Tạp chí Khoa học Trường Đại học Cần Thơ
website: sj.ctu.edu.vn

XÂY DỰNG HỆ THỐNG GỢI Ý PHIM DỰA TRÊN MÔ HÌNH NHÂN TỐ LÁNG GIỀNG

Triệu Vĩnh Viêm[1],Triệu Yến Yến[1] và Nguyễn Thái Nghe[2]
[1] *Khoa Công nghệ Thông tin, Trường Đại học Bạc Liêu*
[2] *Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ*

Thông tin chung:
Ngày nhận: 03/09/2013
Ngày chấp nhận: 21/10/2013

Title:
Building a movie recommender system using factor in the neighbors approach

Từ khóa:
Lọc cộng tác, hệ thống gợi ý

Keywords:
Collaborative filtering, recommender system

ABSTRACT

Recommender system can provide suitable items to users by using data about their behavior in the past to predict the future items that users may like. Two successful approaches in recommender system (relying on the collaborative filtering) are the latent factor models which identify potential relationships on both the user and the item; and neighbor models which use similarities between the items or the similarities between the users. In this study, we introduce an approach which is based on the method proposed by Koren (2010) to utilize the advantages of both the aforementioned approaches. Moreover, besides building a web-based movie recommender system, we try to improve the prediction results by adding to the original model several new regularization coefficients for different models' parameters.

TÓM TẮT

Hệ thống gợi ý có thể đưa ra những mục thông tin phù hợp cho người dùng bằng cách dựa vào dữ liệu về hành vi trong quá khứ của họ để đự đoán những mục thông tin mới trong tương lai mà người dùng có thể thích. Hai tiếp cận thành công trong hệ thống gợi ý thuộc vào nhóm lọc công tác là mô hình nhân tố tiềm ẩn - xác định mối quan hệ tiềm ẩn trên cả người dùng và mục thông tin; và mô hình láng giềng - phân tích độ tương tự giữa các mục thông tin với nhau hay giữa những người dùng với nhau. Trong bài viết này, chúng tôi giới thiệu một tiếp cận tích hợp các ưu điểm của cả hai tiếp cận trên dựa vào phương pháp đã được đề xuất bởi Koren (2010). Ở đây, bên cạnh việc xây dưng một hệ thống trên nền web để gợi ý phim ảnh cho người dùng, chúng tôi cũng đã điều chỉnh mô hình đã có bằng cách đưa vào các hệ số regularization trên từng tham số khác nhau của mô hình nhằm cải tiến kết quả dự đoán.

1 GIỚI THIỆU

Sự phát triển của internet đã đưa chúng ta vào thế giới với một lượng lớn các phần tử thông tin như âm nhạc, phim ảnh, sách vở, trang web,… với những đặc tính khác nhau. Kết quả của những thông tin khổng lồ đó, người ta cảm thấy rối rắm và một câu hỏi đặt ra "Cái nào là thích hợp với tôi hơn?" nảy sinh trong tư duy của họ.

May thay, hệ thống gợi ý có thể chỉ ra các thông tin phù hợp trong số thông tin khổng lồ chưa có trật tự, nó sử dụng các kỹ thuật lọc để chọn ra những loại thông tin đặc trưng nhằm hiển thị các phần tử phù hợp với sở thích của người dùng [1]. Theo cách này, hệ thống có tích hợp tính năng gợi ý sẽ thu hút được người dùng cả về sự hài lòng và tin cậy. Các hệ thống gợi ý tiêu biểu như Amazon, Netflix, IMDb, Youtube, Last.fm, MovieLens… đã

tăng được số lượng khách truy cập nhờ vào tính năng hỗ trợ quyết định này của hệ thống.

Hệ thống gợi ý thường dựa trên lọc cộng tác (Collaborative filtering - CF), dựa trên những hành vi quá khứ của người dùng, ví dụ như: lịch sử giao dịch, đánh giá sản phẩm, thời gian xem một mục tin… và đặc biệt là nó không cần thiết phải tạo ra các hồ sơ tường minh (explicit feedback) cho người dùng. Để gợi ý được các mục tin, hệ thống CF cần so sánh các đối tượng cơ bản khác nhau như các mục tin (items) và người dùng (users). Có hai nhánh nghiên cứu chính của lọc cộng tác là tiếp cận láng giềng (neighborhood approach) và các mô hình nhân tố tiềm ẩn (latent factor models).

Hầu hết các tiếp cận chung nhất của CF là dựa trên mô hình lân cận (Neighborhood Models), mô hình user-user (user-based CF) mà được tác giả phân tích rất rõ trong tài liệu [2]. Bên cạnh đó, một tiếp cận trong [3] dựa trên độ tương tự giữa các phần tử (item-based CF) với quy mô tập dữ liệu rất lớn và đưa ra sự gợi ý chất lượng cao trong thời gian thực hiện.

Mô hình nhân tố tiềm ẩn có dạng tương tự như phương pháp phân tích giá trị đơn (Singular Value Decomposition), chuyển đổi cả các mục tin và người dùng vào cùng một không gian tiềm ẩn của các nhân tố, điều này làm chúng có khả năng so sánh trực tiếp. Bên cạnh đó, nhờ vào khả năng biểu diễn và so sánh các khía cạnh dữ liệu khác nhau, tiếp cận này có xu hướng cung cấp kết quả dự đoán cao hơn mô hình láng giềng [4][5]. Tuy nhiên hầu hết các hệ thống thương mại (Amazon, Tivo,…) vẫn còn sử dụng mô hình láng giềng. Sự phổ biến của mô hình này một phần là nhờ vào tính dễ cài đặt và dễ hiểu.

Trong bài báo này, chúng tôi giới thiệu một mô hình dựa trên mô hình đã được Koren [12] đề xuất. Mô hình này có khả năng cải tiến được độ chính xác ngang bằng với mô hình nhân tố tiềm ẩn. Nó không những duy trì được các thuận lợi của mô hình láng giềng, mà còn xử lý được ngay lập tức vấn đề người dùng mới (còn gọi là Cold Start Problem) khi xếp hạng lần đầu mà không cần phải huấn luyện lại. Tiếp cận này được tác giả đặt tên là "Factor in the Neighbors" (Asymmetric SVD – Koren 2008) với khả năng mở rộng có thể tích hợp được nhiều thông tin đầu vào của người dùng (implicit feedback, explicit feedback) và đạt được độ chính xác đáng kể. Bên cạnh việc kế thừa kết quả nghiên cứu của Koren, chúng tôi đã điều chỉnh mô hình đã có bằng cách đưa vào các hệ số chính tắc hóa (regularized) trên từng tham số khác nhau của mô hình nhằm cải tiến kết quả dự đoán. Chúng tôi sẽ ứng dụng mô hình đề xuất này trong việc xây dựng hệ thống gợi ý phim ảnh.

Cấu trúc còn lại của bài báo được trình bày như sau. Chúng tôi bắt đầu với các công việc liên quan trong phần 2, kể cả việc phân tích thuận lợi và bất lợi của hai tiếp cận chính của lọc cộng tác. Sau đó, chúng tôi trình bày một cách điều chỉnh thêm vào cho tiếp cận Asymmetric SVD, mà cho phép tích hợp mượt mà các thông tin đầu vào trong phần 3. Phần 4 sẽ phô bày kết quả thực nghiệm của mô hình Asymmetric SVD chỉ trên quan hệ giữa các mục tin với điều chỉnh regularization trên các tham số khác nhau, dữ liệu sử dụng là MovieLens 100K. Phần 5 sẽ biểu diễn minh họa một website đã được tích hợp gợi ý. Phần 6, chúng tôi sẽ kết luận về kết quả nghiên cứu và nêu hướng phát triển tiếp theo của phương pháp đề xuất.

2 CÁC TIẾP CẬN LIÊN QUAN TRONG HỆ THỐNG GỢI Ý

Hệ thống biểu diễn các đánh giá của người dùng cho các bộ phim qua ma trận **m** người dùng và **n** bộ phim. Chúng tôi dùng các ký tự để phân biệt người dùng và các mục tin: u, v (đại diện cho người dùng), i, j (đại diện cho các bộ phim). Ký hiệu r_{ui} để chỉ mức độ thích của người dùng u cho một bộ phim i nào đó, giá trị này trong khoảng từ 1 đến 5 đối với dữ liệu MovieLens, $\hat{r}_{ui}$ là dự đoán đánh giá của người dùng u cho bộ phim i. Chúng tôi sử dụng dữ liệu MovieLens 100K với 943 người dùng, 1682 bộ phim, mỗi người dùng đánh giá ít nhất 20 bộ phim, như vậy mật độ của ma trận đánh giá chỉ có 6.3%; trường hợp này còn gọi là vấn đề thưa thớt dữ liệu trong lọc cộng tác. Để đối phó với vấn đề "học vẹt" khi dữ liệu thưa, chúng tôi đã sử dụng các hằng số regularization (λ_1, λ_2, λ_3…) trên từng tham số khác nhau. Giá trị tốt nhất của các hằng số này được xác định thông qua nghi thức kiểm tra chéo.

2.1 Ước lượng cơ sở (Baseline Estimates)

Dữ liệu lọc cộng tác tiêu biểu phô bày ảnh hưởng lớn đến người dùng và phần tử, đó là có vài người dùng đánh giá cao hơn những người khác và cho vài phần tử nhận được đánh giá cao hơn những phần tử khác, mà thường được khắc phục bằng ước lượng cơ sở (baseline estimates, biases) sau [6]:

$$b_{ui} = \mu + b_u + b_i \quad (1)$$

Với μ được biết như là trung bình toàn cục các đánh giá của ma trận m xn. Tham số b_u và b_i cho biết độ lệch quan sát được của người dùng u và

phần tử i. Ví dụ, chúng ta muốn ước lượng đánh giá của người dùng "Trung" cho bộ phim "Chúa tể của những chiếc nhẫn" (Lord of the rings - LoR), với trung bình xếp hạng trên tất cả bộ phim là μ=4.5. Mặt khác, trung bình các đánh giá của LoR có xu hướng cao hơn μ là b_i=0.2 và trung bình các đánh giá của "Trung" thấp hơn μ là b_u=0.5. Như vậy, b_{ui} sẽ là 4.2 (4.5-0.5+0.2) với cặp (Trung, LoR).

2.2 Mô hình láng giềng (Neighborhood models)

Nhiều hệ thống vẫn còn tin cậy để sử dụng tiếp cận này, hình thức trước đây của nó là mô hình user-based CF được tác giả phân tích rất rõ ràng trong tài liệu [2]. Phương pháp này dự đoán các đánh giá bằng cách dựa trên việc ghi nhận lại những người dùng có cùng sở thích. Một cách tiếp cận khác cũng dựa trên độ tương tự nhưng làm việc với các mục tin thay vì người dùng, một đánh giá được ước lượng thông qua các đánh giá đã biết của cùng người dùng trên các mục tin tương tự. Item-based CF thì được sử dụng nhiều hơn nhờ vào khả năng mở rộng và cải tiến được độ chính xác của nó [Bell and Koren 2007b, Takács 2007].

Linden .G [3] đã đề cập các kỹ thuật chung nhất của hệ thống gợi ý, rồi phân tích các đặc điểm quan trọng và xác định độ phức tạp về thời gian của chúng. Như kỹ thuật user-based CF, Linden .G cho rằng mô hình dạng này vấp phải vấn đề chi phí thời gian khi gợi ý, do đối mặt với nguồn dữ liệu lớn (hơn 10 triệu khách hàng, hơn 1 triệu phần tử). Ngoài ra, trong [3] tuy sử dụng item-based CF nhưng không xây dựng ma trận item-item, do nhiều cặp phần tử không có những khách hàng chung, hao tốn về thời gian xử lý, cũng như không gian bộ nhớ lưu trữ. Nhóm tác giả đã xây dựng giải thuật tìm các phần tử gợi ý bằng cách tìm độ tương tự của một phần tử i (đã được khách hàng mua hoặc đánh giá – người dùng cần gợi ý) với tập các phần tử i' trong R có liên quan với nó (lịch sử những khách hàng khác đã mua hoặc đánh giá i mà cũng mua i'∈ R), rồi gợi ý các phần tử phổ biến hoặc tương quan nhất. Tiếp cận này gọi tóm tắt là tìm các phần tử mà khách hàng có xu hướng mua cùng nhau.

Các phương pháp láng giềng trở nên phổ biến bởi vì chúng trực quan và dễ dàng liên hệ để cài đặt. Một số đặc tính hữu ích của nó là: Khả năng giải thích (Explainability) – người dùng mong chờ một hệ thống có thể đưa ra lý do cho các gợi ý của nó, khác hơn là phải đối mặt những gợi ý "hộp đen" ("black box"); xử lý được các đánh giá mới, có khả năng cung cấp lời gợi ý ngay lập tức với Item-based CF.

Tuy nhiên, [Bell and Koren 2007b] đã nhấn mạnh vài điểm đáng chú ý về mô hình láng giềng. Một câu hỏi đặt ra của họ là tính phù hợp của độ đo tương tự khi chỉ cô lập cho 2 mục tin mà không phân tích trên một tập đầy đủ các láng giềng. Để khắc phục khó khăn này, các tác giả đã xây dựng một phương pháp láng giềng mới mà cần phải tính các trọng số thêm vào mô hình như sau:

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in S^k(i,u)} \theta_{ij}^u \left(r_{uj} - b_{uj}\right) \qquad (2)$$

Chi tiết hơn về mô hình mới này có thể xem trong tài liệu [7]. Mô hình này được tác giả cải tiến với trọng số quan hệ giữa các phần tử không phụ thuộc vào người dùng cụ thể, cải tiến về độ phức tạp và khả năng tận dụng các thông tin đầu vào với tiếp cận nhân tố trong mô hình láng giềng.

2.3 Mô hình nhân tố tiềm ẩn (Latent factor models)

Tiếp cận phổ biến gần đây của mô hình này là phân giải giá trị đơn (Singular Value Decomposition - SVD) nhờ vào sự hấp dẫn về độ chính xác đạt được và khả năng mở rộng hệ thống. Ngoài ra, phân rã ma trận (Matrix Factorization - MF) hiện đang phát triển rất mạnh với khả năng dự đoán cho lỗi rất thấp.

Đối với mô hình phân rã ma trận cơ bản thì chỉ có một quan hệ giữa 2 thực thể chính cần dự đoán được xem xét, do vậy thiếu độ chính xác dự đoán trong toàn bộ lĩnh vực thực nghiệm [8]. Từ đó, nghiên cứu trong [9] cố gắng cải tiến tiếp cận phân rã ma trận cơ bản với nhiều hướng mở rộng khác nhau của việc cực đại hóa lề trong phân rã ma trận (Maximum Margin Matrix Factorization - MMMF), nghiên cứu trong [10] cũng cố gắng phối hợp thông tin nội dung trực tiếp như là ràng buộc tuyến tính vào tiếp cận phân rã ma trận. Đặc biệt bài báo [11] đã xây dựng một tiếp cận mới cho mô hình nhân tố tiềm ẩn đó là phân rã ma trận đa quan hệ (Multi-Relational Matrix Factorization - MRMF) cho lĩnh vực thực nghiệm là phim ảnh (movies) và gene function prediction.

Mặt khác, các phương pháp mô hình nhân tố tiềm ẩn thường vấp phải vấn đề học lại (relearning) khi có các đánh giá mới (new ratings) [12]. Mô hình dạng này cho kết quả dự đoán có độ chính xác cao nhưng khó đưa ra được lời giải thích cho điều đó.

Trong phần tiếp theo, chúng tôi sẽ giới thiệu phương pháp "Asymmetric SVD với chính tắc hóa

trên từng tham số" dựa trên phương pháp của Koren [12].

3 ASYMMETRIC SVD VỚI CHÍNH TẮC HÓA TRÊN TỪNG THAM SỐ

Với tiếp cận lọc cộng tác, nếu sử dụng phương pháp dựa vào láng giềng thì tùy vào tính quy mô của hệ thống mà ảnh hưởng đến thời gian phản hồi gợi ý, độ chính xác dự đoán chưa cao. Hơn nữa, trong [13] còn đặt một câu hỏi là "tính phù hợp của độ đo tương tự khi nó chỉ cô lập quan hệ của hai phần tử, mà không phân tích sự tương tác bên trong tập hợp đầy đủ các láng giềng", vấn đề người dùng mới cũng tác động đến phương pháp này. Mặt khác, nếu dùng phương pháp dựa trên nhân tố tiềm ẩn thường mất chi phí về thời gian để tìm kiếm tham số, tuy nó giải quyết được vấn đề thưa thớt của dữ liệu và độ chính xác của dự đoán tương đối cao nhưng phải học lại mô hình khi có các đánh giá mới và khó đưa ra được lời giải thích với kết quả dự đoán. Trong [14], Gábor Takács đã đề nghị một phương pháp kết hợp cả Neighborhood Model và Matrix Factorization mà tạo ra hệ thống gợi ý đơn giản, có khả năng mở rộng và đạt được độ chính xác.

Chúng tôi chọn mô hình Asymmetric-SVD được biết như là "Factor in the Neighbors" [12][13] để làm chức năng gợi ý của hệ thống, một phương pháp không những cho phép tận dụng được những ưu điểm của mô hình láng giềng và mô hình nhân tố tiềm ẩn mà còn đạt hiệu quả tốt về thời gian và không gian [12], công thức dự đoán đánh giá như sau:

$$\hat{r}_{ui} = \mu + b_u + b_i + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) q_i^T x_j \quad (3)$$

Tài liệu [13] đã phân tích được nhiều đặc điểm thuận lợi của tiếp cận này. Ngoài ra, tài liệu [12] còn nói rõ hơn sự kết hợp mượt mà của nhiều nguồn thông tin đầu vào từ người dùng bao gồm: phản hồi explicit và implicit.

Công thức (3) ở đây của chúng tôi chỉ thực nghiệm trên việc phân tích mối quan hệ giữa các mục tin bên trong không gian tiềm ẩn. Đặc biệt hơn, chúng tôi đã chính tắc hóa (regularized) trên từng tham số khác nhau trong hàm chi phí như sau:

$$\min_{q_*, x_*, b_*} E = \sum_{(u,i) \in K} \left(r_{ui} - \mu - b_u - b_i - q_i^T |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j \right)^2 + \left(\lambda_1 b_u^2 + \lambda_2 b_i^2 + \lambda_3 \|q_i\|^2 + \lambda_4 \sum_{j \in R(u)} \|x_j\|^2 \right) \quad (4)$$

Mô hình này sử dụng L2 – regularization (cụ thể là chuẩn Frobenius) với các hằng số regularization khác nhau (λ_1, λ_2, λ_3, λ_4) để có thể cải tiến được độ chính xác dự đoán so với khi chỉ dùng duy nhất một hằng số cho tất cả các tham số. Sự chính tắc (regularization) làm việc bằng cách thêm vào "hình phạt" liên quan tới tham số trong hàm chi phí của giả thuyết để trừng phạt các trọng số lớn [15].

Hệ thống dựa trên các phản hồi tường minh cố gắng học được các nhân tố thể hiện quan hệ giữa các bộ phim đánh giá trong quá khứ của người dùng và các bộ phim cần dự đoán, trong đó các đánh giá được tập hợp từ MovieLens. Chúng tôi sử dụng giải thuật ***stochastic gradient descent*** để phát hiện quan hệ giữa các bộ phim đã đánh giá trước đây của người dùng và các bộ phim cần dự đoán (mô hình phân rã trọng số quan hệ giữa các bộ phim).

Dưới đây là giải thuật cài đặt cho mô hình huấn luyện tìm các tham số.

FactorizedNeighborhoodModel(Known ratings: r_{ui}, numberOfFactor: f, learnRate: β, Iterations, λ_1, λ_2, λ_3, λ_4)
{
  $q_i \leftarrow N(\mu, \sigma^2)$; $x_j \leftarrow N(\mu, \sigma^2)$
  for(count=1; count<=Iterations; count++) {
    for(u=1; u<=m; u++) {
      $p_u = |R(u)|^{-\frac{1}{2}} \sum (r_{uj} - b_{uj}) x_j$
      sum $\leftarrow$ 0
      for(all i $\in$ R(u)) {
        $\hat{r}_{ui} = \mu + b_u + b_i + q_i p_u^T$;
        $e_{ui} = r_{ui} - \hat{r}_{ui}$;
        sum = sum + $e_{ui} \cdot q_i$;

$$q_i = q_i + \beta \cdot (e_{ui} \cdot p_u - \lambda_3 \cdot q_i)$$
$$b_u = b_u + \beta \cdot (e_{ui} - \lambda_1 \cdot b_u)$$
$$b_i = b_i + \beta \cdot (e_{ui} - \lambda_2 \cdot b_i)$$

```
    } // end all i ∈ R(u)
    for(all i ∈ R(u)) { // tổng bên trong cho x
```

$$x_i = x_i + \beta \cdot \left(|R(u)|^{-\frac{1}{2}} \cdot (r_{ui} - b_{ui}) \cdot \text{sum} - \lambda_4 \cdot x_i \right)$$

```
    }
  } // end all u
} // end steps Iterations
return (qi, xi,bi| i=1,…,n;bu | u=1,…,m
} // end Algorithm
```

Các tham số của mô hình lúc này là q_i, x_i, b_u, b_i, Koren đề xuất phân biệt cách tính các thành phần bias khác nhau [12], khi xét quan hệ giữa các mục tin thì vẫn chọn các bias là hằng số ban đầu, còn các b_u, b_i bên ngoài quan hệ sẽ được tính toán và cập nhật trong quá trình huấn luyện. Với tiếp cận này thì độ phức tạp về thời gian tính toán dự đoán là $0(f - \sum_u |R(u)|)$ và độ phức tạp về không gian tuyến tính với kích cỡ đầu vào 0(m+nf).

4 KẾT QUẢ THỰC NGHIỆM

4.1 Tập dữ liệu huấn luyện

Nghiên cứu này, chúng tôi sử dụng tập dữ liệu MovieLens 100K[1], mỗi người dùng đánh giá ít nhất 20 bộ phim với các tập dữ liệu dùng cho kiểm tra chéo 5-fold sẵn có được chia ra tập train (uX.base) và tập test (uX.test) với X từ 1 đến 5. Chúng tôi tiền xử lý các tập dữ liệu này bao gồm xóa bỏ trường timestamp, trộn ngẫu nhiên các mẫu, chèn vào số mẫu ở đầu mỗi tệp.

4.2 Độ đo

Khi huấn luyện, chúng tôi thử tìm kiếm trên nhiều trường hợp của các siêu tham số (meta-parameters) khác nhau, sử dụng kỹ thuật tìm kiếm lưới (grid search = raw search + smooth search) để đạt được các siêu tham số cho lỗi RMSE trên tập kiểm tra tốt nhất. Lỗi RMSE được xác định bằng công thức:

$$\text{RMSE} = \sqrt{\frac{1}{|D^{\text{test}}|} \sum_{u,i,r \in D^{\text{test}}} (r_{ui} - \hat{r}_{ui})^2} \qquad (5)$$

4.3 Kỹ thuật huấn luyện và kết quả

Chúng tôi sử dụng bước lặp giới hạn (Max_Num_Iters) thay vì lặp cho đến khi hội tụ để quá trình huấn luyện nhanh hơn và giải quyết vấn đề phần tử mới với dự đoán trung bình toàn cục.

Sau đây, chúng tôi thống kê một số kết quả theo từng tốc độ học (LR), số nhân tố (NF) và các regularization (Rb_u, Rb_i, Rq_i, Rx_i). Kết quả thực nghiệm tìm kiếm thô được thể hiện trên 3 tốc độ học: 0.001, 0.005 và 0.01 với số nhân tố 16, 32, 64, cùng với số vòng lặp giới hạn như các bảng sau (trích từ dữ liệu với lỗi RMSE thấp):

Bảng 1: Iters=50, NF=64

LR	Rb_u	Rb_i	Rq_i	Rx_i	RMSE	Time(m)
0.01	0.005	0.005	0.05	0.5	0.9289	0.2290
0.005	0.05	0.05	0.05	0.5	**0.9224**	0.2233
0.001	0.05	0.005	0.005	0.005	0.9318	0.2249

Bảng 2: Iters=100, NF=64

LR	Rbu	Rbi	Rqi	Rxi	RMSE	Time(m)
0.01	0.05	0.005	0.5	0.05	0.9346	0.4467
0.005	0.05	0.005	0.05	0.5	0.9288	0.4456
0.001	0.05	0.005	0.05	0.05	0.9245	0.4449

Bảng 3: Iters=200, NF=64

LR	Rb_u	Rb_i	Rq_i	Rx_i	RMSE	Time(m)
0.01	0.005	0.005	0.5	0.05	0.9347	0.896
0.005	0.005	0.005	0.5	0.05	0.9328	0.892
0.001	0.05	0.05	0.05	0.5	**0.9225**	0.891

Với tiếp cận nhân tố trong mô hình láng giềng [12] số nhân tố cao sẽ cho lỗi dự đoán tốt hơn số nhân tố thấp trên cùng bộ siêu tham số, nên các bảng 1-2-3 thể hiện các giá trị tốt nhất với NF là 64. Sau khi tìm kiếm thô, chúng tôi tiếp tục tìm kiếm mịn hơn trên các giá trị siêu tham số tốt nhất này (dòng tô đen trong Bảng 1), nhằm đạt được độ đo lỗi kiểm tra tốt hơn. Sau đây là kết quả mịn hóa tốt nhất (Bảng 4):

Bảng 4: Tìm mịn

Rb_u	Rb_i	Rq_i	Rx_i	RMSE	Time(m)
0.05	0.0525	0.0475	0.5	**0.92190**	0.2220
0.0475	0.0475	0.0525	0.485	0.92196	0.2179
0.0525	0.0525	0.0475	0.5	0.92199	0.2204
0.0525	0.05	0.0475	0.515	0.92212	0.2202
0.0475	0.0475	0.05	0.515	0.92224	0.2238
0.05	0.0525	0.05	0.485	0.92225	0.2206
0.05	0.05	0.0475	0.485	0.92229	0.2197

Do tính ngẫu nhiên của dữ liệu nên chúng tôi sẽ lặp lại 20 lần việc tính toán kết quả của 8 trường hợp trên rồi lấy kết quả trung bình để tìm được chính xác trường hợp nào là tốt nhất thật sự (Bảng 5).

Bảng 5: Tìm mịn lặp

Rb_u	Rb_i	Rq_i	Rx_i	RMSE	Time(m)
0.0525	0.05	0.0475	0.515	**0.922784**	0.216735
0.05	0.0525	0.05	0.485	0.922829	0.217155
0.05	0.05	0.0475	0.485	0.922841	0.215957
0.0475	0.0475	0.05	0.515	0.922914	0.217372
0.05	0.0525	0.0525	0.5	0.922917	0.216496
0.05	0.0525	0.0475	0.5	0.922925	0.216537
0.0525	0.0525	0.0475	0.5	0.922952	0.218518
0.0475	0.0475	0.0525	0.485	0.922967	0.220743

Thống kê (Bảng 4 và 5) cho ta thấy dữ liệu ngẫu nhiên đã ảnh hưởng ít nhiều đến RMSE (0.9219 và 0.922784).

Các bảng trên là kết quả huấn luyện tìm kiếm siêu tham số trên tập dữ liệu (u1.base, u1.test). Qua đó rõ ràng các siêu tham số regularization khác nhau (Rb_u, Rb_i, Rq_i, Rx_i) sẽ đạt kết quả tốt hơn nếu chỉ dùng cùng một regularization cho tất cả các tham số. Tiếp theo chúng tôi lấy trung bình lỗi trên tất cả các tập dữ liệu của MovieLens 5-fold với cùng các siêu tham số trên tập u1, kết quả cuối cùng như sau:

Bảng 1: Kết quả trên toàn bộ các tập dữ liệu với nghi thức 5-fold

Siêu tham số tìm thô				**Dữ liệu**	**RMSE**
LR=0.005, NF=64, Num_Iter=50				u1	0.9225545
				u2	0.9156835
				u3	0.9087205
Rb_u	Rb_i	Rq_i	Rx_i	u4	0.9106193
0.05	0.05	0.05	0.5	u5	0.9135651
Trung bình					**0.914229**
Siêu tham số tìm mịn				**Dữ liệu**	**RMSE**
LR=0.005, NF=64, Num_Iter=50				u1	0.9228189
				u2	0.9154739
				u3	0.9081727
Rb_u	Rb_i	Rq_i	Rx_i	u4	0.9104163
0.0525	0.05	0.0475	0.515	u5	0.9136031
Trung bình					**0.914097**

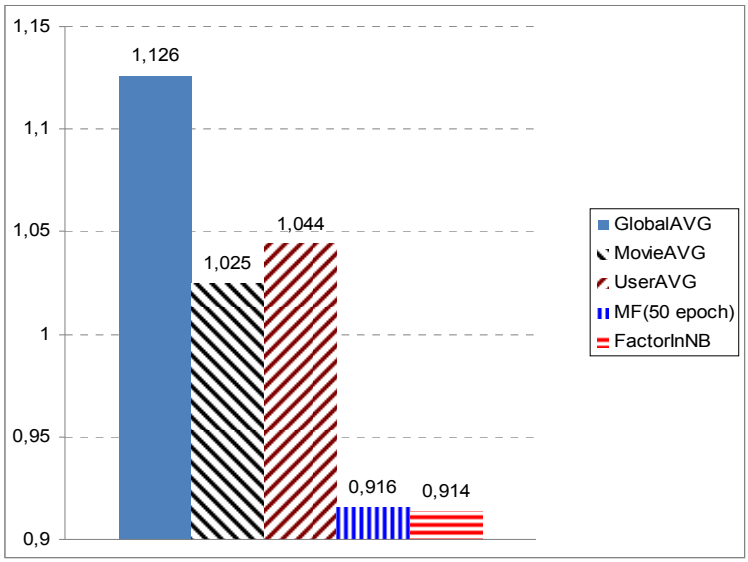
Mỗi tập dữ liệu chúng tôi lặp 5 lần huấn luyện để tính RMSE chính xác

Số liệu trong Bảng 6 đã nói lên được kết quả tìm kiếm mịn có thể cho độ đo RMSE tốt hơn tìm kiếm thô.

4.4 Biểu đồ so sánh

Phần này trình bày so sánh tiếp cận của bài báo với các tiếp cận khác thông qua biểu đồ. Chúng tôi so sánh tiếp cận này với dự đoán trung bình toàn cục (GlobalAVG), trung bình trên bộ phim (MovieAVG), trung bình trên người dùng (UserAVG) và phân rã ma trận có xử lý "cold start problem" (Matrix Factorization - MF) trên cùng tập dữ liệu MovieLens với nghi thức kiểm tra chéo 5-fold.

Dưới đây là biểu đồ biểu diễn độ đo RMSE của các phương pháp đã đề cập.



Hình 1: So sánh các tiếp cận gợi ý

Biểu đồ cho ta thấy lỗi RMSE của tiếp cận Asymmetric SVD với chính tắc hóa trên từng tham số cũng khá tốt hơn phân rã ma trận và tốt hơn nhiều khi so sánh với dự đoán bằng trung bình toàn cục cũng như trung bình trên bộ phim và người dùng.

Thực nghiệm ở đây, chúng tôi chỉ mới xét kết hợp mối quan hệ tiềm ẩn giữa các mục tin, nếu hệ thống được bổ sung thêm nhiều thông tin đầu vào như các phản hồi không tường minh (implicit feedbacks) thì theo [12] RMSE đạt được có thể còn tốt hơn.

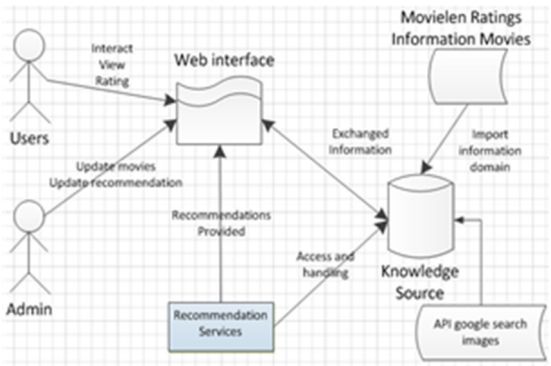
5 XÂY DỰNG HỆ THỐNG

Chúng tôi đã xây dựng một hệ thống gợi ý trên nền web với công nghệ servlets/jsp và cơ sở dữ liệu MySQL.

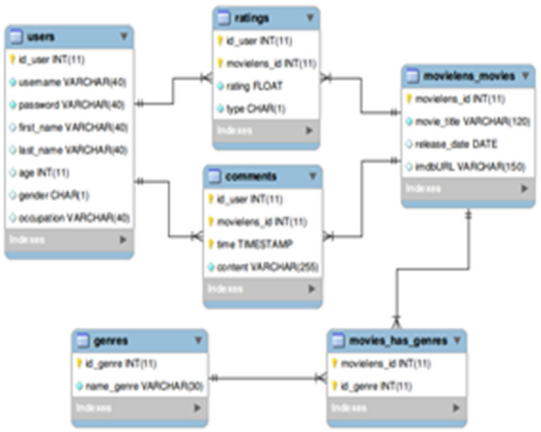
5.1 Kiến trúc tổng quát

Hệ thống rút trích dữ liệu từ tập Movielens 100K và cũng do trong tập dữ liệu này chưa có nhiều nội dung về các bộ phim (đặc biệt là hình ảnh), cho nên nhóm đã sử dụng dịch vụ tìm kiếm ảnh API Google Search Images để tìm dữ liệu cho hệ thống. Người dùng hệ thống có thể chia ra làm 3 nhóm khác nhau: khách vãng lai, người dùng đã đăng ký, người quản trị hệ thống.

Khách vãng lai chỉ được xem và tìm kiếm nội dung thông tin của phim ảnh. Người dùng đã đăng ký sau khi đăng nhập thành công sẽ được tất cả các quyền của khách vãng lai, quyền xếp hạng phim ảnh, bàn luận về chúng và điều đáng nói là họ nhận được các lời gợi ý riêng của mình. Nhóm quản trị hệ thống có thể thêm phim mới và cập nhật được tham số hỗ trợ tính toán gợi ý.



Hình 2: Kiến trúc hệ thống



Hình 3: Mô hình dữ liệu

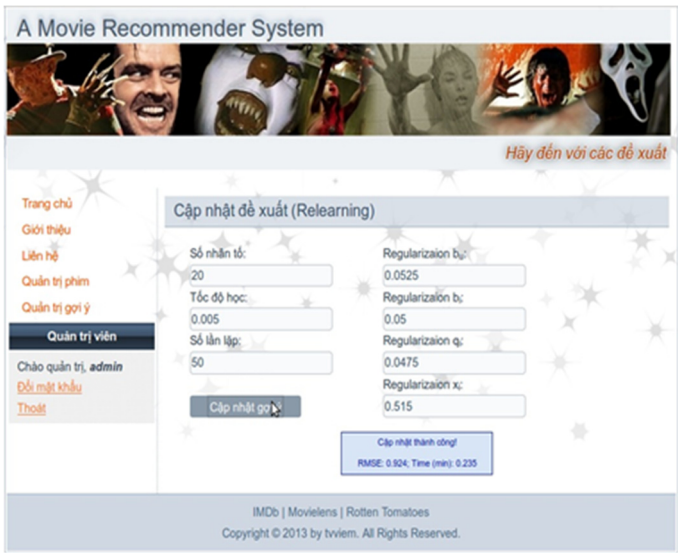
5.2 Mô hình cơ sở dữ liệu

Hệ thống sử dụng MySQL Server Community như là kho dữ liệu. Chúng tôi đã lưu tất cả các tham số sau khi huấn luyện vào cơ sở dữ liệu, những tham số này được dùng để tính toán lời gợi ý ngay sau khi người dùng đăng nhập. Phần tiếp theo sẽ minh họa một số giao diện của hệ thống mà nhóm đã xây dựng.

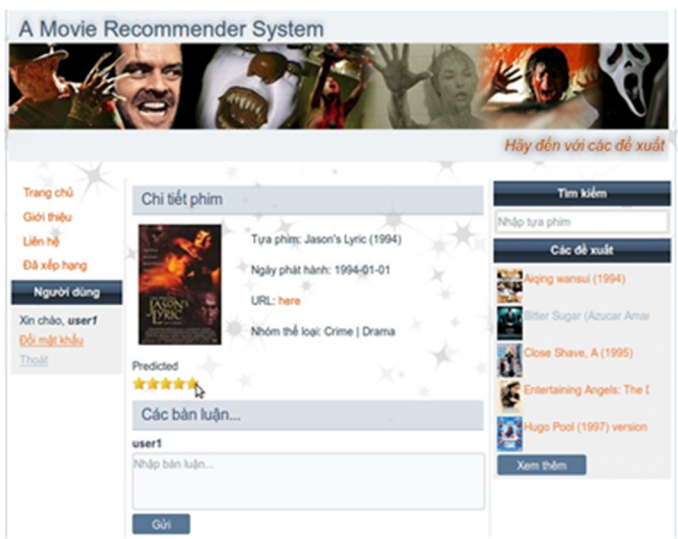
5.3 Các giao diện chính của hệ thống

Người dùng đăng nhập thành công có thể thấy được lời gợi ý cá nhân (nhận thêm đề xuất khác), họ có thể xem chi tiết các bộ phim và xếp hạng lại chúng.

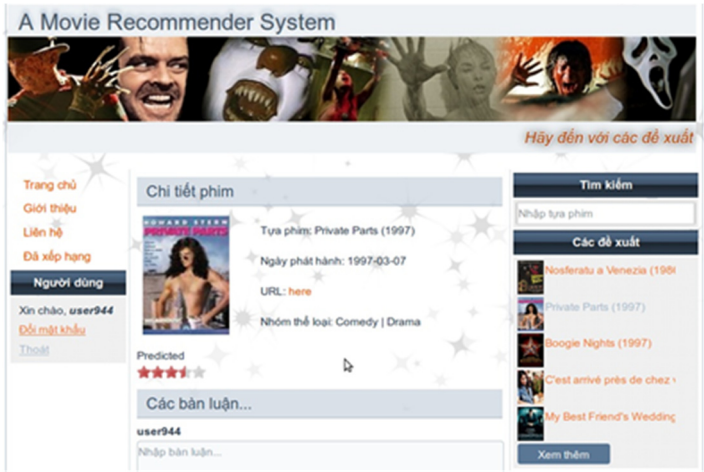
Hơn nữa, hệ thống còn có khả năng giải quyết vấn đề cold start problem với giá trị trung bình toàn cục.



Hình 4:Trang quản trị gợi ý cho phép cập nhật các tham số cho hệ thống (admin)



Hình 5: User đăng nhập thành công, nhận gợi ý, xem chi tiết phim, bàn luận



Hình 6: User944 là người dùng mới được dự đoán trung bình toàn cục và ngẫu nhiên

6 KẾT LUẬN

Phương pháp "Factor in the neighbors" thay vì cung cấp tham số rõ ràng cho những người dùng cụ thể thì nó biểu diễn người dùng thông qua các phần tử mà họ đã đánh giá. Qua kết quả thực nghiệm, ngoài các đặc điểm thuận lợi của tiếp cận này được đề cập trong [12][13], chúng tôi đã xác định được các regularization khác nhau trên từng tham số sẽ cho kết quả tốt hơn thay vì chỉ dùng một regularization cho tất cả các tham số. Bên cạnh đó, chúng tôi đã áp dụng kỹ thuật tìm kiếm lưới cho các siêu tham số để đạt kết quả dự đoán tốt hơn khi chỉ tìm kiếm sơ bộ một vài trường hợp cụ thể nào đó. Nhóm đã xây dựng hệ thống gợi ý nền web với dữ liệu được tập hợp từ Movielens 100K, hệ thống đã thành công trong việc tính toán cũng như cập nhật lời gợi ý trực tuyến. Tuy nhiên, về vấn đề thời gian đăng nhập vẫn còn hạn chế do quá trình nhận các đường dẫn hình ảnh từ dịch vụ API Google Search Images. Vì vậy, tương lai nhóm sẽ lưu trữ dữ liệu hình ảnh trên cơ sở dữ liệu riêng không phụ thuộc vào API google và tiếp tục tận dụng nhiều loại dữ liệu đầu vào của người dùng để dự đoán gợi ý chính xác hơn.

TÀI LIỆU THAM KHẢO

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), (2005).
2. Herlocker, J. L., Konstan, J. A., Borchers, A., Andriedl, J. 1999. An algorithmic framework for performing collaborative filtering. InProceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99). ACM, New York, NY, 230–237.
3. G. Linden, B. Smith, and J. York, 2003. Amazon.com recommendations: Item - item collaborative filtering. IEEE Internet Comput. 7, 1, 76–80.
4. Gábor Takács, István Pilászy, Bottyán Németh: Major components of the Gravity Recommendation System, Volume 9, Issue 2, p80-83, 2007.
5. Arkadiusz Paterek: Improving regularized singular value decomposition for collaborative filtering, KDDCup.07 August 12, 2007, San Jose, California, USA.
6. Yehuda Koren, Yahoo Research, Robert Bell and Chris Volinsky, AT&T Labs - Research: "Matrix Factorization techniques for recommender systems", Published by the IEEE Computer Society, 2009.
7. Bell, R.M. and Koren, Y. 2007b. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In Proceedings of the IEEE International Conference on Data Mining (ICDM). IEEE Computer Society, 43–52.
8. Gábor Takács, István Pilászy, Bottyán Németh: "Investigation of Various Matrix Factorization Methods for Large Recommender Systems", 2nd Netflix-KDD Workshop, August 24, 2008, Las Vegas, NV, USA.
9. Markus Weimer, Alexandros Karatzoglou, Alex Smola: "Improving maximum margin matrix factorization", Mach Learn (2008) 72: 263–276.

10. Peter Forbes, Mu Zhu: “Content-boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation”, RecSys’11, October 23 - 27, 2011, Chicago, Illinois, USA.
11. Christoph Lippert, Stefan Hagen Weber, Yi Huang, Volker Tresp, Matthias Schubert, Hans-Peter Kriegel: “Relation Prediction in Multi-Relational Domains using Matrix-Factorization”, NIPS 2008 workshop on "Structured Input, Structured Output" (SISO 2008).
12. Koren, Y. (2010)”: “Factor in the neighbors: Scalable and accurate collaborative filtering.”, AT&T Labs - Research 180 Park Ave, Florham Park, NJ 07932.
13. Y. Koren: “Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model”, Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008.
14. Gábor Takács, István Pilászy, Bottyán Németh: “Investigation of Various Matrix Factorization Methods for Large Recommender Systems”, 2nd Netflix-KDD Workshop, August 24, 2008, Las Vegas, NV, USA.
15. Aleks Jakulin: “Machine Learning Based on Attribute Interactions”, University of Ljubljana, Se˘zana, June 13, 2005.