DỰ ĐOÁN SỞ THÍCH CỦA NGƯỜI DÙNG THEO THỜI GIAN VỚI PHƯƠNG PHÁP TIMESVD++

Nguyen Manh Thang - 20195915
Instructor: Dr. Nguyen Thi Ngoc Anh

Hanoi, 3/2023

Outline

Outline

Nhu cầu sử dụng gợi ý sản phẩm

NETFLIX

Giới thiệu bài toán

- Sử dụng phương pháp phân rã ma trận TIMESVD++ để dự đoán sở thích người dùng
- So sánh kết quả mô hình sử dụng TIMESVD++ với các phương pháp phân rã ma trận tích hợp thời gian khác
- Đối tượng nghiên cứu: Tập dữ liệu đánh giá sản phẩm làm đẹp của Amazon
- Phạm vi nghiên cứu: Dữ liệu đánh giá từ năm 2009 đến năm 2014

Outline

Hệ thống gợi ý là gì?

Hệ thống gợi ý (Recommender Systems-RS) là hệ thống lọc thông tin, từ đó gợi ý thông tin cho người dùng cái mà có thể hữu ích đối với người dùng. Hệ thống thu nhập thông tin từ người dùng và sản phẩm $\rightarrow$ phân tích $\rightarrow$ đưa ra gợi ý. Ví dụ:

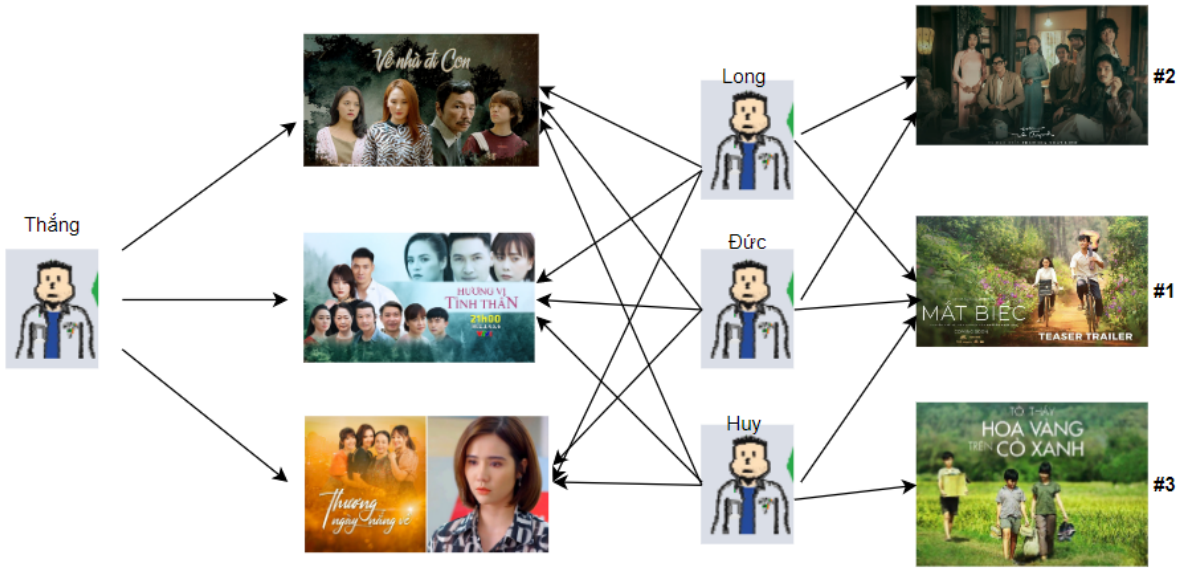
- Tăng trải nghiệm người dùng -> Tăng doanh thu của công ty

Lọc cộng tác

- Thuật ngữ được định nghĩa bởi Tapestry, hệ gợi ý đầu tiên trên thế giới [4]
- Phương pháp này phân tích mối liên quan giữa người dùng và các thuộc tính của sản phẩm để định nghĩa quan hệ giữa người dùng - sản phẩm
- Lọc cộng tác phân tích được nhiều khía cạnh của dữ liệu nhưng không hiệu quả khi gặp dữ liệu mới ít thống tin (cold start problem) [1]
- Được chia làm 2 cách tiếp cận: các phương pháp lân cận (the neighborhood methods) và các mô hình yếu tố tiềm ẩn (latent factor models)

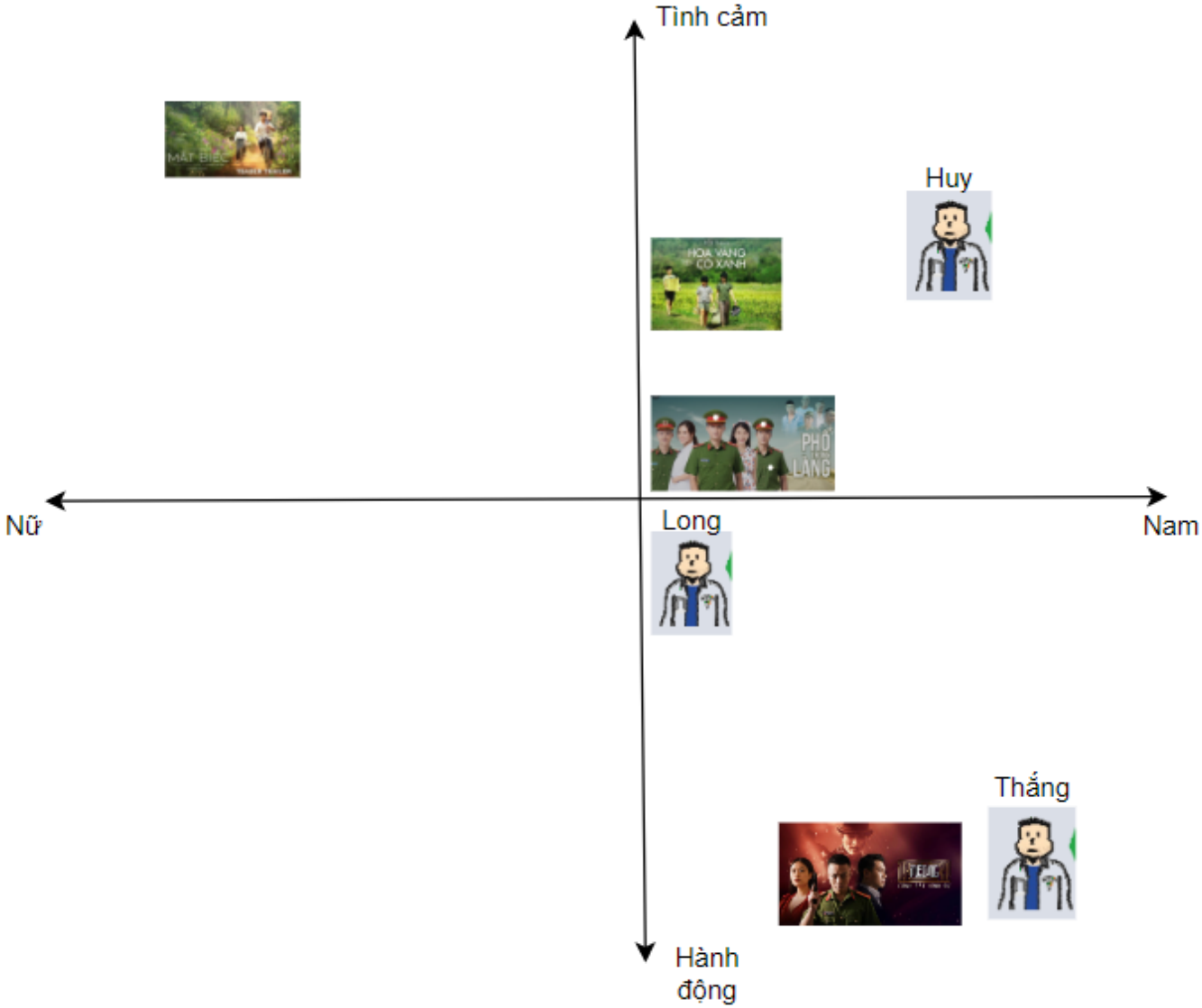
Phương pháp vùng lân cận

- Tập trung vào việc tính toán mối quan hệ giữa các sản phẩm hoặc giữa các người dùng
- Tiếp cận theo định hướng đánh giá sản phẩm dựa trên cách người dùng đánh giá của các "sản phẩm hàng xóm"
- Ưu điểm: Các thực hiện đơn giản,hiệu quả và đưa ra được các gợi ý chính xác và mang tính cá nhân hóa.
- Nhược điểm: Chưa mô tả được sự tương quan giữa người dùng và sản phẩm



Mô hình yếu tố tiềm ẩn

- Mô hình sẽ giải thích sự đánh giá bằng cách mô tả sản phẩm lẫn người dùng theo k yếu tố ẩn (không gian k chiều)
- Các yếu tố này có thể là sở thích, mức độ quan tâm, tuổi tác, giới tính, v.v. của người dùng hoặc các đặc tính của sản phẩm
- Ưu điểm: độ chính xác cao, khả năng áp dụng cho bộ dữ liệu lớn nhưng thưa
- Nhược điểm: mô hình phức tạp, nhiều tham số, có thể bị overfitting khi số lượng nhân tố quá lớn


Tình cảm
Nữ
Nam
Hành động
Huy
Long
Thắng

Sự thay đổi theo thời gian trong hệ thống gợi ý

- Một người có thể yêu thích dòng phim kinh dị nhưng 2 năm sau thì chuyển sang thẻ loại phim tình cảm
- Một bộ phim mới ra năm 2022 có thể là hay vào thời điểm ra mắt nhưng 2 năm sau thì có thể không được đánh giá cao nữa
- Một nười dùng khó tính chỉ đánh giá các bộ phim tối đa 3 sao nhưng sau này họ trở nên dễ tính hơn và bắt đầu đánh giá cao hơn 3 sao.
- Cứ đến ngày lễ Giáng Sinh, mọi người có xu hướng tìm những bộ phim về chủ đề Giáng Sinh nhiều hơn.
- Có nhiều các để xây dựng mô hình gợi ý có tích hợp thời gian như dùng trọng số thời gian (Time Weight Collaborative Filtering)[2], phân rã ma trận với yếu tố thời gian (TIMESVD++)

Outline

Bài toán của hệ thống gợi ý

- Hệ thống gợi ý gồm: người dùng (user), sản phẩm (item) và phản hồi (feedback)
- 3 thành phần được biểu diễn trên 1 ma trận User-item: mỗi hàng là 1 user, mỗi cột là một item, phản hồi của user lên item được ghi vào ô tương ứng.

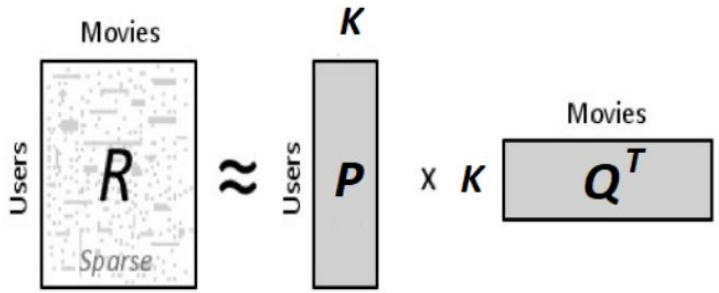
		Items					
		1	*2*	...	*i*	...	*m*
Users	*1*	5	3		1	2	
	2		2				4
	:			5			
	u	3	4		2	1	
	:					4	
	n			3	2		

- U là tập n người dùng (user), u là 1 người dùng cụ thể ($u \in U$)
- I là tập m người đung (item), i là 1 sản phẩm cụ thể ($i \in I$)
- R là tập các giá trị phản hồi của người dùng, $r_{ui} \in R$ là phản hồi của người dùng u cho sản phẩm i

$\rightarrow$ Tìm hàm $\hat{r}_{ui} = U \times I$ để xấp xỉ với hàm r(u,i) với sai số càng nhỏ càng tốt.

Phân rã ma trận-Matrix factorization

Chia một ma trận R lớn thành 2 ma trận P và Q có kích thước nhỏ hơn nhiều so với ma trận R sao cho $R \approx P \times Q^T$



- K là số nhân tố tiềm ẩn của user và item
- $P \in R^{n \times K}$ ma trận user, mỗi dòng là một user có K nhân tố ẩn
- $Q \in R^{m \times K}$ ma trận item, mỗi dòng là một item có K nhân tố ẩn

Xếp hạng của người dùng u cho sản phẩm i được dự đoán theo công thức:

$$\hat{r}_{ui} = \sum_{k=1}^{K} p_{uk} q_{ik} \tag{3.1}$$

Thêm độ thiên lệch

- Có những sự sai lệch trong dự đoán chỉ liên quan đến người dùng hoặc sản phẩm. Ví dụ:Có user dễ và khó tính, cũng có những item được rated cao hơn những items khác chỉ vì user thấy các users khác đã đánh giá item đó cao rồi $\rightarrow$ thiên lệch hay Bias.
- Cách dự đoán này còn được gọi là dự đoán cơ sở (Baseline Predictors)
- Độ lệch đánh giá của người dùng u cho sản phẩm i sẽ là b_{ui} như sau:

$$b_{ui} = \mu + b_i + b_u \tag{3.2}$$

  μ đánh gia trung bình, b_u và b_i độ thiên lệch của người dùng và sản phẩm
- Công thức (3.1) trở thành:

$$\hat{r}_{ui} = \mu + b_i + b_u + \sum_{k=1}^{K} p_{uk} q_{ik} \tag{3.3}$$

Thêm các đánh giá ẩn

- hệ thống RS có thể dùng những feedback ẩn(số lần click chuột,số lần bấm xem,lịch sử mua hàng,...) để học được hành vi của khách hàng.
- Thêm 1 vector nhân tố ẩn $y_i \in R^K$ cho mỗi item i, dùng để mô tả từng user qua những item mà user đó đánh giá.
- R(u) sẽ là tập bao gồm các item được đánh giá bởi user u cụ thể.
- một user u sẽ được mô hình hóa bằng công thức:

$$p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j$$

- Do đó hàm dự đoán sẽ là:

$$\hat{r}_{ui} = \mu + b_i + b_u + (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j) q_i^T \qquad (3.4)$$

Để tối ưu các tham số liên quan như p_u, q_i, b_i, b_u, y_i em sẽ tối thiểu hóa hàm mất mát sau:

$$L = \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\sum_{u=1}^{m} ||p_u||^2 + \sum_{j=1}^{m} ||y_j||^2 + \sum_{i=1}^{n} ||q_i||^2 + \sum_{u=1}^{m} b_u^2 + \sum_{i=1}^{n} b_i^2) \tag{3.5}$$

- $\mathcal{K}$ là tập các chỉ số của các đánh giá quan sát được
- λ là các tham số chính tắt hóa.
- m là số người dùng trong hệ thống, n là số sản phẩm trong hệ thống.

Thời gian thay đổi dự đoán cơ sở

- 1 bộ phim có thể trở nên nổi tiếng và có thể được yêu thích hơn trong 1 khoảng thời gian, do đó b_i sẽ được coi là 1 hàm thay đổi theo thời gian [3]
- 1 người dùng hay đánh giá các bộ phim trung bình 4 sao, nhưng giờ người đấy đánh giá bộ phim đó chỉ 3 sao do họ đổi cách đánh giá của mình, do đó b_u là hàm thay đổi theo thời gian.

Công thức dự đoán cơ sở cho người dùng u đánh giá sản phẩm i ở ngày t là:

$$b_{ui}(t) = \mu + b_u(t) + b_i(t) \tag{3.6}$$

Thời gian thay đổi độ lệch sản phẩm

- Sẽ dễ dàng đánh giá được sự thay đổi của b_i bằng cách chia nhỏ toàn bộ dòng thời gian liệu thành các khoảng thời gian nhỏ (timebased bins) và sử dụng một hằng số độ lệch cho sản phẩm cho mỗi khoảng thời gian.[3]
- Một ngày t sẽ được liên kết với 1 số nguyên Bin(t) (1 < Bin(t) < T, T là số khoảng thời gian được chia)
- Hàm tính độ lệch của item được cập nhật như sau:

$$b_i(t) = b_i + b_{i,Bin(t)} \tag{3.7}$$

Thời gian thay đổi độ lệch người dùng

- Ở phía users, sẽ cần một giải pháp khác để xác định sự thay đổi ngắn hạn của user. Mặt khác, cùng không đủ đánh giá của từng người để chia thành những khoảng thời gian như với items.[3]
- Ở phía user sẽ có 2 sự thay đổi: Sự thay đổi dài hạn(gradual drift) và sự thay đổi ngắn hạn (sudden drifts).
- dùng độ lệch thời gian để bắt những thay đổi dần dần của độ lệch user:

$$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^{\beta}$$

  Với t_u: là ngày xếp hạng trung bình của user u . β là tham số dùng để xác định việc giảm trọng số xếp hạng.
- Để bắt những thay đổi ngắn hạn (ngày) e dùng một tham số b_{ut}.

Hàm tính độ lệch của user được cập nhật như sau:

$$b_u(t) = b_u + \alpha_u \cdot dev_u(t) + b_{ut} \tag{3.8}$$

α_u: tham số đánh giá độ lệch thời gian cho user u

Công thức (3.6) sẽ trở thành:

$$b_{ui}(t) = \mu + b_u + \alpha_u dev_u(t) + b_{u,t} + b_i + b_{i,Bin(t)} \tag{3.9}$$

Tuy nhiên độ lệch của sản phẩm không hoàn toàn độc lập so với người dùng vì mỗi người dùng sẽ có các thang đánh giá khác nhau $\rightarrow$ một hàm đo phụ thuộc thời gian giữa độ lệch item với user $c_u(t)$.

$$b_{ui}(t) = \mu + b_u + \alpha_u dev_u(t) + b_{u,t} + (b_i + b_{i,Bin(t)}) \cdot c_u(t) \tag{3.10}$$

Với $c_u(t) = c_u + c_{u,t}$

- $c_{u,t}$ đại diện cho tính biến thiên theo ngày.
- c_u là phần không thay đổi theo thời gian.

Thời gian thay đổi nhân tố ẩn

- Sở thích người dùng thay đổi qua từng thời điểm, 1 người là fan của dòng phim kinh dị có thể sẽ trở thành fan của phim trinh thám 1 thời gian sau đó.
- người dùng thay đổi nhận thức của họ về một số diễn viên và đạo diễ, có thể từ thích thành ghét. Do đó, p_u là một hàm thay đổi theo thời gian.
- bởi vì có nhiều nhân tố ẩn nên phải đánh giá sự thay đổi theo thời gian của từng nhân tố và mô hình hóa tương tự như với độ lệch user.

$$p_u(t)[k] = p_{uk} + \alpha_{uk} \cdot dev_u(t) + p_{ukt} \quad k = 1, ..., K \tag{3.11}$$

Hàm dự đoán sau khi thêm các tham số thay đổi theo thời gian sẽ là:

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + \left(p_u(t) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right) q_i^T \tag{3.12}$$

Dó đó hàm mất mát của bài toán là:

$$L = \sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t))^2 + \lambda(\sum_{u=1}^{m} ||p_u||^2 + \sum_{j\in R(u)} ||y_j||^2 + \sum_{i=1}^{n} ||q_i||^2$$
$$+ \sum_{u=1}^{m} b_u^2 + \sum_{i=1}^{n} b_i^2 + \sum_{i=1}^{n} \sum_{bin(t)=1}^{T} b_{i,Bin(t)}{}^2 + \sum_{u=1}^{m} (\alpha_u^2 + c_u^2)$$
$$+ \sum_{u=1}^{m} \sum_{t\in\mathcal{K}} c_{u,t}{}^2 + \sum_{u=1}^{m} \sum_{k=1}^{K} (\alpha_{uk}{}^2) + \sum_{u=1}^{m} \sum_{t\in\mathcal{K}} \sum_{k=1}^{K} p_{ukt}{}^2) \quad (3.13)$$

Với m là số user, n là số item, T là số khoảng chia thời gian, K là số chiều của nhân tố ẩn , R(u) là tập bao gồm các item được đánh giá bởi user u và $\mathcal{K}$ là tập training.

Thuật toán tối ưu ADAM

Algorithm 1 Thuật toán tối ưu ADAM [1]

$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.99, \eta = 10^{-8}$ (Tham số mặc định)

$m_0 \leftarrow 0$ // Khởi tạo vecto moment thứ nhất

$v_0 \leftarrow 0$ // Khởi tạo vecto moment thứ hai

$i \leftarrow 0$ // Khởi tạo tham số đếm

while Θ_i chưa hội tụ **do**

$i \leftarrow i + 1$

$g_i = \nabla_\Theta f_i(\Theta_{i-1})$ // Là các công thức gradient của tham số đã viết ở trên

$m_i = \beta_1.m_{i-1} + (1 - \beta_1).g_i$ // Cập nhật cho vecto m

$v_i = \beta_2.v_{i-1} + (1 - \beta_2).g_i^2$ // Cập nhật cho vecto v

$\hat{m}_i \leftarrow m_i/(1 - \beta_1)$ // Tính bias-corrected cho vecto m

$\hat{v}_i \leftarrow v_i/(1 - \beta_2)$ // Tính bias-corrected cho vecto v

$\Theta_i \leftarrow \Theta_{i-1} - \eta.\hat{m}_i/(\sqrt{\hat{v}_i} + \varepsilon)$ // Cập nhật tham số cần tối ưu

end

$$\frac{\partial L}{\partial y_j} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot q_i \cdot |R(u)|^{-\frac{1}{2}} + \lambda \sum_{j\in R(u)} ||y_i|| \;\; \forall j \in R(u)$$

$$\frac{\partial L}{\partial \alpha_{uk}} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot p_u \cdot dev_u(t) + \lambda \sum_{u=1}^{m} \sum_{k=1}^{K} \alpha_{uk}$$

$$\frac{\partial L}{\partial p_{ukt}} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot p_u + \lambda \sum_{u=1}^{m} \sum_{k=1}^{K} \sum_{t\in\mathcal{K}} p_{ukt}$$

$$\frac{\partial L}{\partial b_u} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) + \lambda \sum_{u=1}^{m} b_u$$

$$\frac{\partial L}{\partial b_i} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) + \lambda \sum_{i=1}^{n} b_i$$

$$\frac{\partial L}{\partial b_{i,Bin(t)}} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot c_u(t) + \lambda \sum_{i=1}^{n} \sum_{Bin(t)=1}^{T} b_{i.Bin(t)}$$

$$\frac{\partial L}{\partial \alpha_u} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot dev_u(t) + \lambda \sum_{u=1}^{m} \alpha_u$$

$$\frac{\partial L}{\partial c_u} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}(t)) \cdot (b_i + \sum_{Bin(t)=1}^{T} b_{i,Bin(t)}) + \lambda \sum_{n=1}^{m} c_u$$

$$\frac{\partial L}{\partial c_{u,t}} = -\sum_{(u,i,t)\in\mathcal{K}} (r_{ui}(t) - \hat{r}_{ui}\alpha(t)) \cdot (b_i + \sum_{Bin(t)=1}^{T} b_{i,Bin(t)}) + \lambda \sum_{n=1}^{m} \sum_{t\in\mathcal{K}} c_{u,t}$$

Phương pháp đánh giá mô hình

Ở đây em sẽ dùng phương pháp bình phương tối thiểu(Mean square error - MSE) để đánh giá sự sai số của tập test.

$$MSE = \frac{\sum_{u,i,t \in \mathcal{K}_{test}} [r_{ui}(t) - \hat{r}_{ui}(t)]^2}{|\mathcal{K}_{test}|}$$

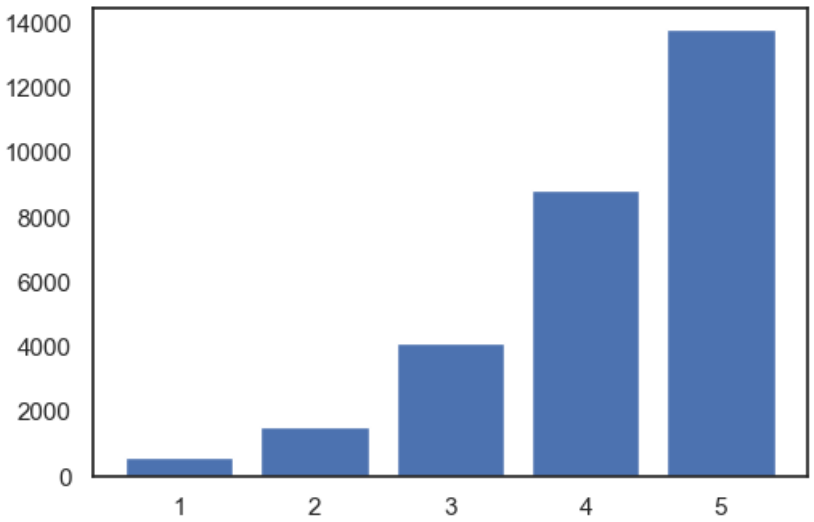
Outline

Khái quát dữ liệu

Bộ dữ liệu được sử dụng trong bài là bộ dữ liệu $reviews - Beauty - 10$ của Amazon kích cỡ 26.2mb: Gồm 1340 người dùng, 733 sản phẩm và 28798 đánh giá.

reviewerID	itemID	Rating	reviewTime
A6VPK7X53QNAQ	B0000CC64W	5	06 18, 2009
A3CHMHGSJSQ02J	B0000CC64W	5	01 18, 2013
A1V1EP514B5H7Y	B0000CC64W	5	11 29, 2011
A1X2LENOF84LCQ	B0000CC64W	4	04 13, 2005
A2PATWWZAXHQYA	B0000CC64W	1	12 21, 2013
A3IOCPLIMYDBCD	B0000CC64W	5	07 4, 2009
A5A3C6XVDYUND	B0000CC64W	3	02 18, 2013
A3V6Z4RCDGRC44	B0000CC64W	4	07 1, 2007
A2WW57XX2UVLM6	B0000CC64W	4	11 22, 2013
A3M7R4PD0FEPUB	B0000CC64W	5	09 28, 2009
A3QYDL5CDNYN66	B0000CC64W	5	12 30, 2012
A3CG93783LP0FO	B0000CC64W	5	07 22, 2010
A2D1LPEUCTNT8X	B000142FVW	5	10 23, 2011
A2H44WVZS59KKT	B000142FVW	5	12 12, 2012
A2ZY49IDE6TY5I	B000142FVW	5	05 30, 2012
ANHL7BB84WJMF	B000142FVW	2	05 16, 2014
A1GUX6R8DV3ZLY	B000142FVW	5	09 16, 2011

Khám phá dữ liệu



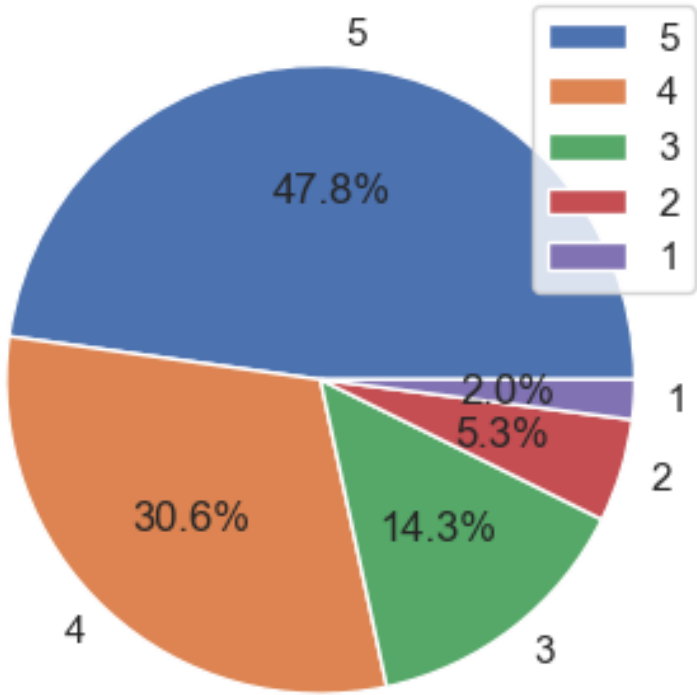


Figure: Đánh giá mà người dùng đã cho

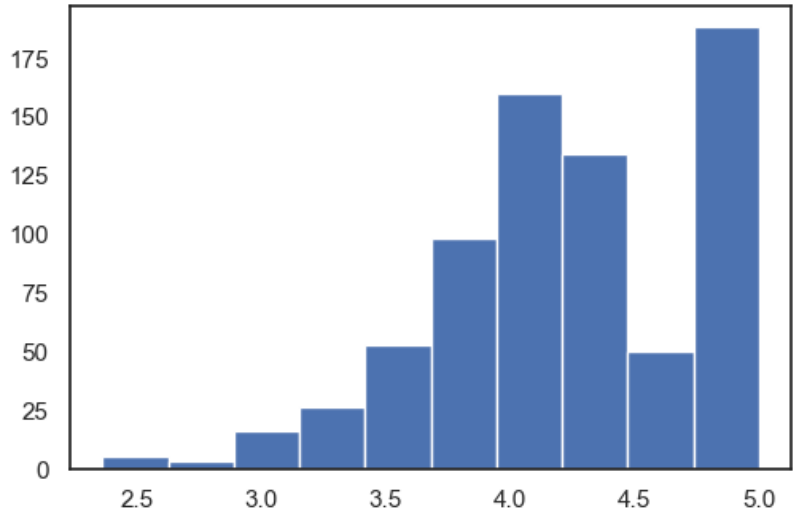


Figure: Đánh giá trung bình của mỗi sản phẩm

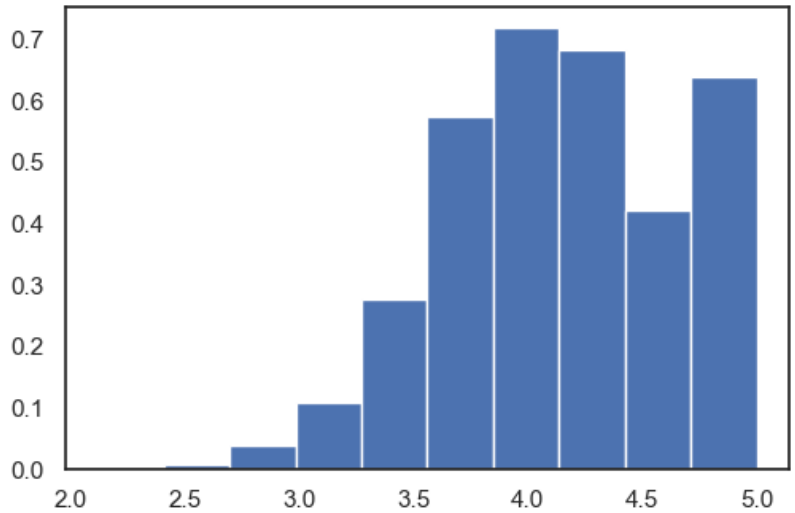


Figure: Đánh giá trung bình được cho bởi người dùng

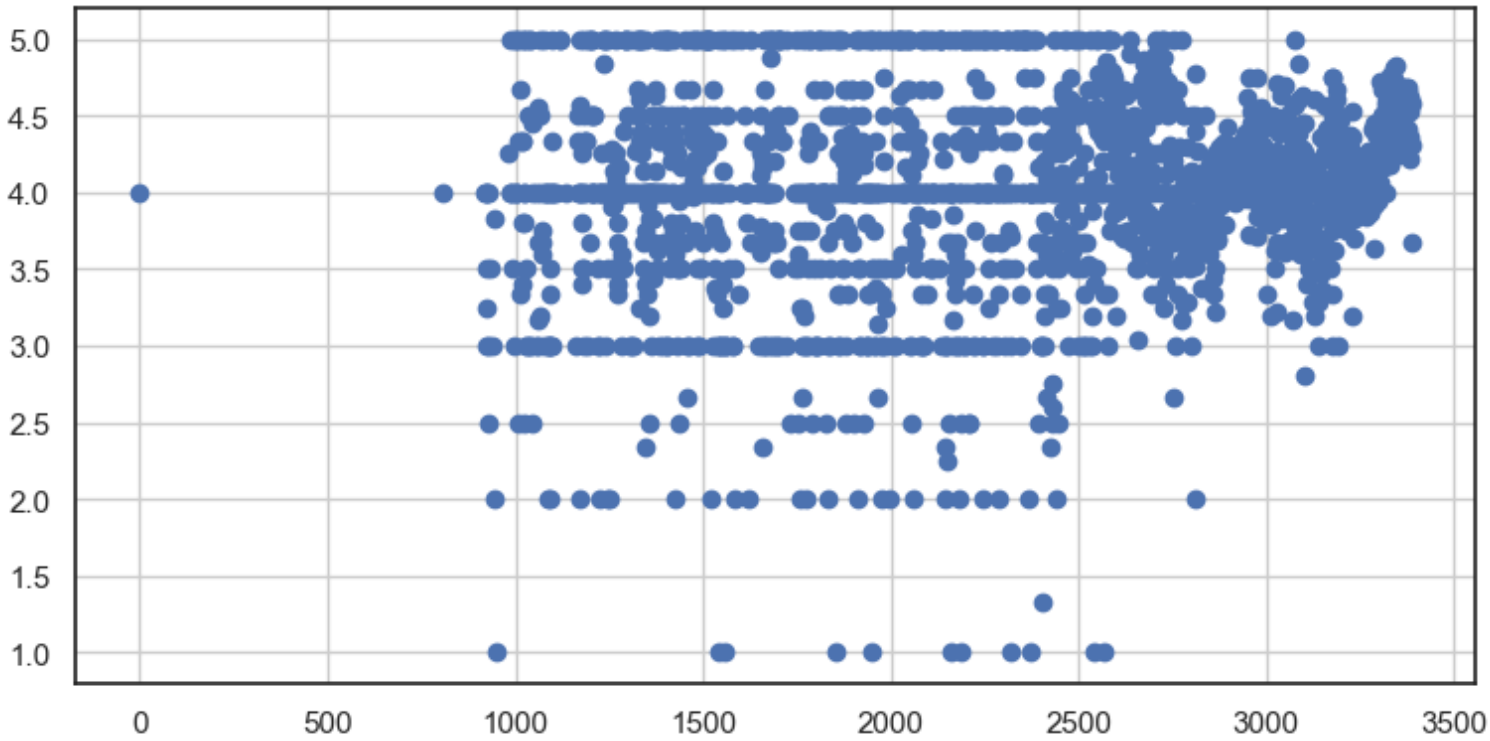
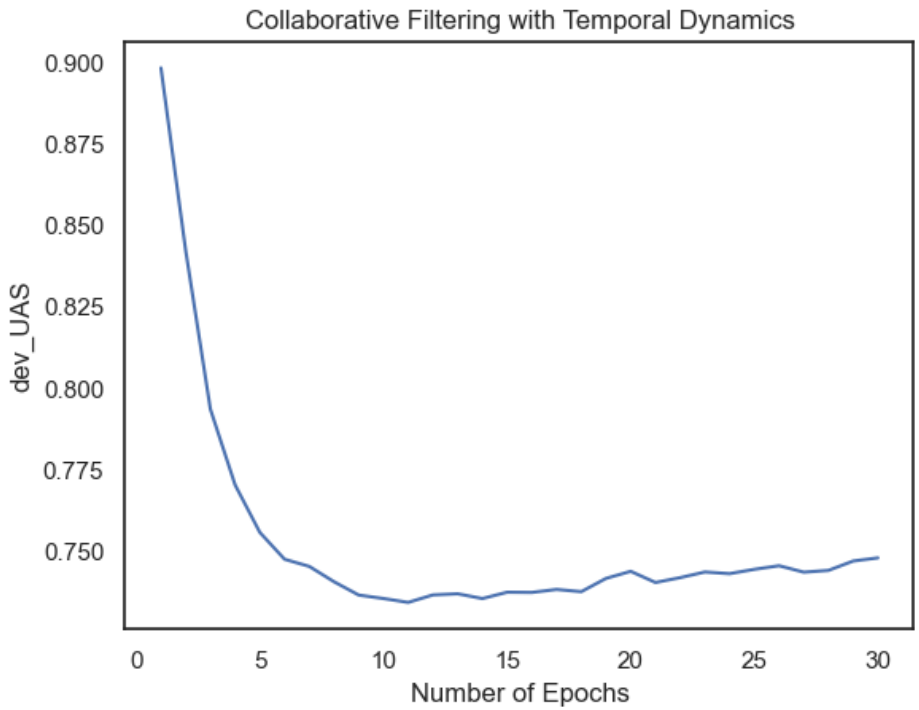


Figure: Đánh giá trung bình cho từng ngày

Kết quả thực nghiệm

Tập dữ liệu được chia tập train/ tập dev/ tập test theo tỷ lệ 60/20/20. Sau 30 epoch thu được kết quả MSE trên tập dev giảm dần đến epoch thứ 13 và từ epoch 15 có hiện tượng overfitting.



Tập trọng số tốt nhất (MSE nhỏ nhất) trên tập dev sẽ được dùng để đánh giá trên tập test. MSE của tập test là 0.692255910708550 4

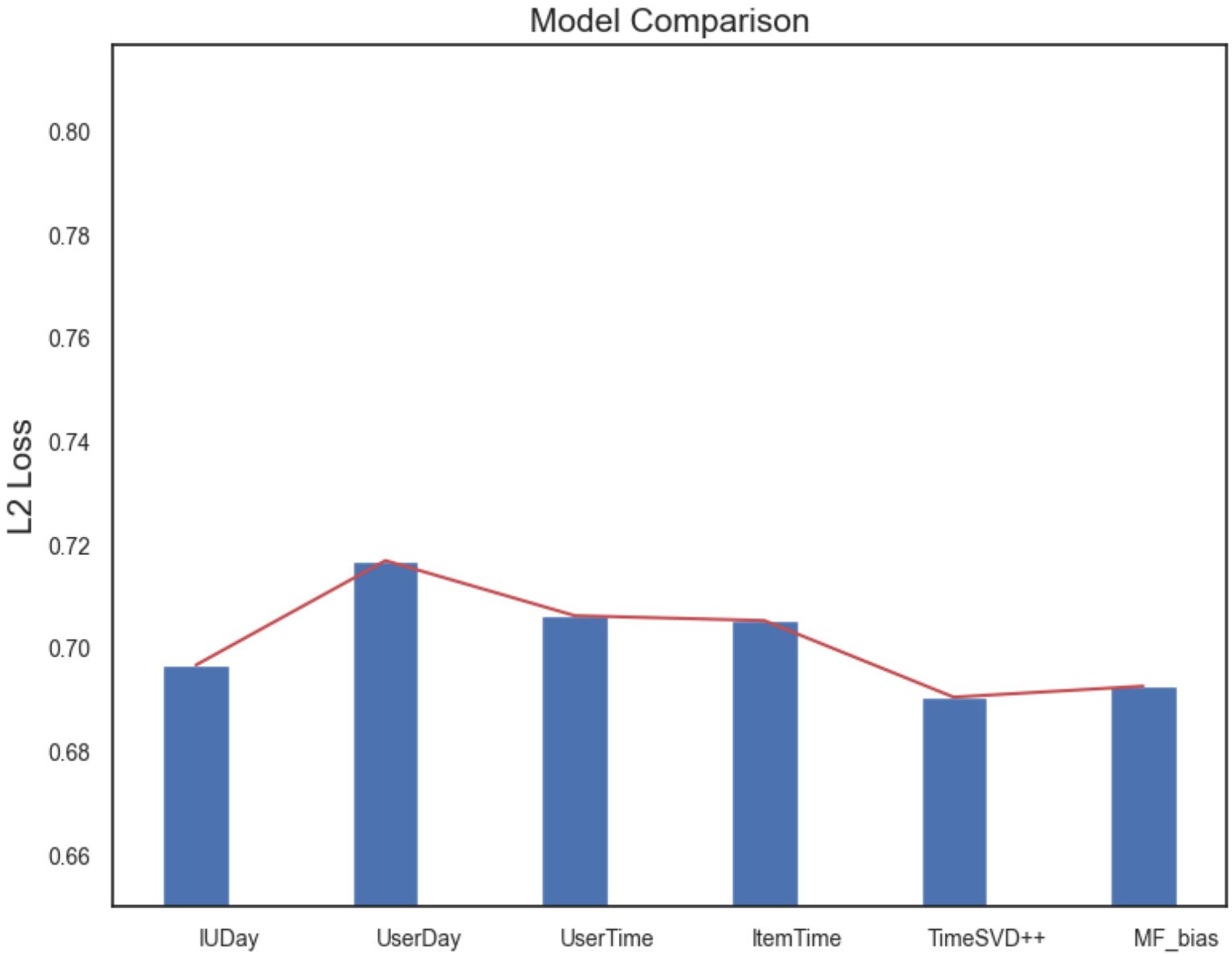


Figure: Kết quả khi so sánh với các phương pháp khác

Outline

Tổng kết

Trong khuân khổ của nội dung đồ án 2 một số nội dung mà em đã đạt được:

- Thành công trong việc giới thiệu phương pháp phân rã ma trận TIMESVD++.
- Ứng dụng để dự báo sở thích người dùng trên tập dữu liệu của AMAZON.
- So sánh hiệu quả mô hình TIMESVD++ với các mô hình sử dụng phương pháp khác.

Tài liệu tham khảo

[1] Jimmy Ba Diederik P. Kingma. "Adam: A Method for Stochastic Optimization". In: Published as a conference paper at the 3rd International Co 322.10 (1905), pp. 891–921.

[2] Yi Ding and Xue Li. "Time Weight Collaborative Filtering". In: Proceedings CIKM '05. Bremen, Germany: Association for Computing Machinery, (2005), pp. 485–492. ISBN: 1595931406. DOI: 10.1145/1099554.1099689. URL: https://doi.org/10.1145/1099554.1099689.

[3] Yehuda Koren. "Collaborative Filtering with Temporal Dynamics". In: In Computer 42 (Aug.2009), pp. 30–37.

[4] R.Bell Y.Koren and C. Volinsky. "Matrix Factorization Techniques for Recom- mender Systems". In: In Computer 42 (Aug.2009), pp. 30–37.