Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jsi.2017.013

PHÂN RÃ MA TRẬN VỚI YẾU TỐ THỜI GIAN TRONG HỆ THỐNG GỢI Ý

Lê Ngọc Quyền, Nguyễn Hữu Hòa và Nguyễn Thái Nghe
Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ

Thông tin chung:
Ngày nhận bài: 15/09/2017
Ngày nhận bài sửa: 10/10/2017
Ngày duyệt đăng: 20/10/2017

Title:
Matrix and tensor factorization with temporal effect in recommender systems

Từ khóa:
Hệ thống gợi ý, làm trơn hàm mũ, phân rã ma trận, phân rã nhân tử

Keywords:
Exponential smoothing, matrix factorization, recommender systems, tensor factorization

ABSTRACT

This paper proposes the construction of a recommender system to predict users' preferences based on matrix factorization techniques. Because of the changes of users' preferences time by time, to achieve more accurate result, exponential smoothing is integrated into the matrix factorization model by utilizing tensor factorization. This usage aims at exploiting and taking advantage of information about the time and the order of users' giving feedbacks. The model is tested relied on the datasets in suggestion and evaluation using the root mean squared error. The experimental results demonstrate fairly good performance of the proposed method.

TÓM TẮT

Bài viết này đề xuất một giải pháp dự đoán sở thích của người dùng dựa trên kỹ thuật phân rã ma trận (Matrix Factorization – MF) có tích hợp yếu tố thời gian trong hệ thống gợi ý (Recommender Systems – RS). Do sở thích của người dùng có thể thay đổi theo thời gian, để kết quả gợi ý có độ chính xác cao hơn chúng tôi đề xuất tích hợp phương pháp dự báo san bằng hàm mũ (Exponential Smoothing - ES) vào mô hình Tensor Factorization với mục tiêu khai thác và tận dụng được các thông tin về thời gian cũng như trình tự (sequence) mà người dùng đã đưa ra phản hồi. Thực nghiệm ban đầu trên các tập dữ liệu chuẩn trong lĩnh vực gợi ý và đánh giá bằng độ đo RMSE (Root Mean Squared Error) đã cho thấy hướng tiếp cận này cho kết quả rất khả quan.

Trích dẫn: Lê Ngọc Quyền, Nguyễn Hữu Hòa và Nguyễn Thái Nghe, 2017. Phân rã ma trận với yếu tố thời gian trong hệ thống gợi ý. Tạp chí Khoa học Trường Đại học Cần Thơ. Số chuyên đề: Công nghệ thông tin: 96-102.

1 GIỚI THIỆU

Hiện nay, hệ thống gợi ý (Recommender Systems - RS) đã được ứng dụng rất rộng rãi trong nhiều lĩnh vực khác nhau, đặc biệt là thương mại điện tử (e-commerce). Từ khi ra đời RS là một giải pháp hữu ích giúp giải quyết vấn đề quá tải thông tin và giúp đưa ra những gợi ý phù hợp với từng người dùng. Để đạt được kết quả cao thì mỗi hệ thống gợi ý cần có một mô hình gợi ý có thể tận dụng và khai thác tốt được dữ liệu đã thu thập để đưa ra các gợi ý phù hợp cho từng người dùng, do đó việc lựa chọn thuật toán cho mô hình gợi ý là yếu tố quan trọng nhất để xây dựng một RS thành công.

Trong hệ thống gợi ý người ta quan tâm đến 3 đối tượng: **người dùng (user)**, **mục tin (item)** và các phản hồi của người dùng đối với mục tin, thường là các **xếp hạng (rating)**. Hiện tại, các nhà nghiên cứu đã đề xuất rất nhiều thuật toán để xây dựng hệ thống gợi ý, tuy nhiên chúng thường được chia thành ba nhóm lớn (Ricci *et al*., 2011; Su and Khoshgoftaar., 2009).

– Lọc dựa trên nội dung (Content-based Filtering): Người dùng sẽ được gợi ý các item tương tự như các item mà người dùng đã ưa thích trước đó dựa trên thuộc tính của item.

– Lọc cộng tác (Collaborative Filtering): Đưa ra gợi ý bằng cách dựa trên sự tương tự giữa những người dùng hoặc giữa những sản phẩm trong hệ thống.

– Giải thuật kết hợp: Đưa ra gợi ý dựa vào việc kết hợp cả nhóm giải thuật ở trên.

Trong nhóm giải thuật lọc cộng tác thì kỹ thuật phân rã ma trận (Matrix factorization - MF) là một trong những phương pháp thành công nhất hiện nay (state-of-the-art) trong lĩnh vực dự đoán xếp hạng của RS (Bell and Koren, 2007; Koren *et al*., 2009) .Tuy nhiên, đa số các giải thuật thuộc nhóm MF chỉ dựa vào sự tương quan giữa user và item để đưa ra dự đoán *(User × Item → Rating)* mà không quan tâm đến yếu tố thời gian khi xây dựng mô hình gợi ý.

Nói cách khác, các nhóm giải thuật MF đa số tập trung vào giới thiệu các mục tin phù hợp với người dùng dựa vào tất cả các dữ liệu trong quá khứ của người dùng đó, mà không xem xét đến yếu tố sở thích của người dùng có thể thay đổi theo thời gian. Khi đó, một số đánh giá khá lâu trước đây sẽ không còn phù hợp với sở thích hiện tại của người dùng. Chẳng hạn, khách hàng có xu hướng thích những sản phẩm mới đưa ra thị trường gần đây hơn là những sản phẩm cũ, mặc dù trong quá khứ sản phẩm đó có thể được nhiều người ưa chuộng nhưng đối với thời điểm hiện tại nó không còn phù hợp.

Trong RS yếu tố thời gian có thể được khai thác theo 2 cách:

– Thời gian tuyệt đối (Concrete time): đại diện cho các điểm thời gian cụ thể, như được sử dụng trong tài liệu (Dunlavy *et al*., 2011). Dạng thời gian này thường được sử dụng trong các hệ thống gợi ý theo ngữ cảnh. Ví dụ: khoảng thời gian trong ngày, ngày trong tuần, tháng hoặc mùa trong năm, ... (Adomavicius *et al*., 2011; Gantner *et al*., 2010).

– Thời gian tương đối (Relative time): mô tả chuỗi dữ liệu có thứ tự (order). Ví dụ: trình tự giải quyết một bài tập trong hệ thống giảng dạy, danh sách các sản phẩm được sắp xếp theo độ yêu thích tăng dần của người dùng... Loại thời gian này thường được sử dụng trong kỹ thuật dự báo hoặc trong mô phỏng dữ liệu tuần tự (Rendle *et al*., 2010; Bengio, 1996).

Trong bài báo này, chúng tôi quan tâm đến thứ tự người dùng đưa ra phản hồi trên sản phẩm, vì vậy yếu tố thời gian sẽ được khai thác theo cách thứ hai, tức là thời gian tương đối. Với cách khai thác đó, thì yếu tố thời gian trong mô hình dự đoán được xác định bằng cách sắp xếp các đánh giá của người dùng theo thứ tự từ cũ đến mới, sau đó áp dụng các phương pháp dự đoán chuỗi thời gian vào tập dữ liệu đã sắp xếp. Như vậy, với bất kỳ tập dữ liệu nào chỉ cần biết được thời gian mà người dùng đưa ra phản hồi thì đều có thể áp dụng mô hình mà chúng tôi đề xuất.

Để tích hợp được yếu tố thời gian vào mô hình MF, chúng ta cần mở rộng chiều của ma trận hiện có *(User × Item × Time → Rating)*. Như đã đề cập ở phần trên, để xử lý được yếu tố thời gian trong mô hình gợi ý cần chọn một phương pháp dự đoán chuỗi thời gian (time series) phù hợp với mô hình dữ liệu.

Hiện nay, có rất nhiều phương pháp dự đoán dựa trên việc phân tích chuỗi thời gian như (Box *et al*., 2015):

– Phương pháp trung bình đơn (Simple Moving Average): Tính trung bình cộng của một dãy số để dự đoán số liệu trong tương lai, trong đó giá trị của các giai đoạn trước đều có trọng số như nhau.

– Phương pháp trung bình có trọng số (Weighted Moving Average): Tương tự như phương pháp trung bình đơn nhưng có gán trọng số cho dữ liệu.

– Phương pháp làm trơn (san bằng) hàm mũ (Exponential Smoothing): Đây là phương pháp dự đoán dựa trên dữ liệu gần nhất cộng với phần trăm chênh lệch giữa số dự đoán và số thực tế ở thời điểm dự đoán. Là phương pháp được sử dụng phổ biến nhất trong tất cả các phương pháp dự đoán.

Do sở thích người dùng có thể thay đổi theo thời gian, những phản hồi mà người dùng đánh giá trước đây hiện tại có thể không còn phù hợp, như vậy dữ liệu dự đoán phải được gán trọng số để cho thấy mức độ quan trọng của dữ liệu ở mỗi thời điểm là khác nhau. Thêm vào đó, sở thích của mỗi người dùng thay đổi không giống nhau, ví dụ: có một số người thích những sản phẩm mới, nhưng lại có một số người thích những sản phẩm quen thuộc, đã từng sử dụng trước đó. Vì thế, việc gán trọng số phải được thực hiện một cách linh hoạt, phù hợp với từng người dùng. Trong các phương pháp dự đoán chuỗi thời gian được đề cập ở trên, phương pháp làm trơn hàm mũ (Exponential Smoothing - ES) là phương pháp phù hợp nhất với nhu cầu đặt ra. Thông qua việc xác định tham số làm trơn mũ ($0 < \alpha < 1$), chúng ta hoàn toàn có thể gán trọng số một cách linh hoạt cho dữ liệu trong mô hình dự đoán (xem chi tiết tại mục 2.3). Hơn nữa, đây là

một phương pháp dự báo nhanh, tương đối đơn giản nhưng độ chính xác khá cao, dễ dàng tích hợp vào mô hình gợi ý.

Từ nhận định trên, chúng tôi đề xuất tích hợp phương pháp dự báo làm trơn (san bằng) hàm mũ (Exponential Smoothing - ES) vào mô hình MF thông qua kỹ thuật phân tích nhân tố tiềm ẩn (Tensor Factorization – TF) với mục tiêu khai thác và tận dụng được các thông tin về thời gian cũng như trình tự người dùng đưa ra phản hồi. Mô hình sẽ được thực nghiệm trên các tập dữ liệu chuẩn trong lĩnh vực gợi ý và đánh giá bằng độ đo RMSE (Root Mean Squared Error) để cho thấy hướng tiếp cận đã đề xuất cho kết quả rất khả quan.

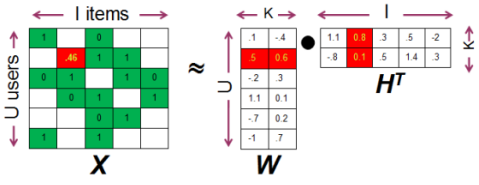
Phần còn lại của bài báo này có cấu trúc như sau: Phần 2 giới thiệu các kỹ thuật được sử dụng để xây dựng mô hình gợi ý. Phần 3 giới thiệu mô hình dự đoán do chúng tôi đề xuất. Phần 4 là kết quả thực nghiệm trên tập dữ liệu mẫu. Cuối cùng phần 5 là phần kết luận.

2 CÁC KỸ THUẬT ĐƯỢC SỬ DỤNG ĐỂ XÂY DỰNG MÔ HÌNH

Trước tiên chúng tôi tóm tắt ngắn gọn kỹ thuật phân rã ma trận (MF) (Koren *et al*., 2009), kỹ thuật phân tích nhân tử (Tensor Factorization - TF) (Kolda and Bader, 2009) (Dunlavy *et al*., 2011b) và phương pháp dự đoán làm trơn hàm mũ (ES) (Yorucu, 2003) (Ostertagová and Ostertag, 2012) để làm cơ sở cho việc đề xuất mô hình phân rã ma trận tích hợp yếu tố thời gian.

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

Kỹ thuật phân rã ma trận (MF) là việc chia một ma trận lớn **X** thành 2 ma trận có kích thước nhỏ hơn rất nhiều so với ma trận ban đầu **W** và **H**, sao cho X có thể được xây dựng lại từ hai ma trận nhỏ hơn này càng chính xác càng tốt (Koren *et al*., 2009), nghĩa là $\mathbf{X} \approx \mathbf{W}\boldsymbol{H}^{\boldsymbol{T}}$ như minh hoạ trong Hình 1.



Hình 1: Minh họa kỹ thuật phân rã ma trận

Trong đó, **X** là tập hợp tất cả các đánh giá (rating) của người dùng (user) với mục tin (item), bao gồm cả những giá trị chưa biết cần được dự đoán tạo nên một ma trận gọi là Utility Matrix. $\mathbf{W} \in \mathbb{R}^{\boldsymbol{U}\times\boldsymbol{K}}$ là một ma trận mà mỗi dòng *u* là một véc tơ bao gồm *K* nhân tố tiềm ẩn (latent factors) mô tả cho user *u*, và $\mathbf{H} \in \mathbb{R}^{\boldsymbol{I}\times\boldsymbol{K}}$ là một ma trận mà mỗi dòng *i* là một véc tơ bao gồm K nhân tố tiềm ẩn mô tả cho item *i*.

Gọi $\boldsymbol{w}_{uk}$ và $\boldsymbol{h}_{ik}$ là các phần tử tương ứng của hai ma trận **W** và **H**, khi đó rating r của user u trên item i được dự đoán bởi công thức:

$$\hat{r}_{ui} = \sum_{k=1}^{K} \mathbf{w}_{uk}\boldsymbol{h}_{ik} = (\boldsymbol{W}\boldsymbol{H}^{\boldsymbol{T}})_{u,i} \quad (1)$$

Như vậy, vấn đề chủ chốt của MF là làm sao tìm được ma trận **W** và **H**. Hai tham số này có thể được xác định bằng cách tối ưu hóa hàm mục tiêu (objective function) (3) theo RMSE (root mean squared error) như sau:

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

$$O^{MF} = \sum_{(u,i)\in \boldsymbol{D}^{\boldsymbol{train}}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\|W\|_F^2 + \|H\|_F^2) \quad (3)$$

Với λ là hệ số chính tắc hóa ($0 \le \lambda < 1$) để tránh tình trạng quá khớp hay còn gọi là học vẹt (overfitting – xảy ra khi mô hình dự đoán cho kết quả tốt trên tập huấn luyện nhưng cho kết quả thấp trên tập thử nghiệm) (Feng *et al*., 2009) và $\|\cdot\|_F^2$ là chuẩn Frobenius (Böttcher and Wenzel, 2008).

Một trong những kỹ thuật để tối ưu hóa hàm mục tiêu là dùng SGD (Stochastic gradient descent) (Koren, 2010), tức là các tham số $\boldsymbol{w}_{uk}$ và $\boldsymbol{h}_{ik}$ sẽ được cập nhật theo công thức:

$$\mathbf{w}_{uk}^{new} = \boldsymbol{w}_{uk}^{old} - \beta\left(\frac{\partial O^{MF}}{\partial \boldsymbol{w}_{uk}^{old}}\right) \quad (4)$$

$$= \boldsymbol{w}_{uk}^{old} + 2\beta(r_{ui} - \hat{r}_{ui})h_{ik}$$

$$\mathbf{h}_{ik}^{new} = \boldsymbol{h}_{ik}^{old} - \beta\left(\frac{\partial O^{MF}}{\partial\, \boldsymbol{h}_{ik}^{old}}\right) \quad (5)$$

$$= \boldsymbol{h}_{ik}^{old} + 2\beta(r_{ui} - \hat{r}_{ui})w_{uk}$$

Với β là tốc độ học (learning rate, $0 < \beta < 1$). Quá trình cập nhật sẽ thực hiện đến khi đạt độ lỗi chấp nhận được hoặc lặp lại đến số lần lặp quy định trước.

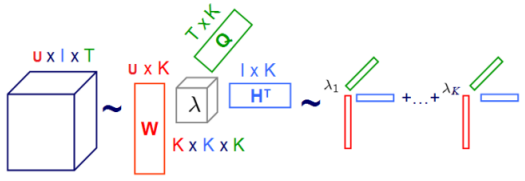
2.2 Phân rã ma trận ba chiều (Tensor Factorization – TF)

Tensor Factorization (TF) (Kolda and Bader, 2009; Dunlavy *et al*., 2011) là một dạng tổng quát của kỹ thuật phân tích ma trận. Cho một tensor **Z** với kích thước $\boldsymbol{U} \times \boldsymbol{I} \times \boldsymbol{T}$, với 2 thành phần đầu tiên *U* và *I* lần lượt thể hiện cho ma trận nhân tử *user* và *item* trong kỹ thuật phân rã ma trận (MF) được đề cập ở phần trước. Thành phần thứ 3 thể hiện cho

ngữ cảnh dự đoán (thời gian, địa điểm...) có kích thước T. Như vậy, Z có thể được viết lại như sau:

$$\mathbf{Z} \approx \sum_{k=1}^{K} \boldsymbol{w}_k \circ \boldsymbol{h}_k \circ \boldsymbol{q}k \qquad (6)$$

Trong đó, mỗi véc tơ $\boldsymbol{w}_k \in \mathbb{R}^U$, $\boldsymbol{h}_k \in \mathbb{R}^I$ và $\boldsymbol{q}_k \in \mathbb{R}^T$ lần lượt thể hiện các nhân tố tiềm ẩn (latent factors) của user, item và time. Các tham số mô hình cũng được tối ưu hóa theo RMSE (root mean squared error) bằng cách sử dụng SGD (Stochastic gradient descent). Một minh hoạ của TF được trình bày trong Hình 2.



Hình 2: Minh họa kỹ thuật phân rã nhân tử

2.3 Phương pháp làm trơn hàm mũ (Exponential Smoothing - ES)

Phương pháp làm trơn (san bằng) hàm mũ (ES) (Yorucu, 2003) là một phương pháp được sử dụng rộng rãi trong các bài toán dự báo chuỗi thời gian (forecast time series) (Box *et al*., 2015). ES sử dụng các số liệu quá khứ của chuỗi thời gian để tạo ra một hàm số mũ xấp xỉ tương đối thích ứng với chuỗi số liệu đó, và từ đó có thể sử dụng hàm này để dự báo cho các đại lượng kinh tế cho tương lai. Theo phương pháp này, giá trị xu thế tại thời điểm t là một trung bình có trọng số của tất cả các giá trị sẵn có trước đó, nơi mà các trọng số giảm dần về mặt hình học khi người ta quay ngược trở lại theo thời gian.

Hiện nay, có rất nhiều phương pháp dự đoán làm trơn hàm mũ như: làm trơn hàm mũ đơn (simple exponential smoothing - SES), phương pháp dự báo Brown, phương pháp dự báo Holt, phương pháp san bằng mũ Damped Trend. Trong phạm vi bài báo này, chúng tôi sử dụng phương pháp làm trơn hàm số mũ đơn (SES) để xây dựng mô hình dự đoán, vì SES dùng cho dữ liệu ổn định, không xu hướng và không có tính mùa (Ostertagová and Ostertag, 2012). Hàm này được biểu diễn bởi công thức như sau:

$$\boldsymbol{S_t} = \alpha y_{t-1} + (1-\alpha) S_{t-1}\,, \quad t \geq 2 \qquad (7)$$

Một cách tổng quát ta có:

$$\boldsymbol{S}_t = \alpha \sum_{i=1}^{t-2} (1-\alpha)^{i-1} y_{t-i} + (1-\alpha)^{i-2} S_2\,, \quad t \geq 2 \qquad (8)$$

Trong đó, $\boldsymbol{S}_t$ là giá trị dự đoán tại thời điểm t, $\boldsymbol{y_t}$ là giá trị thực tế tại thời điểm t và $\boldsymbol{\alpha}$ là hằng số làm trơn mũ có giá trị từ 0 đến 1. Hệ số α trong mô hình dự báo thể hiện tầm quan trọng hay mức độ ảnh hưởng của số liệu hiện tại đến đại lượng dự báo. Nếu α được chọn càng lớn thì trọng số của các dữ liệu cũ càng nhỏ và ngược lại, nếu α càng nhỏ thì dữ liệu cũ sẽ có trọng số lớn hơn. Để chọn α phải dựa vào việc phân tích biến động của hiện tượng và những kinh nghiệm nghiên cứu đã qua. Giá trị α tốt nhất là giá trị làm cho tổng bình phương sai số dự đoán nhỏ nhất (Ostertagová and Ostertag, 2012).

3 MÔ HÌNH ĐỀ XUẤT

Trong mô hình phân rã ma trận MF đã trình bày ở phần 2.1, gợi ý sở thích của người dùng được đưa ra dựa vào thông tin từ các ma trận *user* và *item* và không quan tâm đến thông tin thời điểm mà người dùng đưa ra đánh giá. Tuy nhiên, trong thực tế sở thích của người dùng thường có xu hướng thay đổi dần theo thời gian, chẳng hạn 3 năm trước người dùng thích chiếc xe máy Honda Airblade thì 3 năm sau họ có thể thích chiếc ô tô Honda City (do điều kiện kinh tế, gia đình,.. thay đổi theo thời gian). Mặc dù vậy, một số sở thích khá lâu trước đây có thể đã không còn phù hợp hơn so với các sở thích gần đây và hiện tại của người dùng, vì thế chúng tôi đã đề xuất tích hợp kỹ thuật san bằng hàm mũ đơn vào kỹ thuật MF/TF để giải quyết vấn đề ảnh hưởng của yếu tố thời gian trong RS.

Để có thể giải quyết vấn đề vừa nêu, thay vì chỉ sử dụng thông tin từ các ma trận *user* và *item* như công thức (1), chúng tôi tích hợp thêm thông tin từ các ma trận có liên quan đến yếu tố thời gian khi người dùng đưa ra dự đoán. Chúng tôi cũng sử dụng các giá trị thiên vị (biases) cho mô hình dự đoán để tránh tình trạng thiên vị trong quá trình đưa ra đánh giá (Koren *et al*., 2009).

Đối với đề xuất trên số lượng tham số của mô hình dự đoán sẽ khác với MF, chúng tôi gọi hướng tiếp cận mới này là TFES (Tensor factoriztion - Exponential Smoothing). Khi đó, công thức dự đoán sẽ trở thành:

$$\hat{r}_{ui} = \mu + b_u + b_i + \sum_{k=1}^{K} w_{uk}\, h_{ik}\, \Phi_{tk} \qquad (9)$$

$$\Phi_{tk} = \sum_{t=2}^{L} q_{tk}\, s_{tk}$$

Trong đó,

μ : là giá trị rating trung bình của tất cả các user và các item trong tập dữ liệu huấn luyện

$$\mu = \frac{\sum_{(u,i,r \in D^{train})} r}{|D^{train}|}$$

b_u : là giá trị thiên vị *user*

$$b_u = \frac{\sum_{(u',i,r \in D^{train})|u'=u|}(r-\mu)}{|\{(u',i,r \in D^{train})|u'=u|\}|}$$

b_i : là giá trị thiên vị *item*

$$b_i = \frac{\sum_{(u,i',r \in D^{train})|i'=i|}(r-\mu)}{|\{(u,i',r \in D^{train})|i'=i|\}|}$$

s_{tk} : là hàm làm trơn hàm mũ

$$s_{tk} = \alpha r_{t-1} + (1-\alpha)S_{(t-1)k}, \quad t \geq 2$$

q_{tk} : là véc tơ tiềm ẩn đại diện cho thời gian

L : là độ dài số giao dịch trong quá khứ sử dụng cho mô hình dự đoán

Với mô hình đề đã để xuất ở trên, hàm mục tiêu của mô hình dự đoán trở thành

$$O^{TFES} = \sum_{(u,i,t) \in \boldsymbol{D^{train}}}(r_{ui} - \hat{r}_{ui})^2 + \lambda(\|W\|_F^2 + \|H\|_F^2 + \|Q\|_F^2 + b_u^2 + b_u^2) \quad (10)$$

Trong đó $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{Q}$ lần lượt là các ma trận nhân tố tiềm ẩn đại diện cho *user*, *item*, và *time.* λ là hệ số chính tắc hóa ($0 \leq \lambda < 1$) tương tự như MF. Hàm mục tiêu (10) vẫn được tối ưu bằng phương pháp SGD, tức là các tham số *w, h, q* tương ứng sẽ được cập nhật theo công thức:

$$\mathbf{w}_{uk}^{new} = \boldsymbol{w}_{uk}^{old} - \beta\left(\frac{\partial O^{TFES}}{\partial \boldsymbol{w}_{uk}^{old}}\right) \quad (11)$$

$$\boldsymbol{h}_{ik}^{new} = \boldsymbol{h}_{ik}^{old} - \beta\left(\frac{\partial O^{TFES}}{\partial \boldsymbol{h}_{ik}^{old}}\right) \quad (12)$$

$$\boldsymbol{q}_{tk}^{new} = q_{tk}^{old} - \beta\left(\frac{\partial O^{TFES}}{\partial \boldsymbol{q}_{tk}^{old}}\right) \quad (13)$$

Với β là tốc độ học (learning rate, $0 < \beta < 1$).

Giá trị của $\frac{\partial O^{TFES}}{\partial w_{uk}}$, $\frac{\partial O^{TFES}}{\partial h_{ik}}$ và $\frac{\partial O^{TFES}}{\partial q_{tk}}$ được xác định bởi công thức:

$$\frac{\partial O^{TFES}}{\partial w_{uk}} = -2(r_{ui} - \hat{r}_{ui})h_{ik}\Phi_{tk} + \lambda w_{uk} \quad (14)$$

$$\frac{\partial O^{TFES}}{\partial h_{ik}} = -2(r_{ui} - \hat{r}_{ui})w_{uk}\Phi_{tk} + \lambda h_{ik} \quad (15)$$

$$\frac{\partial O^{TFES}}{\partial q_{tk}} = -2(r_{ui} - \hat{r}_{ui})w_{uk}h_{ik}(\sum_{t=2}^{L} s_{tk}) + \lambda q_{tk} \quad (16)$$

Sau quá trình tối ưu, ta nhận được các tham số W, H, Q. Khi đó, chúng ta có thể dự đoán kết quả xếp hạng cho *user u* trên *item i* thông qua công thức (9).

4 KẾT QUẢ THỰC NGHIỆM

4.1 Dữ liệu

Để thực nghiệm mô hình đề xuất ở trên chúng tôi sử dụng các tập dữ liệu từ hai lĩnh vực khác nhau là trong giải trí và trong giáo dục.

Cụ thể, tập dữ liệu Movielens 100k được công bố năm 1998 bởi nhóm GroupLens. Tập dữ liệu này có 100.000 đánh giá được thực hiện bởi 943 người dùng trên số lượng 1.682 phim, mỗi người dùng có đánh giá ít nhất 20 phim và đánh giá được gán 1 (tệ) đến 5 (tuyệt vời)...

Tập dữ liệu Assistments (2009-2010) trích từ hệ thống Assistments (Feng et al., 2009), tập dữ liệu này có nguồn gốc từ hệ thống trợ giảng thông minh, kết quả đạt được từ các lần sinh viên giải quyết các bài tập, câu hỏi sẽ được dùng để dự đoán khả năng thực hiện của sinh viên khi có một yêu cầu mới. Tập dữ liệu Algebra (2009 - 20010) có các thuộc tính tương tự tập Assistments và được công bố từ KDD Cup 2010 (Bennett *et al*., 2007). Hai tập dữ liệu này có thể được ánh xạ tương ứng qua các khái niệm trong RS như: sinh viên → user; công việc → item; và kết quả → rating. Thông tin của 3 tập dữ liệu trên được mô tả cụ thể trong Bảng 1.

4.2 Kết quả thực nghiệm

Để kết quả thực nghiệm được khách quan, các tập dữ liệu dùng trong thực nghiệm sẽ được phân chia theo phương pháp Splitting (Kohavi, 1995), chọn ngẫu nhiên 70% số phần tử của tập dữ liệu dùng làm tập học và 30% còn lại dùng làm tập kiểm tra.

Bảng 1: Thông tin về dữ liệu sử dụng trong thực nghiệm

Tập dữ liệu	Số user	Số item	Số rating
Movielens 100k	943	1,682	100,000
Assistments (2009 – 2010)	8,519	35,798	1,011,079
Algebra (2009 – 2010)	3,310	1,422,200	8,918,054

Các siêu tham số (hyper-parameters) trong mô hình dự đoán như số lần lặp (Iter), số nhân tố tiềm ẩn K, tốc độ học β, hệ số chính tắc hóa λ và hằng số làm trơn mũ α được xác định bằng phương pháp tìm kiếm siêu tham số (hyper-parameter search) (Cen *et al*., 2006).

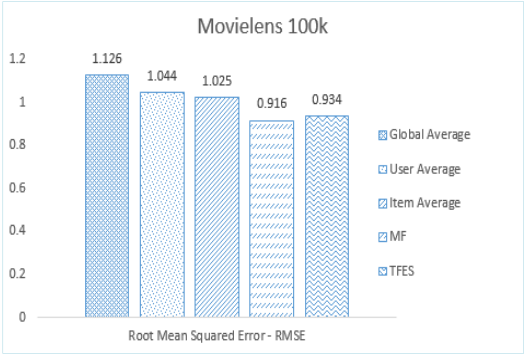
Tuy nhiên, do việc tìm kiếm bằng vét cạn sẽ mất nhiều thời gian nên đề tài chỉ thực hiện việc tìm kiếm thô cho các phương pháp này. Ví dụ: Iter ∈ (50, 100, . . . , 1000), K ∈ (2^3, 2^4, . . . , 2^8), β ∈ (10^{-4}, 10^{-3}, 10^{-2}, $5*10^{-5}$, $5*10^{-4}$, $5*10^{-3}$), λ ∈ ($15*10^{-4}$, $15*10^{-3}$, $55*10^{-5}$, $55*10^{-4}$, $55*10^{-3}$), α ∈ (0.1, 0.2, ... , 0.9). Mỗi lần sẽ sử dụng một bộ siêu tham số (Iter, K, β, λ, α) để xây dựng mô hình trên tập huấn luyện và dự đoán cho tập kiểm tra, tính độ lỗi RMSE. Sau khi thử hết các bộ siêu tham số sẽ

lựa chọn bộ siêu tham số tốt nhất theo tiêu chí độ lỗi RMSE thấp nhất.

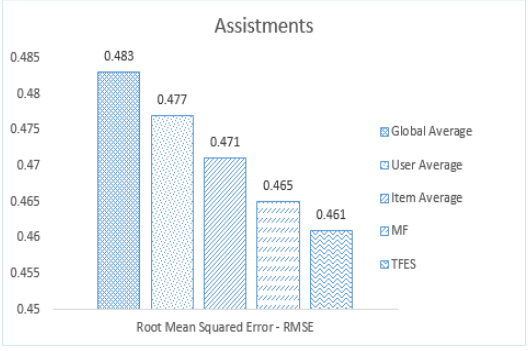
Nhằm mục đích kiểm chứng kết quả của giải thuật thì ngoài việc so sánh độ lỗi RMSE của mô hình TFES với MF, chúng tôi sẽ dùng thêm một số phương pháp baseline như (Su and Khoshgoftaar, 2009): Global average, User average và Item average (Sarwar *et al*., 2001; Nguyen Thai Nghe *et al*., 2010).

Dưới đây là kết quả thực nghiệm đánh giá bằng RMSE trên 3 tập dữ liệu đã nêu ở mục 4.1.

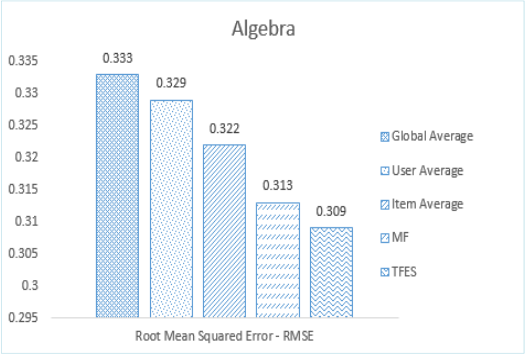
Kết quả thực nghiệm cho thấy TFES cho kết quả cao trên tập dữ liệu thuộc lĩnh vực giáo dục Assistments và Algebra. Tuy nhiên, với tập dữ liệu lĩnh vực giải trí Movielens thì TFES cho kết quả chưa như mong đợi. Điều đó cho thấy việc tích hợp phương pháp làm trơn hàm mũ (ES) vào mô hình gợi ý sẽ đạt kết quả cao trên tập dữ liệu có tính chất tuần tự, tích lũy dần theo thời gian. Vì thế, trong tương lai chúng tôi sẽ tiếp tục cải tiến mô hình để có thể khai thác tốt yếu tố thời gian và cho kết quả cao trên nhiều loại dữ liệu khác nhau.



Hình 3: Kết quả so sánh RMSE trên tập Movielens 100k



Hình 4: Kết quả so sánh RMSE trên tập Assistments



Hình 5: Kết quả so sánh RMSE trên tập Algebra

5 KẾT LUẬN

Trong bài viết này, chúng tôi đã giới thiệu một mô hình dự đoán sở thích của người dùng có tích hợp yếu tố thời gian. Đây là sự kết hợp giữa mô hình phân rã ma trận MF/TF với dự báo chuỗi thời gian Exponential Smoothing, nhằm tận dụng được yếu tố thời gian để đưa ra dự đoán phù hợp với sở thích người dùng.

Tuy nhiên, TFES vẫn còn hạn chế là thời gian huấn luyện mô hình khá chậm so với MF, nguyên nhân là do số lượng tham số mô hình cần tìm của TFES nhiều hơn dẫn đến quá trình tối ưu hóa hàm mục tiêu cũng mất nhiều thời gian hơn. Trong tương lai, chúng tôi sẽ tiếp tục tối ưu và thực nghiệm trên nhiều tập dữ liệu khác để củng cố thêm kết quả của phương pháp đề xuất và nghiên cứu giải pháp để cải thiện tốc độ huấn luyện mô hình cho TFES.

TÀI LIỆU THAM KHẢO

Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A., 2011. Context-Aware Recommender Systems. AI Mag. 32, 67–80. doi:10.1609/aimag.v32i3.2364

Bell, R.M., Koren, Y., 2007. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, in: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM '07. IEEE Computer Society, Washington, DC, USA, pp. 43–52. doi:10.1109/ICDM.2007.90

Bengio, Y., 1996. Markovian Models for Sequential Data, in: Neural Computing Surveys, Vol. 2, Pp. 129- 162, 1999.

Bennett, J., Elkan, C., Liu, B., Smyth, P., Tikk, D., 2007. KDD Cup and Workshop 2007. SIGKDD Explor Newsl 9, 51–52. doi:10.1145/1345448.1345459

Böttcher, A., Wenzel, D., 2008. The Frobenius norm and the commutator. Linear Algebra Its Appl. 429, 1864–1885. doi:10.1016/j.laa.2008.05.020

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Cen, H., Koedinger, K., Junker, B., 2006. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement, in: Intelligent Tutoring Systems, Lecture Notes in Computer Science. Presented at the International Conference on Intelligent Tutoring Systems, Springer, Berlin, Heidelberg, pp. 164–175. doi:10.1007/11774303_17

Dunlavy, D.M., Kolda, T.G., Acar, E., 2011a. Temporal Link Prediction Using Matrix and Tensor Factorizations. ACM Trans. Knowl. Discov. Data 5, 1–27. doi:10.1145/1921632.1921636

Feng, M., Heffernan, N., Koedinger, K., 2009. Addressing the assessment challenge with an online system that tutors as it assesses. User Model. User-Adapt. Interact. 19, 243–266. doi:10.1007/s11257-009-9063-7

Gantner, Z., Rendle, S., Schmidt-Thieme, L., 2010. Factorization Models for Context-/Time-aware Movie Recommendations, in: Proceedings of the Workshop on Context-Aware Movie Recommendation, CAMRa '10. ACM, New York, NY, USA, pp. 14–19. doi:10.1145/1869652.1869654

Kohavi, R., 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1137–1143.

Kolda, T., Bader, B., 2009. Tensor Decompositions and Applications. SIAM Rev. 51, 455–500. doi:10.1137/07070111X

Koren, Y., 2010. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. ACM Trans Knowl Discov Data 4, 1:1–1:24. doi:10.1145/1644873.1644874

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. Computer 42, 30–37. doi:10.1109/MC.2009.263

Ostertagová, E., Ostertag, O., 2012. Forecasting using simple exponential smoothing method. Acta Electrotech. Inform. 12. doi:10.2478/v10198-012-0034-2

Rendle, S., Freudenthaler, C., Schmidt-Thieme, L., 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10. ACM, New York, NY, USA, pp. 811–820. doi:10.1145/1772690.1772773

Ricci, F., Rokach, L., Shapira, B. & Kantor, P.B., eds. (2011)., n.d. Recommender Systems Handbook. Springer.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based Collaborative Filtering Recommendation Algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW '01. ACM, New York, NY, USA, pp. 285–295. doi:10.1145/371920.372071

Su, X., Khoshgoftaar, T.M., 2009. A Survey of Collaborative Filtering Techniques. Adv. Artif. Intell. doi:10.1155/2009/421425

Thai-Nghe, N., Gantner, Z., Schmidt-Thieme, L., 2010. Cost-sensitive learning methods for imbalanced data. IEEE, pp. 1–8. doi:10.1109/IJCNN.2010.5596486

Yorucu, V., 2003. The Analysis of Forecasting Performance by Using Time Series Data for Two Mediterranean Islands. Rev. Soc. Econ. Bus. Stud. 2.