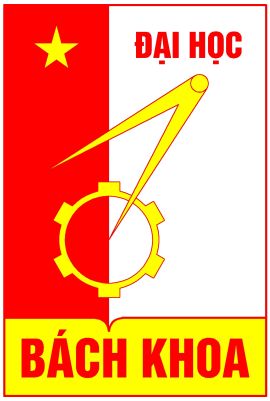
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
_____*******_____



BÁO CÁO
MÔN HỌC
Khai phá Web

Đề tài:

Cài đặt thử nghiệm một số phương pháp gợi ý phim

Sinh viên thực hiện:

Họ và Tên	MSSV	Mã học phần
Đặng Quang Trung	20134145	IT4991

Giáo viên hướng dẫn: TS. Nguyễn Kiêm Hiếu

Hà Nội 11-8-2016

Mục lục

Lời mở đầu

Sự phát triển của công nghệ thông tin và việc ứng dụng công nghệ thông tin trong nhiều lĩnh vực của đời sống, kinh tế xã hội trong nhiều năm qua cũng đồng nghĩa với lượng dữ liệu đã được các cơ quan thu thập và lưu trữ ngày một tích luỹ nhiều lên. Họ lưu trữ các dữ liệu này vì cho rằng trong nó ẩn chứa những giá trị nhất định nào đó. Mặt khác, trong môi trường cạnh tranh, người ta ngày càng cần có nhiều thông tin với tốc độ nhanh để trợ giúp việc ra quyết định và ngày càng có nhiều câu hỏi mang tính chất định tính cần phải trả lời dựa trên một khối lượng dữ liệu khổng lồ đã có. Với những lý do như vậy, các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một khuynh hướng kỹ thuật mới đó là Kỹ thuật phát hiện tri thức và khai phá dữ liệu.

Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới, tại Việt Nam kỹ thuật này tương đối còn mới mẻ tuy nhiên cũng đang được nghiên cứu và dần đưa vào ứng dụng. Khai phá dữ liệu là một bước trong qui trình phát hiện tri thức gồm có các thuật toán khai thác dữ liệu chuyên dùng dưới một số qui định về hiệu quả tính toán chấp nhận được để tìm ra các mẫu hoặc các mô hình trong dữ liệu. Nói một cách khác, mục đích của phát hiện tri thức và khai phá dữ liệu chính là tìm ra các mẫu và/hoặc các mô hình đang tồn tại trong các cơ sở dữ liệu nhưng vẫn còn bị che khuất bởi hàng núi dữ liệu.

Trong bài viết này, em sẽ trình bày một cách tổng quan về Kỹ thuật khai phá dữ liệu. Trên cơ sở đó đưa ra một bài toán Gợi ý phim cho người dùng của một hệ thống và giải quyết bài toán bằng phương pháp Collaborative Filtering.

Lời cảm ơn

Chúng em xin chân thành cảm ơn thầy Nguyễn Kiêm Hiếu đã cung cấp cho chúng em những kiến thức vô cứng bổ ích khi chúng em bắt đầu tìm hiểu về môn học khai phá dữ liệu web cũng như đã giúp đỡ chúng em rất nhiều trong quá trình thực nghiệm và viết bản báo cáo này.

Chương 1

Mô tả bài toán

1.1 Giới thiệu

Hằng ngày chúng ta có các ý kiến về những thứ chúng ta thích hoặc không thích và thậm chí là không quan tâm đến nó. Ví dụ như bạn xem một chương trình truyền hình trên TV, bạn cảm thấy chương trình rất hay và hài hước hoặc thấy nó nhàm chán hay bạn không tìm thấy chương trình đó ở tất cả mọi kênh.Hoặc chương trình đó diễn ra mà chúng ta không để ý.

Các sở thích của mỗi người là khác nhau, nhưng chúng ta tạo ra các dạng mẫu người dùng. Mọi người có xu hướng thích những thứ tương tự như những thứ mà họ thích. Bởi vì nếu tôi thích bộ phim Matrix, bạn có thế đoán là tôi cũng muốn xem báo cáo về Minority, những cái này chủ yếu cùng một thể loại hành động và khoa học viễn tưởng. Tương tự như vậy moi người cũng có khuynh hướng thích những thứ mà người tương tự thích. Gợi ý là đề xuất tất cả về dự đoán những điều thích và không thích này và sử dụng chúng để khám phá những điều mới và đáng mong bạn chưa biết về chúng.

1.2 Gơi ý là gì?

Trong báo cáo này sẽ nói về một số cách mọi người đưa ra các khuyến nghị và khám phá các điều mới. Có một vài chiến lược sử dụng để tạo ra các khuyến nghị: Một là có thể đựa vào những người có cùng thị hiếu sở thích. Hoặc phương pháp khác sẽ tìm ra những thứ giống như những gì chúng ta đã thích.

Ở đây bài báo cáo sẽ trình bày 2 phương pháp dùng để khuyến nghị đó là học cộng tác(Collaborative Filtering) và nhân tố ẩn (Lantent factors).

1.3 Mô tả bộ dữ liệu

Trong bài báo cáo này cài đặt thử nghiệm với 2 bộ dữ liệu:

- ml-100k, bộ dữ liệu này có chứa:
  - Bộ 100,000 ratings (1-5) của 943 users trên 1682 bộ phim.
  - Mỗi user đánh giá ít nhất 20 bộ phim.
  - Dữ liệu rating chứa trong 2 files là u.data và u.item
- ml-1m, bô dữ liệu này có chứa:
  - 100,000,54 ratings (1-5) của 71567 users trên 10681 bộ phim.
  - Mỗi user đánh giá ít nhất 20 bộ phim.
  - Dữ liệu rating chứa trong 3 files là moives.dat , ratings.dat và tags.dat

Bộ dư trện có thể ở: moivelens. Bộ dữ liệu trên đã được trường đại học Minnesota và bất kì nhà nghiên cứu nào có thể đảm bảo tính đúng đắn của dữ liệu, tính phù hợp của nó cho bất kỳ mục đích cụ thể nào, hoặc tính hợp lệ của các kết quả dựa trên việc sử dụng bộ dữ liệu.của nó cho bất kỳ mục đích cụ thể nào, hoặc tính hợp lệ của các kết quả dựa trên việc sử dụng bộ dữ liệu.

Cấu trúc nội dung và cách sử dụng các files trong bộ dữ liệu.

1.3.1 User Ids

Người sử dụng của Movielens đã được lựa chọn ngẫu nhiên để đưa vào. Id của họ đã được ẩn danh.

1.3.2 Cấu trúc file Ratings

Tất cả các rating đều được chứa trong tệp xếp ratings.dat. Mỗi dòng của tệp này đại diện cho một rating của một bộ phim của một người dùng và có định dạng sau:

UserID::MovieID::Rating::Timestamp

Rating được thực hiện theo thang điểm 5 sao với số gia tăng nửa sao. Mẫu thời gian biểu thị bằng giây từ nửa đêm(UTC) ngày 1 tháng 1 năm 1970.

1.3.3 Cấu trúc file Moives

Thông tin phim có trong tệp moives.dat. Mỗi dòng của tệp này đại diện cho một bộ phim và có định dạng sau:

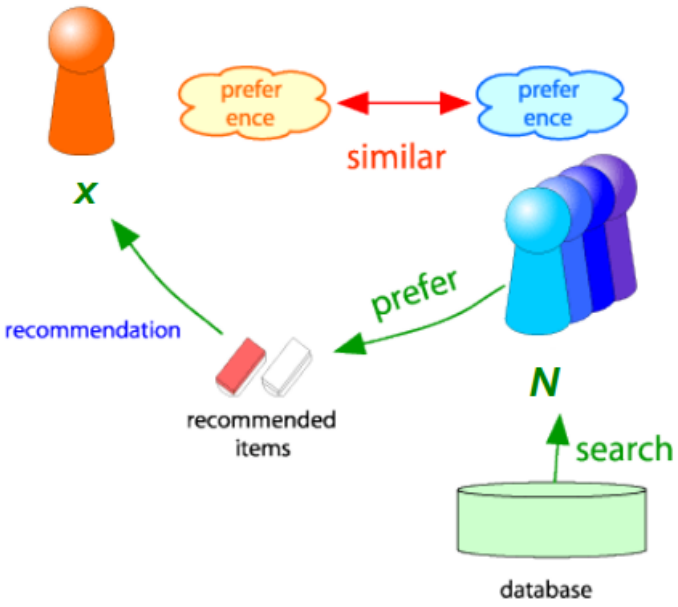
MovieID::Title::Genres

MovieID là id MovieLens thực. Tiêu đề phim, theo chính sách, phải được nhập giống với những bài tìm thấy trong IMDB, kể cả năm phát hành. Tuy nhiên, chúng được nhập bằng tay, do đó, lỗi và không nhất quán có thể tồn tại. Các thể loại là một danh sách được cách ly bằng đường ống, và được chọn từ các loại sau: **Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western**

Chương 2

Collaborative Filtering

2.1 Ý tưởng của phương pháp



- quan sát một người dùng x.
- Tìm tập N các người dùng khác cũng rating giống như các ratings của người dùng x.
- Ước lượng các ratings của người dùng x dựa trên các ratings của những người trong tập N.

2.2 Tìm tập người sử dụng tương đồng

Mỗi người dùng sẽ có một vector ratings biểu diễn đánh giá của người đó về các bộ phim mà họ đã xem ví du: $r_x = [*, \_, \_, *, * * *]$. Để tìm sử tương đồng giữa các người sử dụng với nhau chúng ta có thể sử dụng một số độ đo như:

- Độ đo tương đồng jaccard

- Vấn đề: khi sử dụng độ đo jaccard nó sẽ không quan tâm các giá trị ratings của người dùng

- Độ đo tương đồng Cosine
  - $sim(x, y) = \cos(\vec{r}_x, \vec{r}_y) = \frac{\vec{r}_x . \vec{r}_y}{\|\vec{r}_x\| . \|\vec{r}_y\|}$
  - Vấn đề: khắc phục được nhược điểm của jaccard nhưng lại bỏ qua các ratings không tốt(người dùng đánh giá thấp bộ phim đó)

- Sử dụng hệ số tương phản cá nhân
  - S_{xy} là tập các bộ phim được rating bởi x và y.
  - $sim(x, , y) = \sum s \in S \downarrow xy \uparrow (r \downarrow xs - r \downarrow x) r \downarrow ys - r \downarrow y) / \sqrt{\sum s \in S \downarrow xy \uparrow (r \downarrow xs - r \downarrow x) \uparrow 2}$

Khi cài đặt thư nghiêm sử dụng độ đo cosine. Có ví dụ sau:

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	5	5	
D		3					3

- Ta muốn: $sim(A, B) > sim(A, C)$
- Độ đo jaccard: $1/5 < 2/4$
- Độ đo Cosine: $0.386 > 0.322$
  - Mất đi sự xem xét các ratings bị người dùng đánh giá thấp.
  - Để giải quyết vấn đề này chúng ta sẽ trừ đi trung bình rating của hàng.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

Khi đó: $sim(A, B) = 0.092 > -0.559$

2.3 Dự đoán rating

Từ tập số liệu gợi ý ta có:

- Cho r_x là vector người dùng(những bộ phim mà x đã đánh giá có giá trị từ 1-5 còn chưa có giá trị là 0).
- Cho N là tập k người dùng có sở thích gần giống với x(x là người mà cần gợi ý phim)

Đánh giá theo cách nhìn Item-Item:

- Cho bộ phim i, tìm các bộ phim khác tương tự như i.
- Ước lượng ratings cho bộ phim i dựa trên ratings của các bộ phim tương tự.
- Sử dụng Công thức sau:

$$r_{xi} = \frac{\sum_{j \in N(r,x)} s_{ij} . r_{xj}}{2}$$

Trong đó:

- s_{ij}... là độ tương đồng giữa i và j.
- r_{xj}... là rating của x với bộ phim j.
- $N(i; x)$... tập các bộ phim được rating bởi x.

2.4 Dự đoán rating trong thực tế

Trong thực tế để có được một dự đoán chính xác hơn chúng ta thêm cả ước lượng độ lệch cho rating của người dùng với bộ phim.

2.4.1 Tổng quát:

- Rating trung bình của phim: 3.7 sao.
- Bộ phim Sixth Sense cao hơn trung bình là 0.5 sao.
- Trung bình các đánh giá của Joe về phim thấp hơn 0.2 sao.

  → Ước lượng: bộ phim Sixth Sense Joe sẽ đánh giá là 4 sao.

2.4.2 Hàng xóm lân cận(CF/NN):

- Joe không thích bộ phim Signs
- → Ước lượng cuối cùng cho bộ phim Sixth Sense là 3.8 sao.

Vì thế trong thực tế chúng ta sẽ có được ước lượng tốt nhất theo mô hình sai lệch sau:

$$\hat{r_{xi}} = b_{xi} + \frac{\sum_{j \in N(i;x)} s.(r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s}$$

Với $b_{xi} = \mu + b_x + b_i$ là độ lệch cơ sở của người dùng x khi đánh giá bộ phim i. Trong đó:

- μ là rating trung bình của tất cả người dùng.
- b_x là độ lệch của người dùng x.

  $$b_x = avg(x) - \mu.$$

- b_i là độ lệch của bộ phim i.

  $$b_i = avg(i) - \mu.$$

2.5 Ưu và nhược điểm của phương pháp

2.5.1 Ưu điểm:

- Làm việc với bất kì loại phim không cần quan tâm đến đặc điểm lựa chọn.
- Dễ dàng cài đặt.

2.5.2 Nhược điểm:

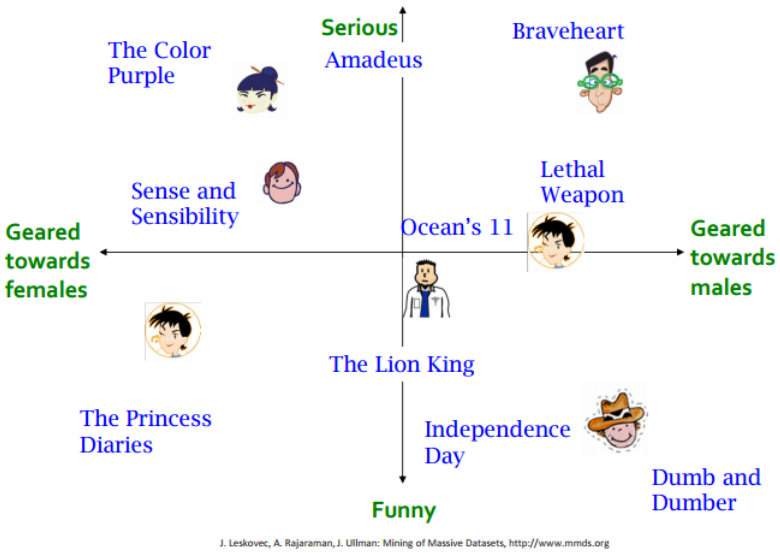
- Khi chạy cần có đủ user trên hệ thống.
- Tính thưa:
  - dữ liệu là một ma trận thưa.
  - Khó có thể tìm thất những người dùng có ratings tương tự nhau.
- Không thể gợi ý những phim mà chưa được ai đánh giá trước đó.

Chương 3

Latent Factor Models

3.1 Ý tưởng của phương pháp

Trong thức tế mỗi người dùng sẽ có những sở thích riêng của mình ví dụ như có người thích phim tình cảm, lãng mạn có người thích phim hành động, khoa học viễn tưởng, ... Những sở thích đó được coi là một nhân tố ẩn của mỗi người dùng.



Hình 3.1: hình ảnh về factors

- Các ratings chịu ảnh hưởng sâu sắc bởi một bộ các nhân tố(factor) rất cụ thể cho miền.
- Một số các factors là rất khó quan sát được và khó ước lượng sự tác động của chúng đối với ratings của người dùng.
- Mục tiêu là suy luận những nhân tố tiềm ẩn từ dữ liệu đánh giá bằng cách sử dụng các kỹ thuật toán học.

3.2 SVD

SVD là một phương pháp trong toán học dùng để giảm chiều ma trận làm sao cho sự mất mát thông tin là ít nhất.

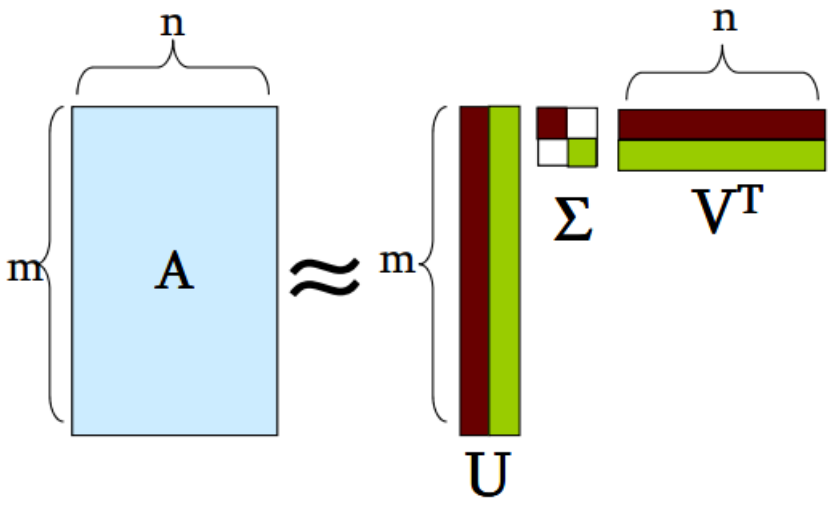
Giả sử ta có ma trận A_{n*p} ta có:

$$A_{n*p} = U_{n*n} S_{n*p} V^T_{p*p}$$

Trong đó:

- U là ma trận với các cột là các vector trái.
- S là ma trận cùng kích thước như A giá trị đợn là đường chéo.
- V^T có các hàng là cá vector đơn phải

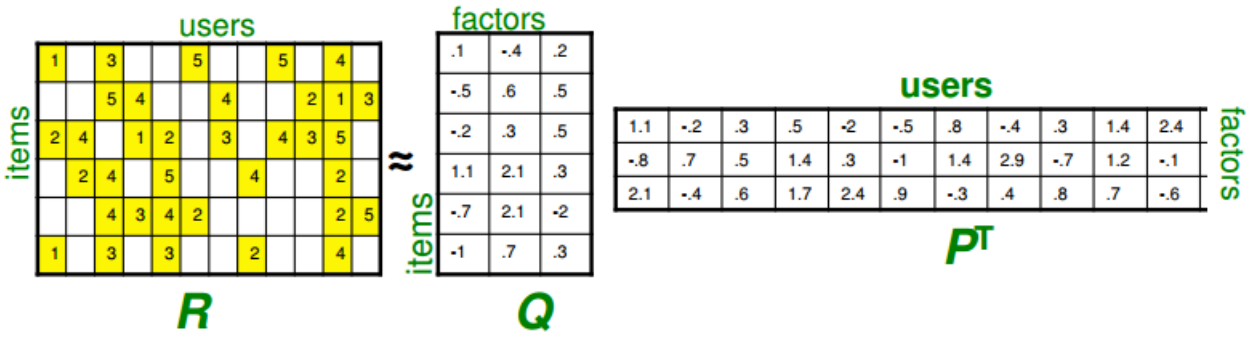
SVD thể hiện sự mở rộng của dữ liệu ban đầu trong một hệ tọa độ nơi ma trận hiệp phương sai là đường chéo.



Hình 3.2: Hình ảnh minh họa svd

3.3 Dự đoán rating

Để dự đoán rating người dùng ban đầu ta có ma trận người dùng sử dụng svd để giảm chiều mà trận và phân tách thành ma trận trận là mat trận người dùng và ma trận user



Hình 3.3: ma trận sau khi giảm chiều sử dụng svd

Trong đó:

- A = R là ma trận người dùng ban đầu.
- Q là ma trận nhân tố của phim.
- P^T là ma trận nhân tố của người dùng.

$$P^T = \sum V^T$$

Để dự đoán rating của người dùng x cho một bộ phim ra chỉ cần tính:

$$\hat{r_{xi}} = q_i . p_x = \sum_f q_{if} . p_{xf}$$

Trong đó:

- q_i là hàng i của ma trận Q.
- p_x là cột x của ma trận P.
- f là số factors(hay độ dài vector của hàng i và cột x).

3.4 Tối thiểu hóa lỗi RMSE

Như đã biết SVD đã tối thiểu hóa lỗi trên toàn ma trận để ít mất mát thông tin nhất.Chúng ta có thể xây dựng lại hàm lỗi:

$$min_{U,V,\sum} \sum (A_{ij} - [U \sum V_T]_{ij})^2$$

Bởi vì ma trận người dùng là thưa nên có rất nhiều rating bằng không vì thế cần tối thiểu RMSE cho các dữ liệu không nhìn thấy.

Ý tưởng: tối thiểu hóa RMSE trên tập dữ liệu.

- k (số nhân tố) bắt tất cả các đặc điểm người dùng.
- Nhưng RMSE trên bộ test bắt đầu với k $>$ 2

Vấn đề Overfitting:

- Với quá nhiều tham số tự do thì mô hình sẽ trở nên fitting.

- Để giải quyết vấn đề overfitting không cho P,Q quá tự do học.

$$\min_{P,Q} \sum_{training} \underbrace{(r_{xi} - q_i p_x)^2}_{error} + \underbrace{[\lambda_1 \sum_x \|p_x\|^2 + \lambda_2 \sum_i \|q_i\|^2]}_{length}$$

Ở đây:

- λ_1 và λ_2 là các hệ số phạt của p và q

- Sử dụng gradient descent để tối thiểu hóa lỗi của hàm trên

3.5 Mở rộng mô hình latent với độ lệch

Để có một ước lượng chính xác hơn trong thực tế chúng ta có thể thêm vào độ lệch rating của người dùng và của phim.Khi đó ta có ratings của người dùng x sẽ đánh giá bộ phim i như sau:

$$r_{xi} = \underbrace{\mu}_{overallmeanrating} + \underbrace{b_x}_{biasforuserx} + \underbrace{b_i}_{biasformoivei} + q_x.p_i$$

3.5.1 Vấn đề tối ưu hóa lỗi và overfitting:

Giải pháp:

$$\min_{Q,P} \sum_{(x,i)\in R} (r_{xi} - (\mu + b_x + b_i + q_i.p_x))^2$$

$$+(\lambda_1 \sum_i \|q_i\|^2 + \lambda_2 \sum_x \|p_x\|^2 + \lambda_3 \sum_x \|b_x\|^2 + \lambda_4 \sum_i \|b_i\|^2)$$

Trong đó:

- $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ là các hệ số phạt.

  + Để đơn giản ở đây tất cả dùng chung λ

- b_i, b_x là độ lệch của moive và người dùng.

- μ là raitng trung bình của ma trận người dùng.

- q_i, p_x là các vector hàng và cột của ma trận Q,P.

3.5.2 Sử dụng Stochastic Gradient Descent

Xét người dùng x và bộ phim i. Giả sử đạo hàm riêng phần theo q_{if} Ta có:

$$\nabla Q = [\nabla q_{if}] \; and \; \nabla q_{if} = \sum_{x,i} -2(r_{xi} - q_i p_x) p_x f + 2\lambda q_{if}$$

- Ở đây $q_i f$ nhân tố f của hàng q_i của matrix Q

Cập nhật trọng số cho Q theo Stochastic Gradient Descent

$$Q = Q - \mu \nabla Q(r_{xi})$$

Cập nhật tổng quát:

$$\varepsilon_{xi} = 2 * (r_{xi} - \mu B_x - B_i - P_x^T Q_i)$$
$$P_x = P_x + \alpha * (\varepsilon_{xi} Q_i - \lambda P_x)$$
$$Q_i = Q_i + \alpha * (\varepsilon_{xi} P_x - \lambda Q_i)$$
$$B_x = B_x + \alpha * (\varepsilon - \lambda B_x)$$
$$B_i = B_i + \alpha * (\varepsilon_{xi} - \lambda B_i)$$

3.6 Ưu và nhược điểm của phương pháp

3.6.1 Ưu điểm:

- Dễ triển khai, hiểu và sử dụng. Có rất nhiều hiện thực và khả năng mở rộng có sẵn.
- Thời gian chạy: Vì nó chỉ liên quan đến phân rã ma trận tài liệu, nó nhanh hơn so với các mô hình giảm kích thước khác.
- Áp dụng nó trên dữ liệu mới là dễ dàng hơn và nhanh hơn so với các phương pháp khác.

3.6.2 Nhược điểm:

- Bởi vì nó là một mô hình phân phối, do đó, không phải là một đại diện hiệu quả, khi so sánh với các phương pháp hiện đại (như mạng nơ-ron sâu).
- Đó là một mô hình tuyến tính, do đó, không phải là giải pháp tốt nhất để xử lý các phụ thuộc phi tuyến tính.

Chương 4

Kết quả thực nghiệm

4.1 Xử lí dữ liệu

Dư liệu được lấy từ trang moivelen với các rating thực của người dùng thực trong hệ thống phim của moivelen. Bộ dữ liệu được chia làm 5 folds phụ vụ cho việc đánh giá Cross-Validation.
Running split_ratings.sh will use ratings.dat as input, and produce the fourteen output files described below. Multiple runs of the script will produce identical results.

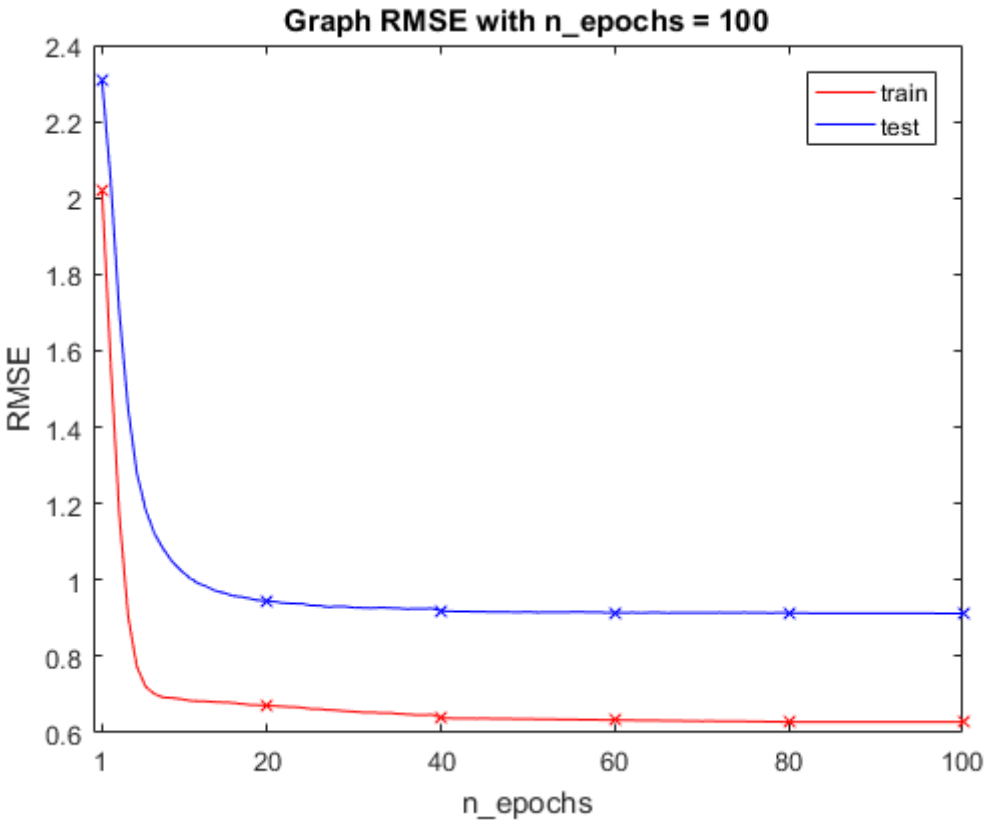
File Names	Mô tả
r1.train, r2.train, r3.train, r4.train, r5.train r1.test, r2.test, r3.test, r4.test, r5.test	Các bộ dữ liệu r1.train và r1.test qua r5.train và r5.test là 80% / 20% phân chia dữ liệu xếp hạng thành dữ liệu huấn luyện và kiểm tra. Mỗi của r1, . . . , r5 có các tập kiểm tra rời rạc Điều này nếu để xác nhận chéo 5 lần
ra.train, rb.train ra.test, rb.test	Tập dữ liệu ra.train, ra.test, rb.train và rb.test chia dữ liệu xếp hạng thành tập huấn luyện và tập kiểm tra với chính xác 10 xếp hạng cho mỗi người dùng trong tập kiểm tra. Bộ ra.test và rb.test không liên kết.

4.2 Kết quả Collaborative Filtering

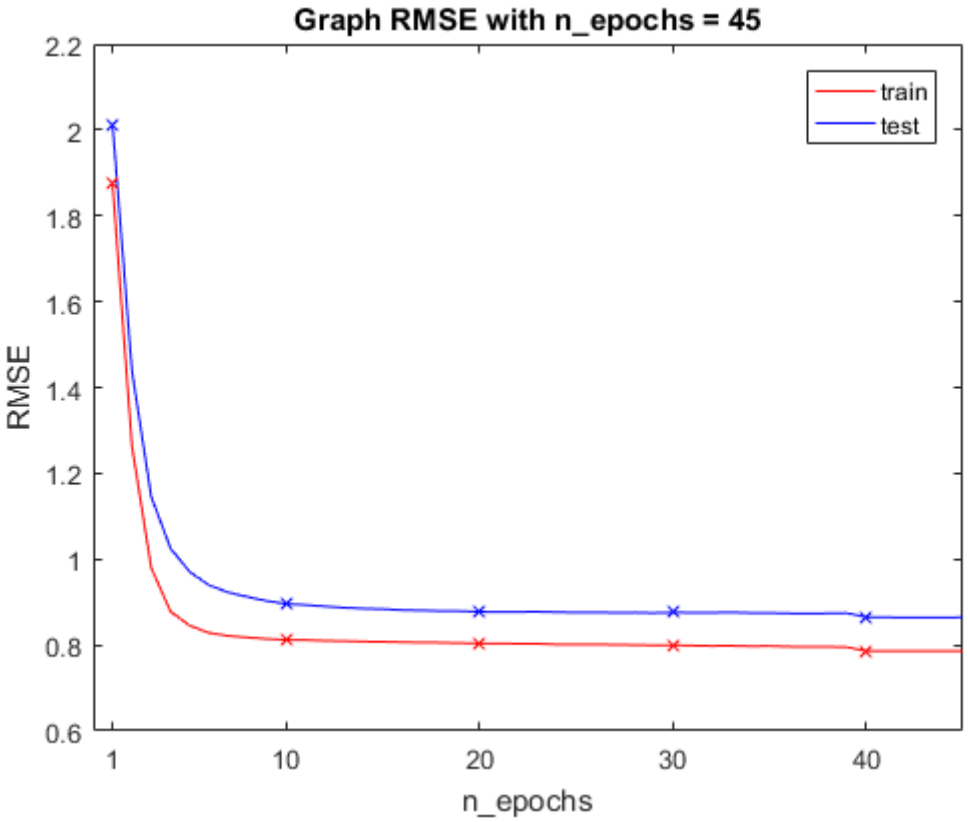
Bộ dư liệu	k(hàng xóm)	RMSE
ml-100k	5	0.9537
ml-1m	5	0.8835

4.3 Kết quả Latent factor model

Bộ dữ liệu	k(factor)	n_eopchs	RMSE
ml-100k	40	100	0.9161
ml-1m	40	45	0.8626



Hình 4.1: Đồ thị RMSE cho bộ ml-100k



Hình 4.2: Đồ thị RMSE cho bộ ml-1m

Tài liệu tham khảo

[1] http://www.mmds.org/

[2] Slide chapter 9 Stanford University

[3] Mining of Massive Datasets of Jure Leskovec Stanford Univ. Anand Rajaraman Milliway Labs Jeffrey D. Ullman Stanford Univ.