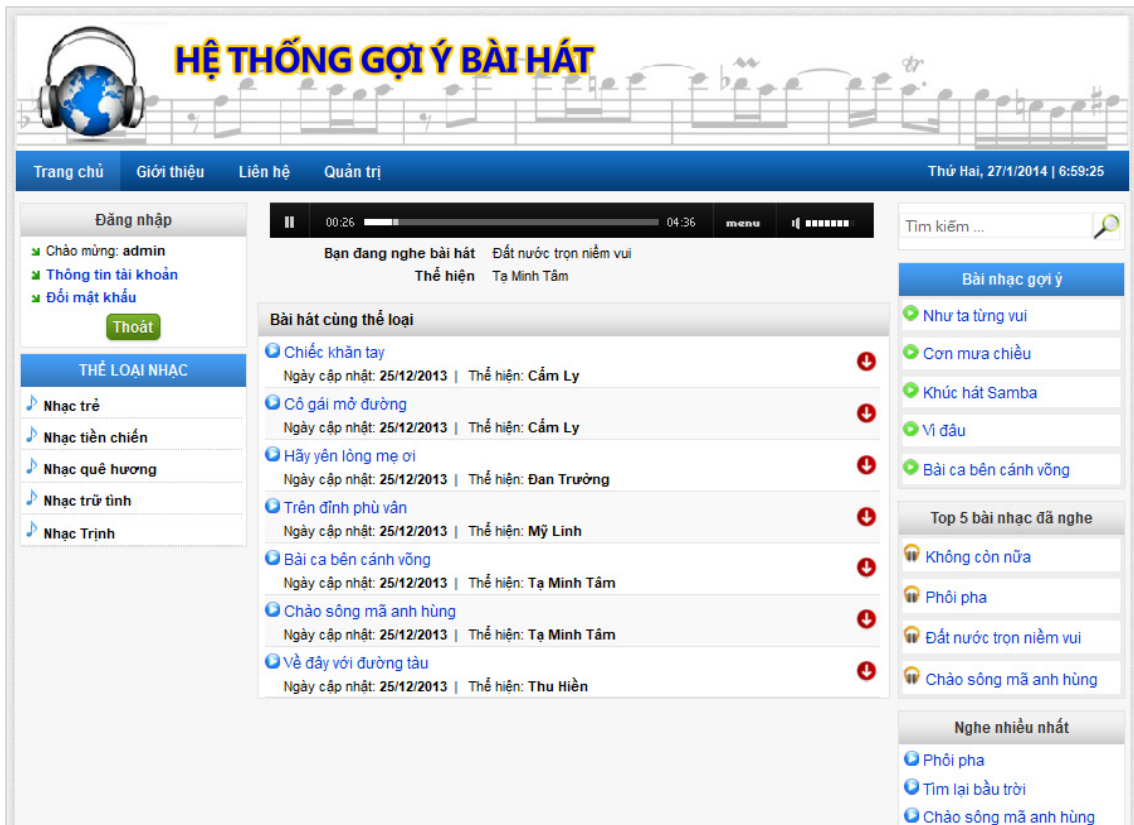
Hình 7: Kiến trúc hệ thống gợi ý bài hát

6 KẾT QUẢ MINH HỌA

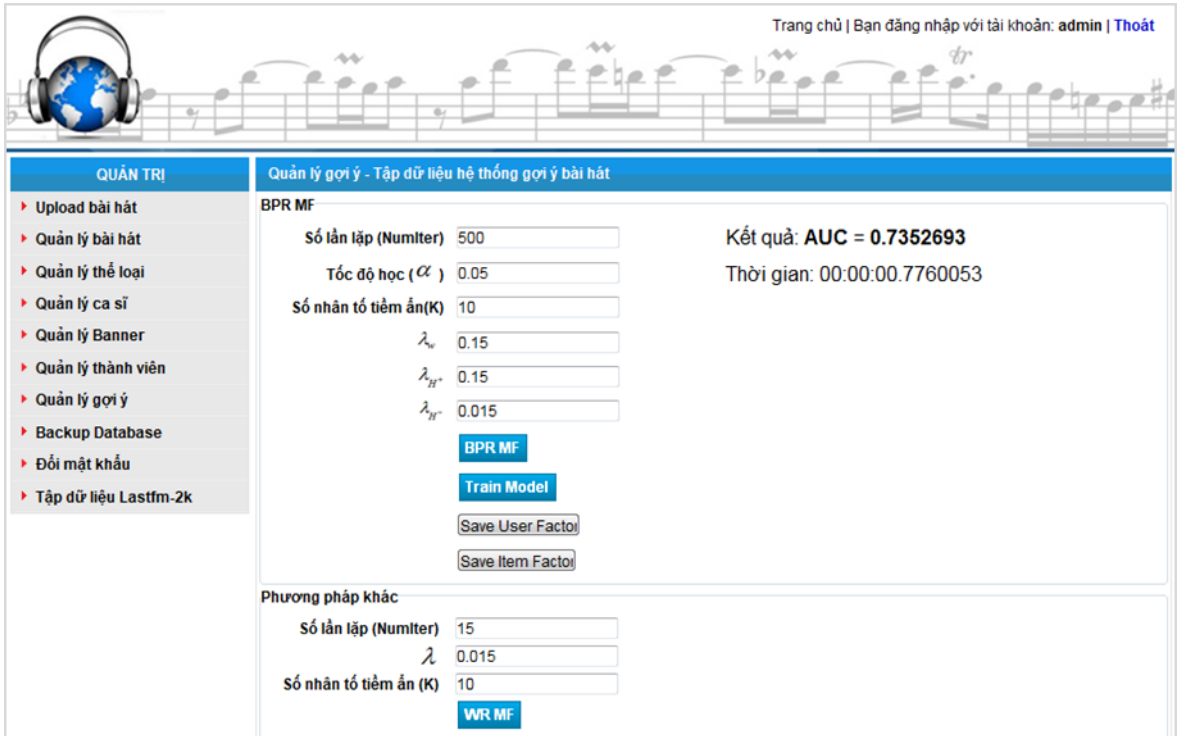
a. Một số giao diện chính của hệ thống gợi ý bài hát

Sau khi người dùng đã đăng nhập vào hệ thống, hệ thống sẽ gợi ý Top-N những bài nhạc được dự đoán là phù hợp nhất đến người dùng theo giải

thuật đã trình bày. Bên cạnh đó, hệ thống cũng gợi ý cho người dùng dựa trên những phương pháp khác như: Bài hát cùng thể loại, cùng ca sĩ, top các bài hát được nghe nhiều nhất, danh sách các bài người dùng đã nghe trong quá khứ, như được minh họa trong Hình 8.



Hình 8: Gợi ý bài hát cho người dùng theo nhiều cách



Hình 9: Quản trị chức năng gợi ý và những chức năng khác

Bên cạnh đó, hệ thống cũng cung cấp các module quản trị khác như mô tả ban đầu, đặc biệt là chức năng quản trị mô hình gợi ý của hệ thống nhằm giúp admin có thể tự huấn luyện và cập nhật lại mô hình sau một thời gian sử dụng, như minh họa trong **Error! Reference source not found.**

b. Đánh giá hệ thống

Dữ liệu:

Chúng tôi thu thập dữ liệu từ thực tế gồm 109 người dùng (user), 50 bài hát (item) và 1109 thông tin phản hồi (rating).

Phương pháp đánh giá:

Mặc dù, có nhiều phương pháp khác nhau có thể sử dụng để đánh giá giải thuật như: F-Measure, AUC (Area Under the ROC Curve), Root mean square error (RMSE),... Tuy nhiên, mỗi phương pháp đánh giá sẽ thích hợp cho từng lĩnh vực cụ thể, chẳng hạn như: F-Measure và AUC thường được dùng trong truy tìm thông tin và phân lớp (classification) và ranking, còn root mean square error (RMSE) được dùng trong dự đoán xếp hạng (rating prediction). Ở đây chúng tôi sử dụng độ đo AUC để đánh giá hiệu quả của mô hình do nó phù hợp với các hệ thống dạng ranking.

Sử dụng nghi thức kiểm tra (leave one out). Đối với mỗi người dùng thực hiện bỏ ngẫu nhiên một dòng dữ liệu đã nghe (u, i, r_{ui}) từ tập dữ liệu thu thập được I_u^+ (tập các mục người dùng u đã nghe). Dòng dữ liệu bỏ này sẽ được đưa vào tập dữ liệu kiểm tra S_{test} và phần còn lại đưa vào tập dữ liệu huấn luyện S_{train} . Dùng tập dữ liệu huấn luyện S_{train} để xây dựng mô hình học và dùng tập dữ liệu kiểm tra S_{test} để đánh giá mô hình bằng độ đo AUC.

Tìm kiếm siêu tham số (hyper-parameter search):

Siêu tham số là các tham số đầu vào của giải thuật, các tham số này cần phải được xác định trước khi chạy giải thuật. Do không gian giá trị của các siêu tham số này là khá lớn, ứng với mỗi tập dữ liệu khác nhau thì sẽ có các siêu tham số cho kết quả tốt tương ứng. Do đó, các siêu tham số cần phải được tìm kiếm sao cho phù hợp với từng tập dữ liệu, để giải thuật cho kết quả tốt nhất có thể. Phương pháp được sử dụng trong luận văn này là phương pháp GridSearch được dùng để tìm kiếm các siêu tham số (Thai-Nghe *et al.*, 2010). Trong giải thuật BPR-MF ta có các siêu tham số cần tìm là:

K: số nhân tố tiềm ẩn

Iter: số lần lặp

α : tốc độ học

λ_w : tham số chuẩn hóa cho nhân tố người dùng (user factor)

λ_{H^+} : tham số chuẩn hóa cho nhân tố bài hát đã nghe (positive item factor)

λ_{H^-} : tham số chuẩn hóa cho nhân tố bài hát chưa nghe (negative item factor)

Do không gian giá trị các tham số là rất lớn, nên phương pháp GridSearch thực hiện qua hai giai đoạn: tìm thô (Coarse Search) và tìm mịn (Granularity Search).

Tìm thô: Thực hiện tìm giá trị các siêu tham số như: K, NumIter, α , λ_w , λ_{H^+} , λ_{H^-} bằng một heuristic nào đó. Ví dụ, $\alpha \in [0..1]$ nên ta sẽ tìm trên 4 giá trị (0.2, 0.4, 0.6, 0.8).

Tìm mịn: Thực hiện tìm mịn (quanh lân cận các giá trị) tham số sau khi đã tìm thô, cuối cùng ta sẽ chọn ra được các tham số cho kết quả tốt nhất. Ví dụ với $\alpha=0.4$ mô hình sẽ cho kết quả tốt nhất giai đoạn tìm thô nên trong giai đoạn này ta sẽ tìm xung quanh giá trị 0.4 (như 0.3, 0.35, 0.45, 0.5).

Các giải thuật liên quan trong bài viết này đều được tìm các siêu tham số tốt nhất.

Kết quả thực nghiệm đánh giá bằng độ đo AUC:

Kết quả độ đo AUC trung bình từ 5 lần chạy của kỹ thuật BPR-MF được sử dụng trong hệ thống là 0.742 trong khi chỉ gợi ý dựa trên các bài hát được người dùng nghe nhiều nhất bằng (Most Popular - Ricci *et al.*, 2011) là 0.616. Như vậy, kết quả này cải thiện hơn rất nhiều so với những hệ thống gợi ý mà chỉ chọn những bài được nghe/tìm nhiều nhất để gợi ý cho người dùng. Việc thu thập mức độ hài lòng của người dùng khi sử dụng hệ thống này so với các hệ thống khác sẽ được thực hiện trong tương lai.

7 KẾT LUẬN

Chúng tôi giới thiệu một giải pháp xây dựng Hệ thống gợi ý bài hát dựa vào phản hồi tiềm ẩn (implicit feedback) từ người dùng để gợi ý những bài hát mới mà họ có thể thích nghe. Chúng tôi xây dựng hệ thống hoàn chỉnh và tích hợp giải thuật gợi ý vào hệ thống, thu thập phản hồi từ người dùng và đánh giá hiệu quả của hệ thống dựa trên những phản hồi đó. Thực nghiệm cho thấy giải

pháp này có thể dùng để tích hợp vào các hệ thống gợi ý âm nhạc hiện có trên thị trường.

Hướng nghiên cứu tiếp theo chúng tôi sẽ cải tiến thuật toán bằng cách kết hợp với phương pháp phân rã ma trận đa quan hệ (Multi-relational Matrix Factorization) nhằm tận dụng tất cả các thông tin có sẵn như bài hát, ca sĩ, thể loại,... cũng như những thông tin tiềm ẩn khác (như số lần tìm trên mỗi bài hát, số lần click/download,...)

TÀI LIỆU THAM KHẢO

1. Asela Gunawardana and Guy Shani, 2009. "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks".
2. B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, 2002. "Incremental singular value decomposition algorithms for highly scalable recommender systems". In Proceedings of the 5th International Conference in Computers and Information Technology.
3. D. Billsus, and M. Pazzani, 1998. "Learning collaborative information filters". In International Conference on Machine Learning, Morgan Kaufmann Publishers.
4. D. Goldberg, D. Nichols, B.M. Oki, D. Terry, 1992. "Using Collaborative Filtering to Weave an Information Tapestry", Communications of the Association of Computer Machinery, 35(12):61-70.
5. F. Ricci, L. Rokach, B. Shapira and P. B. Kantor, 2011. "Recommender Systems Handbook", Springer.
6. J. S. Breese, D. Heckerman, and C. Kadie, 1998. "Empirical analysis of predictive algorithms for collaborative filtering". In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI.
7. L. Bottou, 2004, "Stochastic learning". In O. Bousquet, and von U. Luxburg, editors, Advanced Lectures on Machine Learning, Lecture Notes in Artificial Intelligence, pages 146-168, Springer Verlag, Berlin.
8. R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang, 2008. "One-class collaborative filtering". In IEEE International Conference on Data Mining (ICDM 2008), pages 502 - 511.
9. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme, 2009.

- “BPR: Bayes Personalized Ranking from implicit feedback”. In Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence, AUAI Press.
10. Y. Hu, Y. Koren, and C. Volinsky, 2008. “Collaborative filtering for implicit feedback datasets”, in IEEE International Conference on Data Mining, pages 263-272.
 11. Y. Koren, R. Bell, C. Volinsky, 2009. “Matrix factorization techniques for recommender systems”, IEEE Computer Society Press 42(8), pages 30-37.
 12. Nguyễn Thái Nghe, 2012. Kỹ thuật phân rã ma trận trong xây dựng hệ thống gợi ý. Kỹ yếu Hội thảo Công nghệ thông tin, trang 68-77, Tạp chí khoa học Trường Đại học Đà Lạt.
 13. Nguyen Thai-Nghe, Lucas Drumond, Tomáš Horváth, Artus Krohn-Grimberghe, Alexandros Nanopoulos, Lars Schmidt-Thieme, 2012. Factorization Techniques for Predicting Student Performance, in Book Educational Recommender Systems and Technologies: Practices and Challenges (ERSAT 2012). Santos, O. C. and Boticario, J. G. (Eds.). pp. 129-153. IGI Global
 14. Nguyễn Thái Nghe, Huỳnh Xuân Hiệp (2012): Ứng dụng kỹ thuật phân rã ma trận đa quan hệ trong xây dựng hệ trợ giảng thông minh, Kỹ yếu Hội thảo quốc gia lần thứ XV: Một số vấn đề chọn lọc của CNTT&TT, trang 470-477. Nhà xuất bản Khoa học và Kỹ thuật.
 15. Nguyen Thai-Nghe, Zeno Gantner, Lars Schmidt-Thieme (2010): Cost-Sensitive Learning Methods for Imbalanced Data, in proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2010), IEEE Xplore.
 16. MyMediaLite Recommender System Library <http://mymedialite.net>.
 17. <http://ir.ii.uam.es/hetrec2011/datasets.html>