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

_____*** _____

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

MÔ HÌNH PHÂN RÃ MA TRẬN POISSON KẾT HỢP BỘ TRI THỨC TIÊN NGHIỆM CHO BÀI TOÁN HỆ GỢI Ý

Sinh viên thực hiện : **Nguyễn Văn Túc**
Lớp : **CNTT 2.4 – K59**
Giáo viên hướng dẫn: **TS. Thân Quang Khoát**

HÀ NỘI 6-2019

LỜI CẢM ƠN

Lời đầu tiên em xin được gửi lời cảm ơn chân thành đến các thầy cô giáo và các cán bộ công tác tại trường đại học Bách Khoa Hà Nội. Đặc biệt là các thầy, cô giáo thuộc Viện Công nghệ thông tin và Truyền thông đã tận tình giảng dạy và truyền đạt cho em những kiến thức bổ ích trong suốt 5 năm học vừa qua. Đồng thời em cũng xin được gửi lời cảm ơn đặc biệt đến hai thầy TS. Thân Quang Khoát và ThS. Ngô Văn Linh, các thầy là người chỉ dẫn tận tình, chỉ dạy cho em những kinh nghiệm quý báu để em có thể hoàn thành đồ án tốt nghiệp này. Và luôn động viên, giúp đỡ em trong những thời điểm khó khăn.

Em xin gửi lời cảm ơn chân thành đến các thầy cô Phòng nghiên cứu Khoa học dữ liệu thuộc Viện Công Nghệ Thông Tin và Truyền Thông đã tạo điều kiện cho em sinh hoạt, trao đổi cũng như tạo môi trường học tập, nghiên cứu trong quãng thời gian thực hiện đồ án.

Em cũng xin cảm ơn anh Nguyễn Đức Anh đã hỗ trợ em trong việc thực hiện và hoàn thành đề tài này.

Cuối cùng em xin gửi lời cảm ơn tới gia đình và bạn bè. Lời động viên tinh thần từ gia đình và bạn bè luôn luôn là nguồn động lực để em tiến lên phía trước.

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Với sự phát triển ngày càng mạnh mẽ của các lĩnh vực thương mại điện tử, các giao dịch trực tuyến diễn ra ngày càng nhiều và phổ biến. Theo đó, với một số lượng cực kỳ lớn thông tin trên internet, người dùng cần biết chọn lọc ra những thông tin phù hợp với nhu cầu và sở thích của mỗi cá nhân. Bài toán gợi ý ra đời nhằm giải quyết vấn đề này, một hệ thống với một cơ chế gợi ý hợp lý sẽ thúc đẩy sự tương tác của người dùng đối với hệ thống bằng các gợi ý các sản phẩm, dịch vụ hợp lý.

Trong thực tế, một hệ thống gợi ý hợp lý sẽ giúp tiết kiệm thời gian của người dùng và tăng sự hài lòng của người dùng khi sử dụng hệ thống. Cho đến nay đã có rất nhiều nghiên cứu khác nhau đã đưa ra những mô hình gợi ý và được áp dụng rộng rãi trên nhiều lĩnh vực như: các website thương mại điện tử, trang web tin tức trực tuyến … Tuy nhiên đa số đều chỉ quan tâm đến các lịch sử tương tác giữa người dùng và sản phẩm để đưa ra gợi ý. Dựa trên góc nhìn của người dùng, đa số người dùng tiếp xúc với sản phẩm trước tiên bởi tên và một số mô tả ngắn liên quan đến sản phẩm, ví dụ khi ta trước tiên đọc chi tiết một bài báo điều đầu tiên là phần tiêu đề hoặc thêm mô tả ngắn ở dưới, tương tự trước khi nhấn xem một sản phẩm cũng vậy. Do việc có rất nhiều sản phẩm và bài báo trên internet, thói quen sử dụng như vậy gần như sẽ giống với đại đa số người dùng. Do đó ta mong muốn có thể tận dụng được thông tin mô tả ngắn này giúp tăng chất lượng của hệ thống gợi ý.

Vì vậy, trong đồ án này em tìm hiểu và thử nghiệm mô hình học phân rã ma trận Poisson kết hợp với bộ tri thức tiên nghiệm để khai thác thông tin mô tả ngắn của sản phẩm. Ý tưởng chính của mô hình này là sử dụng phân rã ma trận Poisson để mô hình hóa các tương tác rời rạc và sử dụng bộ tri thức tiên nghiệm từ biểu diễn nhúng của từ sử dụng trong biểu diễn thông tin cho sản phẩm. Đồng thời khảo sát chất lượng của mô hình này đối với một số mô hình gợi ý sử dụng phân rã ma trận mới nhất.

Mục lục

Danh sách các từ viết tắt và thuật ngữ

CF	Collaborative filtering
Factorization	Phân rã
MF	Matrix factorization
WMF	Weighted Matrix Factorization
HPF	Hierarchical Poisson Factorization
Poisson Matrix Factorization using Word Embedding Prior	Phân rã ma trận Poisson kết hợp sử dụng tri thức tiên nghiệm từ nhúng
Feedforward Neural Network	Mạng Neural truyền thẳng
Multinomial Distribution	Phân phối đa thức
Dropout rate	Tỷ lệ loại bỏ
Prior	Tiên nghiệm
Word Embedding	Biểu diễn từ nhúng
Variational Inference	Suy diễn biến phân
Variational distribution	Phân phối biến phân
Content	Nội dung, mô tả của sản phẩm
Precision@K	Độ chính xác của mô hình tại top K
Recall@K	Độ bao phủ của mô hình tại top K
Gaussian distribution	Phân phối Gaussian
Poisson distribution	Phân phối Poisson
Gamma distribution	Phân phối Gamma

Danh sách các ký hiệu sử dụng

U	Số lượng người dùng
I	Số lượng sản phẩm
K	Số lượng thuộc tính ẩn biểu diễn cho mỗi người dùng, sản phẩm
V	Kích thước từ vựng
R	Ma trận tương tác người dùng sản phẩm
r_{ui}	Giá trị tương tác giữa người dùng u và sản phẩm i
E	Ma trận biểu diễn nhúng của từ
θ	Ma trận thuộc tính người dùng
θ_u	Vector K chiều biểu diễn thuộc tính của người dùng u
β	Ma trận thuộc tính sản phẩm
β_i	Vector K chiều biểu diễn thuộc tính sản phẩm i

Danh sách các hình vẽ

Danh mục các bảng

Chương 1: Giới thiệu đề tài

1.1 Đặt vấn đề

Chúng ta đang sống trong cuộc cách mạng công nghiệp lần thứ 4, cuộc cách mạng của Công nghệ thông tin. Trong thời kỳ mà sự bùng nổ và phát triển vô cùng nhanh chóng của các lĩnh vực liên quan đến máy tính điện tử, Internet đặc biệt là các lĩnh vực liên quan đến thương mại điện tử. Điều này có nghĩa là tất cả các sản phẩm, dịch vụ được trưng bày một cách dễ dàng tới người dùng Internet. Nhìn qua ta có thể thấy đó là một điều vô cùng tiện lợi, tuy nhiên với một lượng thông tin vô cùng lớn như vậy người dùng cần phải biết tìm kiếm, chọn lọc ra đâu là những thông tin phù hợp với mình. Xuất phát từ đó, hệ gợi ý ra đời nhằm phục vụ cho nhu cầu này của người sử dụng Internet.

Hệ thống gợi ý (Recommendation System) là một hệ thống có khả năng gợi ý các sản phẩm mà nó cho rằng phù hợp đối với người dùng dựa trên một số tiêu chí nào đó. Ví dụ, dựa trên lịch sử giao dịch người dùng hoặc những người dùng có những hành vi tương tự, hệ gợi ý sẽ tính toán và đưa ra dự đoán về độ phù hợp của sản phẩm nào đó và dựa trên những kết quả đó đưa ra những gợi ý cho người dùng. Một hệ thống gợi ý tốt sẽ giúp tăng hiệu quả tương tác giữa người dùng và kéo theo đó là khả năng thu hút người dùng tiếp tục sử dụng hệ thống, tăng hiệu quả marketing …

Hệ gợi ý là một lĩnh vực đang thu hút rất nhiều sự chú ý trong nghiên cứu và ngoài doanh nghiệp. Do những đóng góp vô cùng quan trọng trong các hệ thống thương mại điện tử nên ta có thể dễ dàng bắt gặp các hệ thống gợi ý như:

- Youtube tự động gợi ý các video liên quan, gợi ý các video mà người dùng có thể thích, có thể xem tiếp theo.
- Facebook gợi ý kết bạn, hiển thị quảng cáo …
- Hệ thống gợi ý phim, videos: Netflix, MovieLens, MyClip.vn …
- Gợi ý sản phẩm, dịch vụ: Amazon.com, TIKI …

1.2 Bài toán hệ gợi ý

Các thành phần cơ bản của một bài toán gợi ý bao gồm:

- Một lượng hữu hạn U người dùng và I sản phẩm.
- Các thông tin về người dùng: hồ sơ cá nhân, tuổi, giới tính, sở thích…
- Các thông tin về sản phẩm: tên, mô tả, danh mục, thể loại…
- Ma trận tương tác R giữa người dùng và sản phẩm, các tương tác ở đây là các hành động của người dùng đối với sản phẩm: xem, click, mua, hoặc đánh giá sản phẩm …

Các hướng tiếp cận để xây dựng một bài toán gợi ý:

- Lọc cộng tác (collaborative filtering)
- Lọc dựa theo nội dung (content-based)
- Phương pháp kết hợp (hybrid)

Content-based: Hệ gợi ý dựa trên nội dung, đối với cách tiếp cận này để xây dựng được mô hình cho hệ gợi ý ta cần tạo một bộ hồ sơ cho mỗi người dùng hoặc sản phẩm có trong hệ thống. Giả sử khi muốn gợi ý một sản phẩm cho một người dùng u, hệ thống sẽ dựa trên đặc điểm của các sản phẩm mà đã được u đánh giá cao, từ đó gợi ý ra những sản phẩm tương đồng với sở thích của u. Nhược điểm của phương pháp này nằm ở việc xây dựng hồ sơ cho mỗi người dùng sản phẩm không phải lúc nào cũng là dễ dàng, nhưng nhờ vậy nó có thể giúp giải quyết vấn đề về người dùng mới hoặc sản phẩm mới.

Collaborative filtering: Ở phương pháp này, mô hình chỉ sử dụng thông tin từ ma trận tương tác giữa người dùng và sản phẩm. Lọc cộng tác dựa vào phản hồi trong quá khứ của người dùng mà không cần phải tạo ra hồ sơ cho người dùng và sản phẩm như đối với phương pháp dựa trên nội dung.

Phương pháp kết hợp (hybrid method): Phương pháp này được sinh ra với mục đích khắc phục nhược điểm của hai phương pháp trên bằng cách vừa sử dụng ma trận tương tác giữa người dùng và sản phẩm đồng thời kết hợp với thông tin mô tả của sản phẩm để cải thiện chất lượng gợi ý cho mô hình.

Từ đó, trong đồ án này em tìm hiểu mô hình phân rã ma trận Poisson sử dụng tri thức tiên nghiệm biểu diễn nhúng của từ vựng cho các sản phẩm (Poisson Matrix Factorization using Word Embedding Prior – PFEP), trong đó tri thức tiên nghiệm biểu diễn nhúng của các từ được đưa vào một mạng neural truyền thẳng giúp cho việc biểu diễn sản phẩm trước khi đưa vào phân rã ma trận. Một số đặc điểm chính:

- Sử dụng phân rã Poisson phù hợp biểu diễn tương tác rời rạc của hệ gợi ý.
- Kết hợp tri thức tiên nghiệm của biểu diễn nhúng của từ giúp tăng thông tin cho biểu diễn sản phẩm.
- Kết hợp việc lọc loại bỏ và học ngẫu nhiên bằng cách ngẫu nhiên loại bỏ đi các tương tác của mỗi người dùng giúp tăng tốc độ và chất lượng của mô hình.

1.3 Bố cục đồ án

Nội dung đồ án tốt bao gồm các phần:

Chương 1: Giới thiệu đề tài.

Chương 2: Trình bày cơ sở lý thuyết và những kiến thức liên quan đến việc xây dựng các mô hình gợi ý.

Chương 3: Trình bày các mô hình WMF, HPF, CTMP cùng với ưu, nhược điểm của các mô hình.

Chương 4: Trình bày, phân tích mô hình phân rã ma trận Poisson kết hợp tri thức từ nhúng.

Chương 5: Trình bày kết quả thử nghiệm, đánh giá mô hình và phân tích hiệu quả, ưu nhược điểm của các mô hình.

Chương 6: Kết luận và tóm tắt kết quả đạt được trong đồ án.

Chương 2: Cơ sở lý thuyết

Chương này em tập trung giới thiệu về một số kiến thức liên quan được sử dụng trong đồ án, bao gồm:

- Hệ gợi ý và các hướng tiếp cận hệ gợi ý.
- Một số phân phối xác suất được sử dụng.
- Mô hình đồ thị xác suất.
- Phương pháp sử dụng suy diễn biến phân.
- Phân rã ma trận.
- Dropout và sử dụng tri thức tiên nghiệm.

2.1 Hệ gợi ý và các hướng tiếp cận

2.1.1 Khái niệm hệ gợi ý

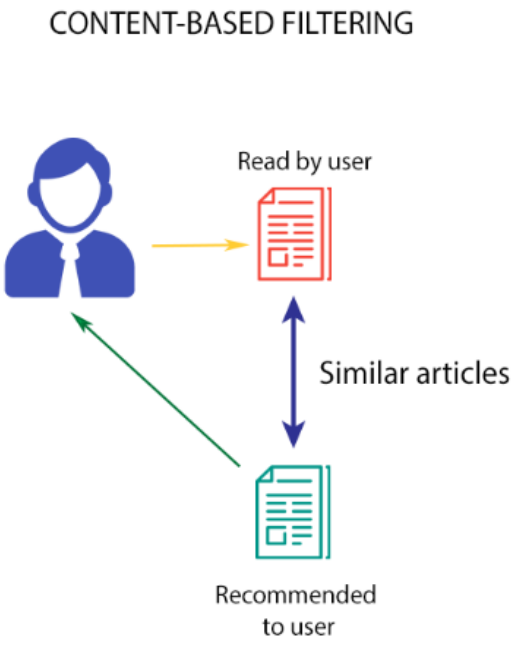
Hệ gợi ý (Recommender System) là một thành phần trong hệ thống thông tin. Dựa trên một số các tiêu chí được xây dựng từ trước mà một hệ gợi ý có khả năng dự đoán đánh giá của người dùng đối với sản phẩm nào đó, đồng thời dựa vào đó để đưa ra những chiến lược gợi ý phù hợp nhất đối với người dùng. Một hệ thống gợi ý càng phù hợp sẽ giúp tiết kiệm thời gian của người dùng và tăng tương tác giữa người dùng đối với hệ thống. Có hai đối tượng chính trong hệ gợi ý là người dùng (users) và sản phẩm (items).

Hai hướng tiếp cận truyền thống để xây dựng hệ gợi ý: Lọc dựa trên nội dung (Content-based filtering) và lọc cộng tác (Collaborative filtering).

2.1.2 Hệ gợi ý dựa trên lọc nội dung (Content-based Filtering)

Hệ gợi ý xây dựng dựa trên lọc nội dung [3] là hệ thống dựa trên nội dung của mỗi người dùng hoặc sản phẩm. Do đó chúng ta cần xây dựng một bộ hồ sơ cho mỗi người dùng hoặc sản phẩm bằng cách biểu diễn chúng dưới dạng một vector đặc trưng. Trong một số trường hợp đơn giản, vector đặc trưng này có thể được trích xuất trực tiếp từ sản phẩm như tên, mô tả của nó. Từ đó việc gợi ý sản phẩm cho một người dùng u bằng cách đưa ra những sản phẩm có độ tương đồng cao với những sản phẩm mà u có tương tác tốt trước đó (được mua, được thích, được đánh giá cao…). Phương pháp này có thể gợi ý cho người dùng hoặc sản phẩm mới vì nó chỉ cần quan tâm đến biểu diễn hồ sơ cho mỗi đối tượng, điều này cũng chính là nhược điểm khi việc xây dựng hồ sơ cho tất cả các đối tượng trong một hệ thống không phải là điều dễ dàng có thể thực hiện không kể đó là một hệ thống lớn và liên tục phát sinh các người dùng, sản phẩm mới.

Ví dụ: một người dùng u thường đọc rất nhiều các bài báo tin tức liên quan đến tin tức chính trị, biển đông vậy thì một hệ thống gợi ý hợp lý sẽ tìm những bài báo có nội dung liên quan đến những vấn đề như vậy để gợi ý cho người dùng. Hình 1 mô tả cách tiếp cận xây dựng hệ thống dựa trên lọc nội dung.



Hình 1: Hướng tiếp cận Content-based trong hệ gợi ý
(Nguồn:http://datameetsmedia.com/an-overview-of-recommendation systems/)

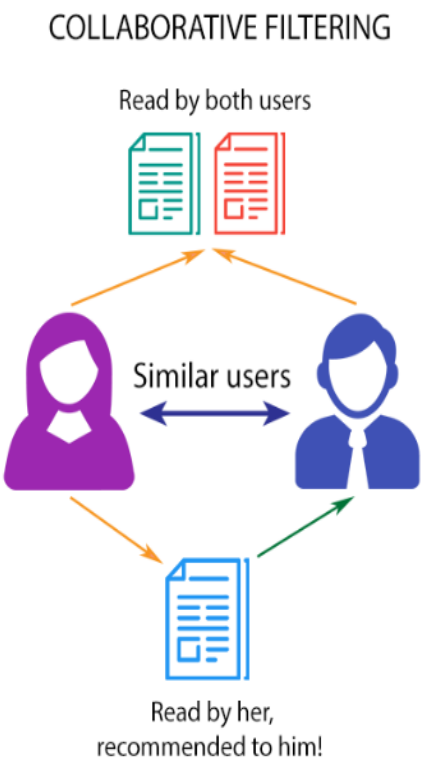
Các bước xây dựng hệ gợi ý dựa trên nội dung:

- Biểu diễn mỗi sản phẩm dưới dạng một vector đặc trưng.
- Tính toán độ tương đồng giữa các vector đặc trưng của mỗi sản phẩm có trong hệ thống.
- Gợi ý sản phẩm cho người dùng u bằng cách đưa ra các sản phẩm có độ tương đồng cao với những sản phẩm mà được thích bởi u.

2.1.3 Hệ gợi ý dựa trên lọc cộng tác (Collaborative filtering)

Do hướng tiếp cận xây dựng hệ gợi ý dựa trên lọc nội dung có nhược điểm là khó khăn khi xây dựng vector biểu diễn cho mỗi đối tượng và phương pháp này chỉ tập trung vào biểu diễn nội dung của mỗi đối tượng mà không quan tâm đến mối quan hệ, tương tác giữa các đối tượng với nhau trong một hệ thống. Từ đó một hướng tiếp cận mới để tận dụng mỗi quan hệ này là dựa trên các mối quan hệ tương tác đã có trong hệ thống được gọi là phương pháp lọc cộng tác. Một hệ thống gợi ý dựa trên phương pháp lọc cộng tác [12,13] là hệ thống không quan tâm đến thuộc tính người dùng và sản phẩm, chỉ quan tâm đến lịch sử tương tác của người dùng đối với các sản phẩm như: xem, click, mua, đánh giá …

Một sản phẩm được gợi ý đến cho một người dùng dựa trên những người dùng có hành vi tương tự. Do đó với cách tiếp cận này hệ gợi ý sẽ tận dụng được hành vi tương tác của người dùng thay vì chỉ quan tâm đến nội dung của sản phẩm như đối với cách tiếp cận dựa trên nội dung. Hình 12 mô tả phương pháp tiếp cận xây dựng hệ gợi ý dựa trên lọc cộng tác.



Hình 2: Hướng tiếp cận Collaborative Filtering trong xây dựng hệ gợi ý
(Nguồn:http://datameetsmedia.com/an-overview-of-recommendation systems/)

Xây dựng hệ gợi ý dựa trên phương pháp lọc cộng tác: Phương pháp này có hai hướng tiếp cận nhỏ hơn là User – based và Item – based.

User – based: Ý tưởng chính của cách tiếp cận này là mức độ phù hợp của user U với sản phẩm I được quyết định dựa trên mức độ phù hợp của những người có độ tương đồng cao với U đối với sản phẩm I, chú ý những người dùng có độ tương đồng cao với U lúc này đều là những người đã đánh giá I. Các bước xây dựng của một hệ thống dựa trên user- based bao gồm:

- Biểu diễn mỗi người dùng bằng một vector dựa trên các sản phẩm đã tương tác.
- Đối với mỗi người dùng U tìm ra nhóm người dùng có độ tương đồng gần nhất thông qua các vetor biểu diễn. Độ đo tương đồng được dùng phổ biến là Cosine[1].
- Độ phù hợp của người dùng U với sản phẩm I được tính dựa trên nhóm K người dùng có độ tương đồng cao nhất đối với U và cùng đã đánh giá I.

Item – based: Tương tự như với cách tiếp cận User- based, để dự đoán mức độ phù hợp của người dùng U với sản phẩm I hệ thống sẽ dựa trên đánh giá của người dùng U với những sản phẩm có độ tương đồng cao đối với sản phẩm I.

[1] https://en.wikipedia.org/wiki/Cosine_similarity

2.2 Một số phân phối xác suất được sử dụng

2.2.1 Phân phối Gauss (Gauss distribution)

Phân phối Gauss là một phân phối của biến ngẫu nhiên liên tục, một biến ngẫu nhiên liên tục X tuân theo phân phối Gauss với kỳ vọng μ và phương sai δ^2, sẽ có hàm mật độ xác suất:

$$f(x|\mu,\delta^2) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(x-\mu)^2}{2\delta^2}}$$

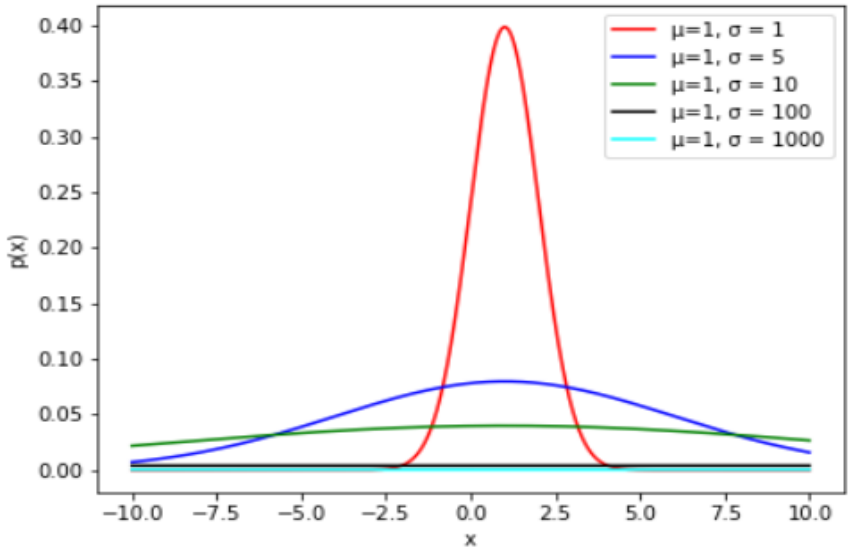
Ký hiệu: X ~ N(μ,δ^2). Đặt $\delta^2 = \frac{1}{c}$ ta có:

$$f(x|\mu,\delta^2) = \frac{\sqrt{c}}{\sqrt{2\pi}} e^{-c\frac{(x-\mu)^2}{2}}$$

Tính chất của các phân phối Gauss:

- Các giá trị sinh bởi phân phối Gauss tập trung quanh kỳ vọng của nó. Minh họa trực tiếp trên hình 3 ta thấy phân phối Gauss có dạng đối xứng quanh giá trị kỳ vọng μ, phương sai δ^2 càng lớn thì mức độ biến động quanh giá trị kỳ vọng càng lớn và ngược lại. Coi c là độ tin cậy với giá trị $c = \frac{1}{\delta^2}$ nếu độ tin cậy c càng nhỏ (tương ứng với phương sai lớn) thì mức độ biến thiên của biến ngẫu nhiên giao động mạnh quanh kỳ vọng và ngược lại.
- Kỳ vọng: $\mathrm{E}(\mathrm{X}) = \mu$
- Phương sai: $\mathrm{Var}(\mathrm{X}) = \sigma^2$

Hình 3 minh họa phân phối Gauss với kỳ vọng và giá trị phương sai khác nhau.



Hình 3: Phân phối Gauss với kỳ vọng μ = 1 và độ lệch chuẩn khác nhau

2.2.2 Phân phối Poisson (Poisson distribution)

Là phân phối xác suất rời rạc biểu thị xác suất của số lần xảy ra sự kiện trong một khoảng thời gian khi biết trong cùng khoảng thời gian này sự kiện đó xảy ra với kỳ vọng không đổi.

Cụ thể, một biến ngẫu nhiên rời rạc X nào đó, nếu giá trị kỳ vọng (hay số lần trung bình mà biến ngẫu nhiên đó xảy ra trong khoảng thời gian đó) là λ, thì xác suất cũng để chính sự kiện đó xảy ra k lần (k là số tự nhiên) được tính bởi:

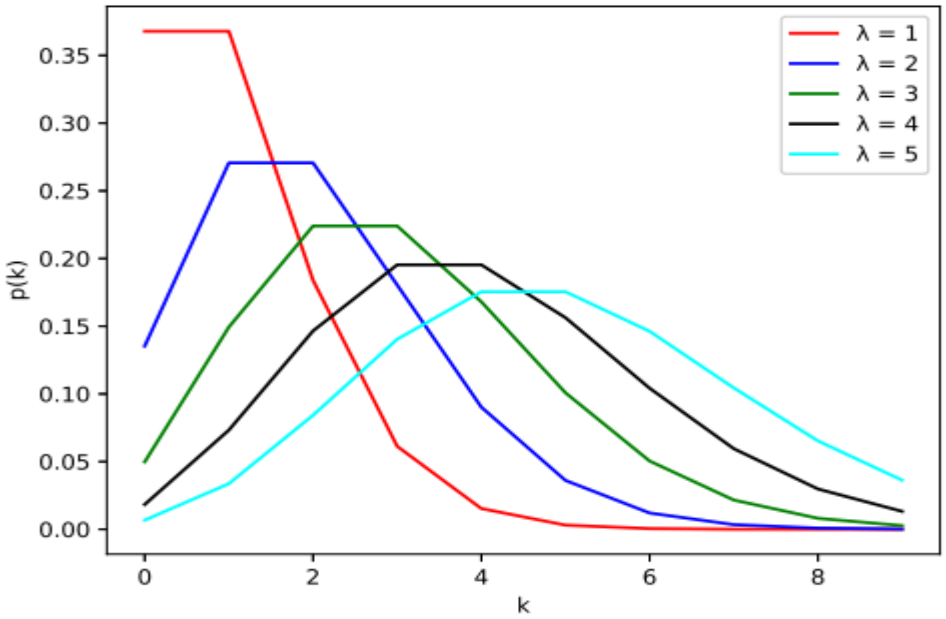
$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Ký hiệu: X ~ Poisson(λ)

Tính chất của các phân phối Poisson:

- Phân phối biểu diễn cho các biến ngẫu nhiên rời rạc.
- E(X) = Var(X) = λ

Hình 4 mô tả phân phối Poisson với các giá trị kỳ vọng khác nhau.



Hình 4: Phân phối Poisson với kỳ vọng λ khác nhau

2.2.3 Phân phối Gamma (Gamma distribution)

Một biến ngẫu nhiên liên tục tuân theo phân phối Gamma với hai tham số γ_{shp} và γ_{rte} nếu hàm mật độ xác suất được tính bởi công thức:

$$f(x;\gamma_{shp},\gamma_{rte}) = \frac{{\gamma_{rte}}^{\gamma_{shp}} . x^{\gamma_{shp}-1} . e^{-\gamma_{rte}x}}{\Gamma(\gamma_{shp})}$$

Với x > 0, $\gamma_{shp} > 0$, $\gamma_{rte} > 0$

Và hàm gamma được định nghĩa:

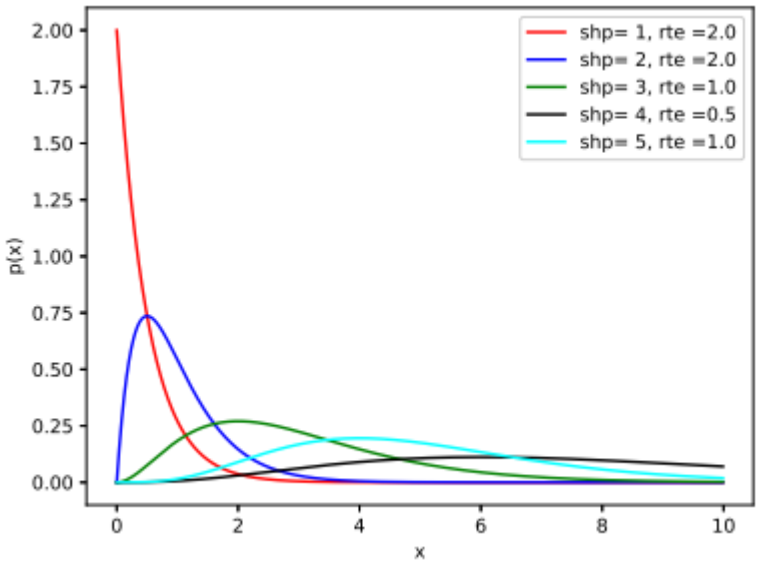
$$\Gamma(\gamma_{shp}) = \int_0^{\infty} x^{\gamma_{shp}-1} . e^{-x} dx$$

Ký hiệu: X ~ Gamma($\gamma_{shp}, \gamma_{rte}$)

Tính chất của các phân phối Gamma:

- E(X) = $\frac{\gamma_{shp}}{\gamma_{rte}}$
- E(ln(X)) = $\Psi(\gamma_{shp})$ – ln(γ_{rte}) trong đó Ψ là hàm digamma[2]

Hình 5 mô tả phân phối Gamma với các tham số shape, rate khác nhau.



Hình 5: Phân phối Gamma với các giá trị tham số khác nhau

2.2.4 Phân phối đa thức (Multinomial distribution)

Phân phối đa thức được định nghĩa bằng cách xác suất thu được khi tung một xúc xắc K mặt trong N lần, xét một sự kiện có K khả năng xảy ra với xác suất lần lượt $p_1, p_2, ..., p_K$ thỏa mãn $\sum_{k=1}^{K} p_K = 1$. Với N lần tung, giả sử số lần thu được mặt k là x_k sao cho $\sum_{k=1}^{K} x_k = N$. Khi đó xác suất để thu được k mặt x_k trong tổng N lần tung sẽ thỏa mãn:

$$f(x \mid p_1, p_2, ..., p_K) = \frac{n!}{x_1!...x_K!} p_1^{x_1} ... p_K^{x_K}$$

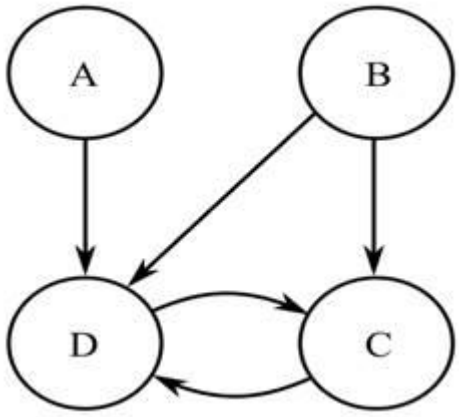
[2] https://en.wikipedia.org/wiki/Digamma_function

Tính chất của các phân phối đa thức:

- Kỳ vọng: $E(x_k) = Np_k$
- Phương sai: $Var(x_k) = Np_k(1-p_k)$

2.3 Mô hình đồ thị xác suất (probabilistic graphical model)

Xác suất là công cụ toán học được dùng để mô hình hóa những yếu tố như tính ngẫu nhiên, tính không chắc chắn, quan hệ giữa các biến số… Đây đều là những đặc điểm của dữ liệu trong thực tế. Trên cơ sở đó đã có rất nhiều các mô hình học máy được xây dựng dựa trên lý thuyết xác suất thống kê. Mô hình đồ thị xác suất [1] sẽ coi các thành phần trong bài toán là tập các biến ngẫu nhiên và minh họa mối quan hệ ràng buộc giữa chúng bằng môt mạng đồ thị. Có hai loại đồ thị là đồ thị có hướng và đồ thị vô hướng. Vì tính phù hợp của mạng đồ thị có hướng trong xây dựng mô hình nên trong phạm vi đồ án chỉ đề cập tới mô hình đồ thị mạng có hướng (mạng Bayes). Hình 6 trình bày ví dụ về một đồ thị mạng có hướng.



Hình 6: Mô hình đồ thị xác suất
(Nguồn: https://www.chegg.com/graph-of-the-probabilistic-model-31)

Một mô hình đồ thị gồm n nút tương ứng n sự kiện $X_1, X_2, ..., X_n$, giữa các nút có mỗi liên kết với nhau theo một quan hệ nào đó được biểu diễn bởi các mũi tên có hướng.

Tính chất của mạng có hướng Bayes:

- Các đường có hướng thể hiện mối quan hệ cha con của các nút trong đồ thị
- Xác suất hợp: $P[X_1,...,X_n] = \prod_{i=1}^{i=n} P[X_i \mid pa_i]$, với pa_i là tập các nút cha của X_i trong đồ thị.

2.4 Phương pháp suy diễn biến phân (variational inference)

Phương pháp suy diễn biến phân [3] là một phương pháp dùng để xấp xỉ một xác suất hậu nghiệm bằng một phân phối biến phân [4]. Bằng cách đưa ra một phân phối biến phân phù hợp sau đó biến đổi sử dụng bất đẳng thức Jensen thu được hàm biên dưới.

Việc còn lại là cực đại hóa hàm này sao cho việc xấp xỉ cho phân phối hậu nghiệm với các biến ẩn đạt giá trị tốt nhất có thể.

Ví dụ, cho mô hình với các biến ẩn Z và dữ liệu quan sát X, thay vì trực tiếp tính toán xác suất hậu nghiệm $p(Z \mid X)$, ta sẽ xấp xỉ bằng phân phối biến phân $q(Z)$:

$$q(Z) \approx p(Z \mid \mathrm{X})$$

Áp dụng bất đẳng thức Jensen để thu được:

$$\log p(X) = \log \int_Z p(X, Z) dz$$

$$= \log \int_z \frac{p(\mathrm{X}, \mathrm{Z})}{q(Z)} q(Z) dz$$

$$\geq \int_z q(Z) \log \frac{P(X, Z)}{q(Z)} \, dz = \mathrm{L}$$

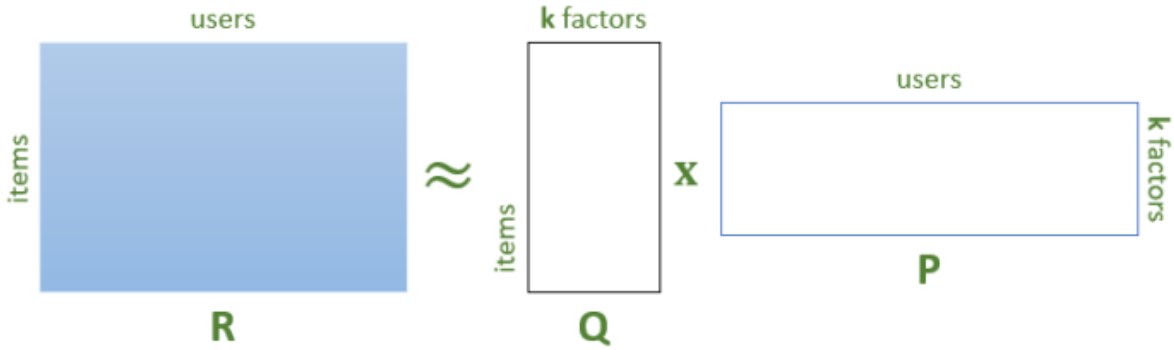
Bằng lựa chọn hàm q phù hợp ta hy vọng L sẽ được tối ưu một cách dễ dàng. Trên thực tế, biến ẩn Z có thể gồm tập hợp n biến ẩn thành phần: $Z_1, Z_2, ...Z_n$ khi đó ta có thể giả thiết ở hàm q với sự độc lập của các biến thành phần:

$$q(Z) = \prod_{i=1}^{i=n} q(Z_i \mid X) \qquad (2.1)$$

2.5 Phân rã ma trận

2.5.1 Tổng quan về phân rã ma trận

Phân rã ma trận trong bài toán gợi ý [2], một ma trận R có kích thước là U x I sẽ được phân thành tích của hai ma trận thấp chiều hơn: β có kích thước K x I và θ có kích thước là K x U sao cho thỏa mãn: $R \approx \beta^T \times \theta$



Hình 7: Minh họa phân rã ma trận
(Nguồn: https://machinelearningcoban.com/2017/05/31/matrixfactorization/)

Trong bài toán hệ gợi ý:

- Số lượng người dùng U
- Số lượng sản phẩm I
- Ma trận θ biểu diễn thuộc tính người dùng, kích thước K x U

- Ma trận β biểu diễn thuộc tính sản phẩm, kích thước K x I
- Ma trận tương tác người dùng sản phẩm R kích thước U x I

Bằng cách giả sử rằng mỗi người dùng, sản phẩm được biểu diễn bởi K các thuộc tính ẩn và cho rằng các thuộc tính ẩn đó mô tả sự liên quan giữa các tương tác người dùng và sản phẩm. Ví dụ đối với hệ thống gợi ý phim, tính chất ẩn có thể là thể loại phim như hình sự, chính trị, hài hước… hoặc cũng có thể là một sự kết hợp nào đó mà ta không thực sự cần đặt tên. Mỗi sản phẩm sẽ mang tính chất ẩn ở mức độ nào đó tương ứng với các hệ số trong vector K chiều của nó, hệ số càng cao thì mang tính chất ẩn đó càng cao. Tương tự, mỗi người dùng u cũng sẽ có xu hướng thích những tính chất ẩn nào đó và được mô tả bởi các hệ số trong vector K chiều biểu diễn cho u. Giá trị tích vô hướng của hai vector thuộc tính người dùng và sản phẩm càng cao nếu các thành phần tương ứng của 2 vector đặc trưng này đều cao. Có nghĩa, khi sản phẩm mang tính chất ẩn mà người dùng thích thì ta gợi ý sản phẩm này cho người dùng đó.

Cụ thể với β_i là vector thuộc tính biểu diễn cho sản phẩm i và θ_u là vector thuộc tính biểu diễn cho người dùng u trong không gian K chiều. Khi đó mức độ phù hợp của người dùng u đối với sản phẩm i được tính bởi:

$$r_{ui} = \beta_i^T \theta_u$$

Phân rã ma trận trong hệ gợi ý [15] dựa trên cách tiếp cận trên, bằng cách xây dựng các vector thuộc tính cho mỗi người dùng và sản phẩm, dựa vào đó đưa ra mức độ phù hợp cho mỗi cặp người dùng sản phẩm mà chưa được ghi nhận trước đó trong ma trận tương tác người dùng- sản phẩm. Thông thường ma trận tương tác người dùng- sản phẩm rất thưa vì trong thực tế không phải người dùng nào cũng có thể tương tác với toàn bộ các sản phẩm có trong hệ thống.

Mục đích là học hai ma trận biểu diễn β và θ sao cho $\beta^T \times \theta$ gần với R nhất. Xây dựng hàm lỗi Euclid, ta sẽ cần cực tiểu hóa hàm lỗi:

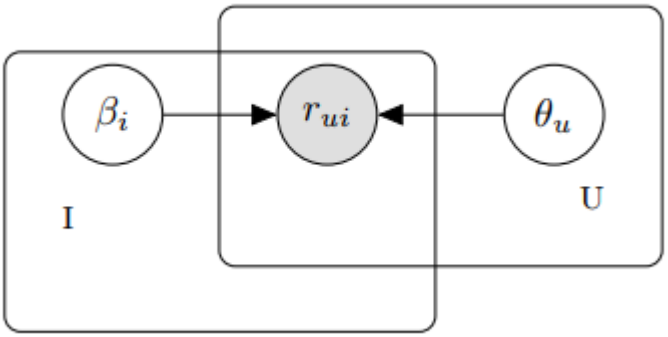
$$L(\beta,\theta) = \frac{1}{2s} \sum_{u=1}^{U} \sum_{i:r_{ui}=1} (r_{ui} - \beta_i \theta_u)^2 \qquad (2.2)$$

- r_{ui} = 1 nếu sản phẩm i đã được đánh giá bởi người dùng u, ngược lại có giá trị bằng 0.
- s là số lượng đánh giá của người dùng cho sản phẩm trong ma trận đánh giá R.

Một phương pháp tiếp cận khác theo hướng xác suất. Xét đối với một tương tác cụ thể của người dùng u đối với sản phẩm i, của giá trị tương tác thực tế r_{ui} tuân theo một hàm phân phối xác suất nào đó với kỳ vọng $\beta_i^T \theta_u$, ký hiệu:

$$p_{ui}(r_{ui} \mid \beta_i^T \theta_u)$$

Hình 8 mô tả mô hình đồ thị xác suất cho phân rã ma trận.



Hình 8: Mô hình đồ thị xác suất trong phân rã ma trận

Khi đó ta có:

$$P(\mathrm{R} \mid \beta, \theta) = \prod_{u=1,i=1}^{U,I} p_{ui}$$

Đưa về dạng log, thu được hàm mục tiêu:

$$L = \sum_{u=1,i=1}^{U,I} \log p_{ui} \tag{2.3}$$

Phần tiếp theo sẽ trình bày về các phân phối sẽ được sử dụng cho p, ở đây em sẽ đề cập đến hai phân phối Gauss và Poisson.

2.5.2 Phân rã ma trận sử dụng phương pháp Gauss

Với kỳ vọng $\beta_i^T \theta_u$ ta có hàm xác suất ở công thức (2.3) được xác định bởi phần phối Gauss, thu được:

$$\mathrm{p_{ui}}(\mathrm{r_{ui}} \mid \beta_i^T \theta_u) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{r_{ui} - \beta_i^T \theta_u}{\sigma}\right)^2} \tag{2.4}$$

Trong đó: σ là giá trị ảnh hưởng đến sự biến động của r_{ui} quanh giá trị kỳ vọng $\beta_i^T \theta_u$

Đặt $c = \frac{1}{\sigma^2}$, c là độ tin cậy của giá trị tương tác r_{ui} quanh kỳ vọng $\beta_i^T \theta_u$. Khi đó (2.4) trở thành:

$$\mathrm{p_{ui}}(\mathrm{r_{ui}} \mid \beta_i^T \theta_u) = \frac{\sqrt{c}}{\sqrt{2\pi}} e^{-\frac{c}{2}(r_{ui} - \beta_i^T \theta_u)^2}$$

Khi độ tin cậy c càng nhỏ giá trị tương tác r_{ui} có thể khác xa so với giá trị kỳ vọng và ngược lại. Tiếp tục thay vào công thức (2.3) thu được:

$$L = \sum_{u=1,i=1}^{U,I} \log p_{ui} = \sum_{u=1,i=1}^{U,I} \left[\log \sqrt{\frac{c}{2\pi}} - \frac{c}{2}\left(r_{ui} - \beta_i^T \theta_u\right)^2 \right]$$

$$= \sum_{u=1,i=1}^{U,I} -\frac{c}{2}\left(r_{ui} - \beta_i^T \theta_u\right)^2 + const \tag{2.5}$$

Công thức (2.5) thu được chính là hàm lỗi Euclid của ma trận tương tác người dùng – sản phẩm R với ma trận tích của hai ma trận thuộc tính β và θ. Việc tối ưu đồng thời là tương đối phức tạp, thay vào đó ta sử dụng phương pháp Coordinate Ascent, tối ưu một biến trong khi cố định biến còn lại cho tới khi hội tụ.

Tối ưu hàm lỗi:

Khi cố định β_i đạo hàm theo biến θ_u cho mỗi người dùng u:

$$L(\theta_u) = -\sum_{i=1}^{I} \frac{c}{2}(\mathrm{r}_{ui} - \beta_i^T \theta_u)^2 + const$$

Với r_u là một vector có kích thước I x 1 biểu diễn các tương tác của người dùng u với I item, đạo hàm theo θ_u ta được:

$$\frac{dL}{d\theta_u} = -c(\beta\beta^T\theta_u - \beta \mathrm{r}_u)$$

Cho đạo hàm này bằng 0 ta được: $\theta_u = \left(\beta\beta^T\right)^{-1}\beta r_u$

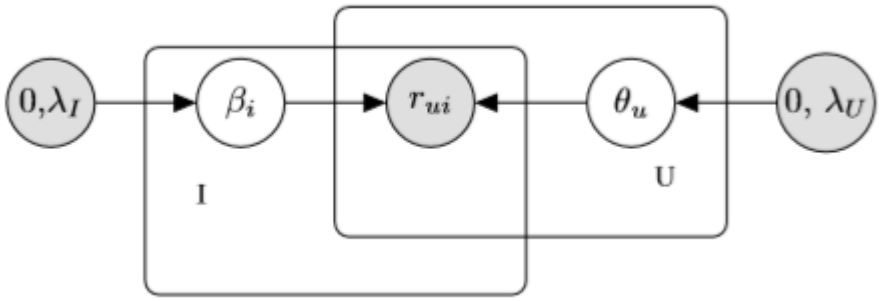
Tương tự cho mỗi vector β_i ta được: $\beta_i = (\theta\theta^T)^{-1}\theta r_i$

Phân rã ma trận Gaussian với ràng buộc biến

Giả sử rằng, các giá trị thuộc tính người dùng θ_u và thuộc tính sản phẩm β_i biến động quanh kỳ vọng 0 với độ tin cậy lần lượt là λ_u và λ_i. Hay:

$$p(\theta_u \mid \lambda_U) \sim \mathrm{N}(0, \lambda_U^{-1} I_K) \quad (*)$$
$$p(\beta_i \mid \lambda_I) \sim N(0, \lambda_I^{-1} I_K) \; (**)$$

Hình dưới mô tả phân rã ma trận Gauss với ràng buộc biến:



Hình 9: Mô hình phân rã ma trận Gauss với ràng buộc biến

Ta có : $\mathrm{p}(\mathrm{R} \mid \theta, \beta, \lambda_U, \lambda_I) = p(R \mid \theta, \beta)\, p(\theta \mid \lambda_U)\, \mathrm{p}(\beta \mid \lambda_I)$

Với (*) và (**) ta đưa vào log ta được:

$$\mathrm{L} = -\frac{c}{2}\sum_{u=1,i=1}^{U,I}\left(r_{ui} - \beta_i^T\theta_u\right)^2 - \frac{\lambda_U}{2}\sum_{u=1}^{U}\theta_u^T\theta_u - \frac{\lambda_I}{2}\sum_{i=1}^{I}\beta_i^T\beta_i + const \qquad (2.6)$$

Ta có thể nhận thấy công thức (2.6) chính là hàm lỗi Euclid của ma trận tương tác người dùng, sản phẩm và tích của hai ma trận thuộc tính người dùng, sản phẩm với các giá trị phía sau được coi là phần regularization cho người dùng, sản phẩm.
Khi cho c =1 với ý nghĩa của tất cả các quan sát được đều có vai trò như nhau. Đạo hàm của (2.6) lần lượt theo θ_u và β_i sau đó giải phương trình đạo hàm bằng 0, thu được công thức:

$$\theta_u = (\beta\beta^T + \lambda_U \mathrm{I}_K)^{-1} \beta r_u$$
$$\beta_i = \left(\theta\theta^T + \lambda I_K\right)^{-1} \theta r_i$$

2.5.3 Phân rã ma trận sử dụng phương pháp Poisson

Khác so với hướng tiếp cận theo phương pháp Gauss, hướng tiếp cận này coi mức độ tương tác giữa mỗi người dùng và sản phẩm là các giá trị rời rạc, nên coi nó là một biến ngẫu nhiên rời rạc. Khi đó độ tương tác giữa người dùng u và sản phẩm I sẽ tuân theo phân phối Poisson với kỳ vọng $\beta_i^T \theta_u$:

$$\mathrm{p_{ui}}(\mathrm{r_{ui}} \mid \beta_i^T \theta_u) = \mathrm{Poisson}(\beta_i^T \theta_u) = (\beta_i^T \theta_u)^{r_{ui}} \frac{e^{-\beta_i^T \theta_u}}{r_{ui}!}$$

Thay vào công thức (2.3) thu được:

$$L = \sum_{u=1,i=1}^{U,I} (\mathrm{r_{ui}} \log\left(\beta_i^T \theta_u\right) - \log\left(r_{ui}!\right)) - \left(\sum_{i=1}^{I} \beta_i^T\right)\left(\sum_{u=1}^{U} \theta_u\right) \quad (2.7)$$

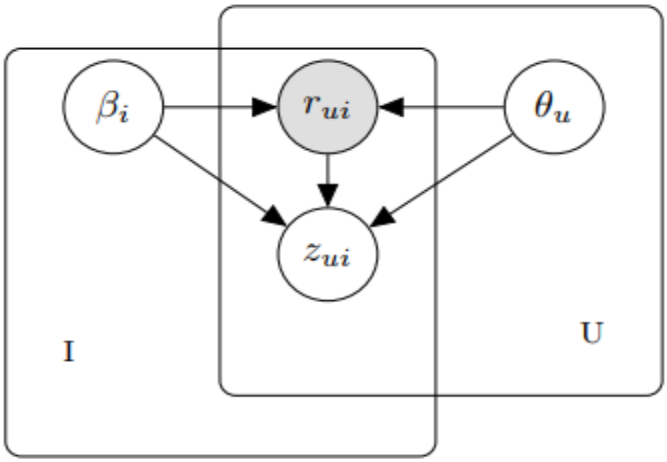
Trực tiếp tối ưu cho hàm mục tiêu (2.7) là khá khó khăn, ở đây sẽ sử dụng biến ẩn và suy diễn biến phân [11]. Với z là một biến ẩn thỏa mãn:

- $\mathrm{z_{uik}} \sim \mathrm{Poisson}(\beta_{ik}\theta_{uk})$
- $\sum_{k=1}^{K} z_{uik} = \mathrm{r_{ui}}$

Khi đó ta có mô hình phân rã ma trận Poisson được minh họa như hình 10. Biến ẩn z ở đây là một vector K chiều, và thành phần thứ k của nó thể hiện sự tác động của thuộc tính ẩn thứ k của người dùng u và thuộc tính k của sản phẩm i lên việc đánh giá cho tương tác r_{ui}. Tương tác r_{ui} thu được chính là tổng của các sự tương tác mỗi thuộc tính ẩn này giữa người dùng và sản phẩm.

Đặt phân phối biến phân cho z:

- $\mathrm{q}(\mathrm{z_{ui}} \mid \phi_{ui}) \sim Multinomial(\phi_{ui})$
- $\sum_{k=1}^{K} \phi_{uik} = 1$
- $\mathrm{E_{q(z)}}\, \mathrm{z_{uik}} = \mathrm{r_{ui}} \phi_{uik}$



Hình 10: Sử dụng biến bổ trợ z cho mô hình phân rã ma trận Poisson

Áp dụng suy diễn biến phân, ta được hàm biên dưới (lower bound):

$$
\begin{aligned}
l &= E_{q(z)} \log p(R, \beta, \theta, Z) - E_{q(z)} \log q(z) \\
&= E_{q(z)} \log p(R \mid \beta, \theta) + E_{q(z)} \log p(Z \mid \beta, \theta, R) - E_{q(z)} \log q(z) \\
&= E_{q(z)} \left(\sum_{u,i,r_{ui}>0}^{U,I} \left(r_{ui} \log\left(\beta_i^T \theta_u\right) - \log\left(r_{ui}!\right)\right)\right) - \left(\sum_{i=1}^{I} \beta_i\right)^T \left(\sum_{u=1}^{U} \theta_u\right) \\
&+ \sum_{u,i}^{U,I} E_{q(z)} \left(\log \frac{r_{ui}}{\prod_k z_{uik}} \prod_k \left(\frac{\beta_{ik}\theta_{uk}}{\beta_i^T \theta_u} \right)^{z_{uik}} \right) - \sum_{u,i}^{U,I} E_{q(z)} \log \frac{1}{\prod_k z_{uik}} \prod_k (\phi_{uik})^{z_{uik}} \\
&= -\left(\sum_{i=1}^{I} \beta_i\right)^T \left(\sum_{u=1}^{U} \theta_u\right) + \sum_{u,i} \sum_k E_{q(z)} z_{uik} \log \beta_{ik}\theta_{uk} - \sum_{u,i} \sum_k E_{q(z)} z_{uik} \log \phi_{uik} + const
\end{aligned}
\qquad (2.8)
$$

Đạo hàm của l trong công thức (2.8) theo ϕ, β, θ và cho đạo hàm chúng bằng 0, ta được:

$$
\begin{aligned}
\phi_{uik} &\propto \beta_{ik}\theta_{uk} \\
\beta_{ik} &= \frac{\sum_u r_{ui}\phi_{uik}}{\sum_u \theta_u} \\
\theta_{uk} &= \frac{\sum_i r_{ui}\phi_{uik}}{\sum_i \beta_i}
\end{aligned}
$$

Nhận xét: Từ công thức (2.7) ta thấy hàm mục tiêu của phương pháp này chỉ phụ thuộc vào những sản phẩm đã được rate bởi người dùng, tức $r_{ui} > 0$. Như vậy sẽ hoạt động tốt trong cả trường hợp dữ liệu đánh giá thưa.

2.6 Dropout và sử dụng tri thức tiên nghiệm

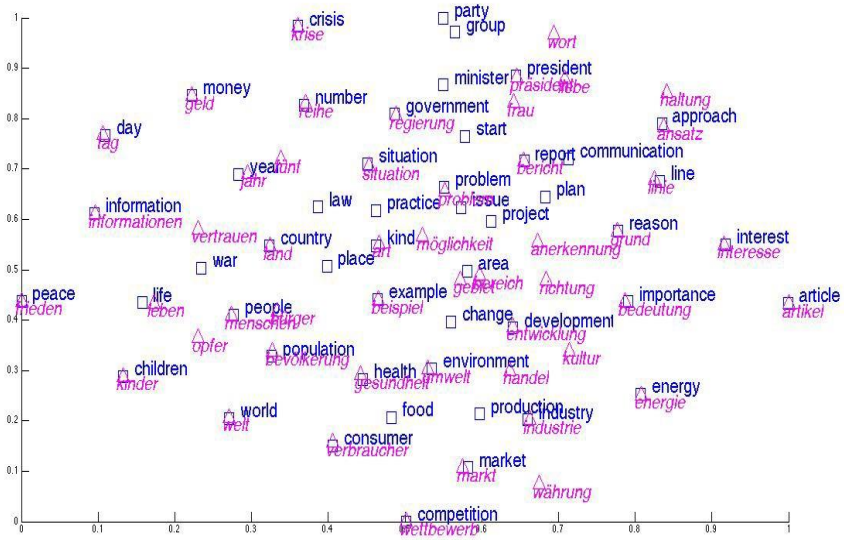
Học loại bỏ là một kỹ thuật ban đầu được đưa vào học trong các mạng neural [19] với ý tưởng là ngẫu nhiên loại bỏ đi các liên kết trong một mạng đầy đủ ban đầu. Tương tự như vậy, dropout được áp dụng trong trường hợp này bằng cách loại bỏ ngẫu nhiên một số đầu vào với một xác suất cố định và sau đó học trên các dữ liệu còn lại.

Kỹ thuật này tuy đơn giản nhưng thực sự đạt hiệu quả tốt trên thử nghiệm, điều đó chứng tỏ rằng việc loại bỏ một cách ngẫu nhiên dữ liệu đầu vào giúp làm giảm nhiễu cho mô hình. Ta có thể giải thích như sau, do số lượng các sản phẩm trong thực tế đều vô cùng lớn nên mỗi người dùng thường chỉ tương tác tới một số nhỏ các sản phẩm thường xuyên sử dụng hoặc do vì sở thích cá nhân. Dẫn đến ma trận tương tác người dùng sản phẩm thường rất thưa. Do vậy, trong ma trận tương tác có rất nhiều sản phẩm có giá trị tương tác là 0, mặt khác khi một tương tác có giá trị bằng 0 thì có thể có hai khả năng: một là người dùng không thích đến sản phẩm đó, hai là người dùng chưa biết đến sản phẩm đó. Việc học trên tất cả các tương tác này tức là chúng ta đang ép mô hình mô hình hóa cho cả những tương tác không thích và những tương tác chưa biết của người dùng.

Do số lượng các tương tác bằng 0 chiếm một tỷ lệ rất lớn trên tổng các tương tác của người dùng, khi đó việc loại bỏ với một tỷ ngẫu nhiên thì xu hướng các đánh giá bằng 0 này có khả năng được loại bỏ cao hơn nhiều so với các đánh giá khác 0. Dẫn đến mô hình giảm bớt các trường hợp không mang ý nghĩa cần phải mô hình hóa. Một lợi ích khác dễ thấy nhất khi học loại bỏ là giúp giảm lượng dữ liệu training, tiết kiệm tài nguyên tính toán và tăng tốc độ học cho mô hình.

Học sử dụng tri thức tiên nghiệm Tri thức tiên nghiệm là tri thức đã được biết trước về một sự kiện hay một đối tượng nào đó. Đối với trong bài toán này, với một số lượng từ ngữ và ngữ cảnh hạn chế nếu chỉ dựa vào đó để xây dựng ma trận biểu diễn của từ thì khó có thể đảm bảo được tính ngữ nghĩa và chính xác. Do vậy sử dụng tri thức tiên nghiệm là một giải pháp thay thế trong bài toán này. Cụ thể, trong đồ án sử dụng bộ tri thức tiên nghiệm từ nhúng trong dự án Glove[3] của đại học Stanford, trên bộ từ vựng tiếng anh học từ Wikipedia. Hình 11 mô tả biểu diễn nhúng của từ.

[3] https://nlp.stanford.edu/projects/glove/



Hình 11: Minh họa biểu diễn nhúng của từ
(Nguồn: https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding)

Chương 3: Một số mô hình sử dụng phân rã ma trận

Ở chương 2 em đã đề cập đến các phương pháp phân rã ma trận sử dụng với các phân phối Gauss, Poisson. Ở chương này trình bày một số nghiên cứu sử dụng các phương pháp phân rã ma trận cho hệ gợi ý đã được công bố, bao gồm:

- WMF – Weighted Matrix Factorization.
- HPF – Hierarchical Poisson Factorization.
- CTMP – Collaborative Topic Model for Poisson distributed ratings.

3.1 WMF

WMF [10] là phương pháp phân rã ma trận theo hướng tiếp cận Gauss. Trong WMF các tương tác giữa người dùng đều được nhị phân hóa về hai giá trị 0 và 1 theo quy tắc sau:

$$p_{ui} = \begin{cases} 1, r_{ui} > 0, \\ 0, r_{ui} = 0 \end{cases}$$

Do với tương tác tiềm ẩn, giá trị có thể là số lần người dùng đó xem một đoạn phim, số lần đọc một quyển sách, thời gian xem một bộ phim hoặc chính là mức độ yêu thích của người dùng đối với sản phẩm, do đó ta sử dụng thêm một độ tin cậy cho mỗi tương tác r_{ui}:

$$c_{ui} = 1 + \rho . r_{ui}$$

Trong đó: ρ là giá trị hằng số cho trước. Khi đó ta đi tìm θ, β để tối ưu hàm mục tiêu:

$$L = \sum_{u=1,i=1}^{U,I} c_{ui} \left(p_{ui} - \theta_u{}^T \beta_i \right)^2 + \frac{\lambda_U}{2} \sum_{u=1}^{U} \theta_u^T \theta_u + \frac{\lambda_I}{2} \sum_{i=1}^{I} \beta_i^T \beta_i$$

với λ_u, λ_i là các trọng số regularization cho người dùng và sản phẩm.

Đối với mỗi sản phẩm i ta đặt C^i là ma trận đường chéo kích thước U x U sao cho $C^i_{uu} = c_{ui}$ và đối với mỗi người dùng u ta đặt C^u là ma trận đường chéo kích thước I x I sao cho $C^u_{ii} = c_{ui}$, p_u là các vector thể hiện tương tác của người dùng u với I sản phẩm và p_i thể hiện tương tác của U người dùng với sản phẩm i, thay giá trị của c vào hàm mục tiêu trên và tính đạo hàm ta được:

$$\theta_u = (\beta C^u \beta^T + \lambda_U \mathrm{I}_K)^{-1} \beta r_u \quad (3.1)$$

$$\beta_i = \left(\theta C^i \theta^T + \lambda_I I_K \right)^{-1} \theta r_i \quad (3.2)$$

Thuật toán học cho mô hình WMF:

Input: Dữ liệu quan sát R, các siêu tham số ρ, λ_U, λ_I

Output: Ước lượng β_i, θ_u

Khởi tạo β, θ

Repeat

For u=1: U **do**

Cập nhật θ_u theo công thức (3.1)

Endfor

For i=1: I **do**

Cập nhật β_i theo công thức (3.2)

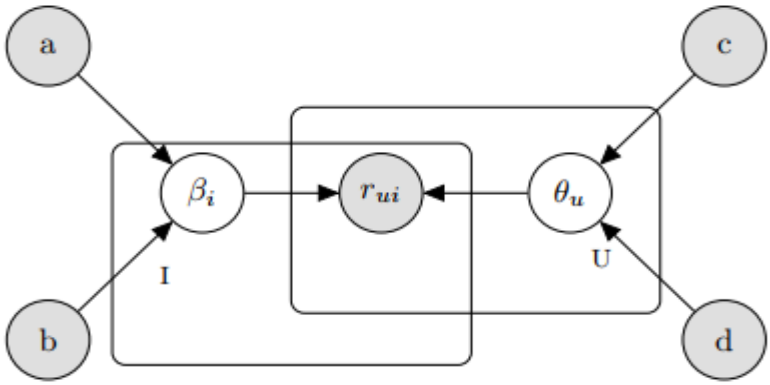
Endfor

Until hội tụ

3.2 HPF

Mô hình phân rã ma trận Poisson phân cấp – HPF [9]. Khác với phân rã sử dụng Poisson được trình bày ở phần trước, HPF sử dụng các mức phân cấp cho thuộc tính người dùng, sản phẩm θ_u, β_i. Việc sử dụng phân cấp để biểu diễn cho các thuộc tính người dùng, sản phẩm cho phép thể hiện được sự đa dạng của hệ thống trong thực tế, ví dụ như có nhóm các người dùng có xu hướng mua sắm nhiều hơn so với số còn lại. Hoặc tương tự đối với các nhóm sản phẩm có xu hướng phổ biến hơn so với các sản phẩm khác. Việc lợi ích thứ hai của phương pháp này là có thể mô tả được vector thưa trong biểu diễn thuộc tính ẩn của người dùng, sản phẩm. Cụ thể, như em đã đề cập ở phần 2.5.1 mỗi thuộc tính ẩn biểu diễn một tính chất nào đó của đối tượng, ví dụ đối với một bộ phim thì thường chỉ thuộc một hoặc một vài thể loại nhất định cho nên khi biểu diễn chúng thì các thành phần tương ứng với thuộc tính đó sẽ có giá trị trội hơn so với các thuộc tính khác, do vậy khi sử dụng HPF có khả năng sẽ mô tả được những trường hợp như vậy vì HPF sử dụng phân phối tiên nghiệm Gamma để biểu diễn cho người dùng, sản phẩm bằng cách chọn tham số shape nhỏ thì phần lớn các trọng số sẽ tiến gần về 0 và một số lượng nhỏ còn lại sẽ có giá trị trội hơn. Dưới đây em có trình bày tóm tắt về các phân cấp của mô hình phân rã ma trận Poisson phân cấp.

Ở mức phân cấp thứ nhất



Hình 12: Mô hình HPF ở mức phân cấp thứ nhất

Mô hình sinh cho phân rã ma trận Poisson phân cấp 1

- Cho mỗi sản phẩm i:
  - Cho mỗi thành phần k, lấy mẫu $\beta_{ik} \sim Gamma(a,b)$
- Cho với mỗi người dùng u:
  - Cho mỗi thành phần k, lấy mẫu $\theta_{uk} \sim Gamma(c,d)$
- Cho mỗi tương tác người dùng u, sản phẩm i: $r_{ui} \sim Poisson(\beta_i^T \theta_u)$

Trong mô hình HPF với mức phân cấp thứ nhất, ta có các giả thiết vector biểu diễn thuộc tính người dùng, sản phẩm tuân theo phân phối Gamma với các tham số shape, rate như sau:

$$p(\beta_{ik} \mid a,b) \sim Gamma(a,b)$$
$$p(\theta_{uk} \mid c,d) \sim Gamma(c,d)$$

Tương tự với phân rã ma trận Poisson ở 2.5.3 tiếp tục sử dụng biến phụ z, ta có phân phối biến phân có dạng:

$$q(\beta,\theta,z) = \prod_{i,k} q(\beta_{ik} \mid \zeta_{ik}) \prod_{u,k} q(\theta_{uk} \mid \gamma_{uk}) \prod_{u,i} q(z_{ui} \mid \phi_{ui})$$

Với các ràng buộc sau:

$$q(\beta_{ik} \mid \zeta_{ik}) \sim \text{Gamma}(\zeta_{ik}^{shp}, \zeta_{ik}^{rte})$$
$$q(\theta_{uk} \mid \gamma_{uk}) \sim \text{Gamma}(\gamma_{uk}^{shp}, \gamma_{uk}^{rte})$$
$$q(z_{ui} \mid \phi_{ui}) \sim \text{Multinomial}(\phi_{ui})$$

Ta có công thức cập nhật của các biến:

$$\phi_{ui} \propto \exp\{\Psi(\gamma_{uk}^{shp}) - \log \gamma_{uk}^{rte} + \Psi(\zeta_{ik}^{shp}) - \log \zeta_{ik}^{rte}\}$$
$$\gamma_{uk}^{shp} = c + \sum_{i=1}^{I} r_{ui} \phi_{uik}$$

$$\gamma_{uk}^{rte} = d + \sum_{i=1}^{I} \frac{\zeta_{ik}^{shp}}{\zeta_{ik}^{rte}}$$

$$\zeta_{ik}^{shp} = a + \sum_{u=1}^{U} r_{ui}\phi_{uik}$$

$$\zeta_{ik}^{rte} = b + \sum_{u=1}^{U} \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$$

Cuối cùng, ta có công thức tính các giá trị thuộc tính:

$$\theta_{uk} = \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}}$$

$$\beta_{ik} = \frac{\zeta_{ik}^{shp}}{\zeta_{ik}^{rte}}$$

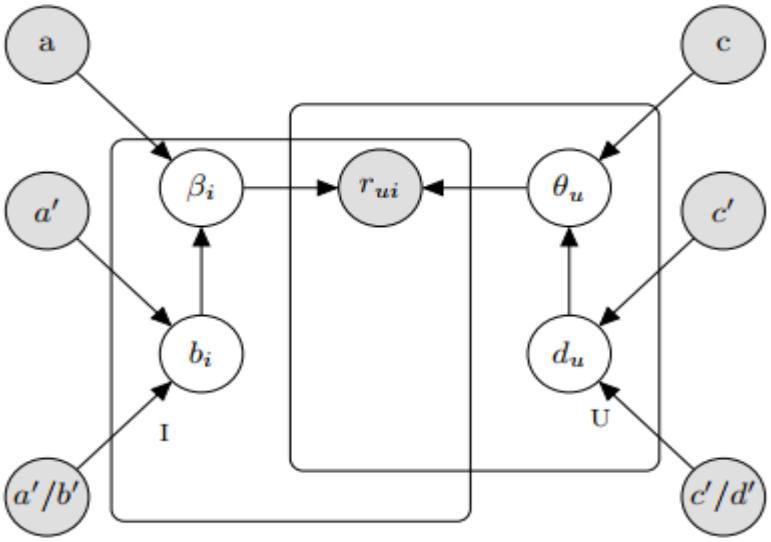
Nhận xét: Trong việc biểu diễn các đối tượng người dùng, sản phẩm thành một vector thuộc tính ẩn K chiều đại diện cho đối tượng đó. Việc sử dụng mô hình phân cấp Poisson phù hợp hơn là hướng tiếp cận Gauss. Do khi biểu diễn vector thuộc tính cho mỗi đối tượng khởi tạo trọng số sử dụng phân phối Gamma thì các thành phần trọng số sẽ là một số dương, còn khi sử dụng phân phối Gauss sẽ có khả năng các thuộc tính của đối tượng sau khi biểu diễn sẽ có giá trị âm.

Ở mức phân cấp thứ hai

Khác với phân rã ma trận Poisson một cấp, ở phân rã ma trận Poisson cấp 2 ta tiếp tục giả sử các tham số b, d tuân theo một phân phối xác suất Gamma như sau:

$b_i \sim \text{Gamma}(a', a' / b')$

$d_u \sim \text{Gamma}(c', c' / d')$



Hình 13: Mô tả mô hình gợi ý Poisson phân cấp mức 2

Mô hình sinh cho phân rã ma trận Poisson phân cấp 2

- Cho mỗi sản phẩm i:
  - Lấy mẫu $b_i \sim \text{Gamma}(a', a' / b')$
  - Cho mỗi thành phần k, lấy mẫu $\beta_{ik} \sim Gamma(a, b_i)$
- Cho với mỗi người dùng u:
  - Lấy mẫu $d_u \sim \text{Gamma}(c', c' / d')$
  - Cho mỗi thành phần k, lấy mẫu $\theta_{uk} \sim Gamma(c, d_u)$
- Cho mỗi tương tác người dùng u, sản phẩm i: $r_{ui} \sim Poisson(\beta_i^T \theta_u)$

Lúc này phân phối biến phân có dạng:

$$q(\beta, \theta, b, d, z) = \prod_{i,k} q(\beta_{ik} | \zeta_{ik}) \prod_{u,k} q(\theta_{uk} | \gamma_{uk}) \prod_i q(b_i | \kappa_i) \prod_u q(d_u | \tau_u) \prod_{u,i} q(z_{ui} | \phi_{ui})$$

Bổ xung thêm hai thành phần mới so với phân cấp bậc 1:

$$q(b_i | \kappa_i) \sim Gamma(\kappa_i^{shp}, \kappa_i^{rte})$$
$$q(d_u | \tau_u) \sim Gamma(\tau_u^{shp}, \tau_u^{rte})$$

Tiếp tục sử dụng suy diễn biến phân với các giá trị tham số:

$$\kappa_i^{shp} = a' + Ka$$
$$\tau_u^{shp} = c' + Kc$$

Kết quả cuối cùng ta được công thức cập nhật:

$$\phi_{ui} \propto \exp\{\Psi(\gamma_{uk}^{shp}) - \log \gamma_{uk}^{rte} + \Psi(\zeta_{ik}^{shp}) - \log \zeta_{ik}^{rte}\} \quad (3.3)$$

$$\gamma_{uk}^{shp} = c + \sum_{i=1}^{I} r_{ui} \phi_{uik} \quad (3.4)$$

$$\gamma_{uk}^{rte} = \frac{\tau_u^{shp}}{\tau_u^{rte}} + \sum_{i=1}^{I} \frac{\zeta_{ik}^{shp}}{\zeta_{ik}^{rte}} \quad (3.5)$$

$$\tau_u^{rte} = \frac{c'}{d'} + \sum_k \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}} \quad (3.6)$$

$$\zeta_{ik}^{shp} = a + \sum_{u=1}^{U} r_{ui} \phi_{uik} \quad (3.7)$$

$$\zeta_{ik}^{rte} = \frac{\kappa_i^{shp}}{\kappa_i^{rte}} + \sum_{u=1}^{U} \frac{\gamma_{uk}^{shp}}{\gamma_{uk}^{rte}} \quad (3.8)$$

$$\kappa_i^{rte} = \frac{a'}{b'} + \sum_k \frac{\zeta_{ik}^{shp}}{\zeta_{ik}^{rte}} \quad (3.9)$$

Thuật toán học mô hình phân cấp Poisson mức 2

Input: Dữ liệu quan sát R, siêu tham số a, c, a', b', c', d'
Output: Ước lượng $\gamma^{shp}, \gamma^{rte}, \zeta^{shp}, \zeta^{rte}$
Repeat

For u,i, $r_{ui} > 0$ **do**

Cập nhật ϕ_{ui} theo công thức (3.3)

End for

For u = 1: U, k =1: K **do**

Cập nhật γ_{uk}^{shp} theo công thức (3.4)
Cập nhật γ_{uk}^{rte} bằng công thức (3.5)
Cập nhật τ_{u}^{rte} bằng công thức (3.6)

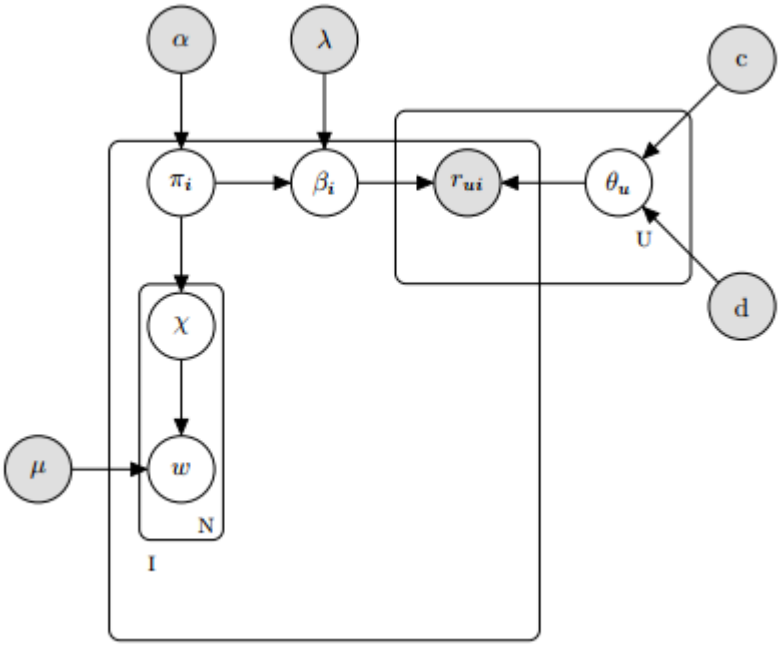
End for
For i = 1: I, k =1: K **do**

Cập nhật ζ_{ik}^{shp} theo công thức (3.7)
Cập nhật ζ_{ik}^{rte} theo công thức (3.8)
Cập nhật κ_{i}^{rte} theo công thức (3.9)

End for
Until hội tụ

3.3 CTMP

CTMP [20] viết tắt của Collaborative Topic Model for Poisson distributed ratings là một mô hình gợi ý xây dựng dựa trên ý tưởng kết hợp sử dụng nội dung mô tả sản phẩm trước khi cho vào phân rã Poisson. Việc thông tin mô tả sản phẩm của mô hình CTMP được khai thác thông qua mô hình LDA (Lantent Dirichlet Allocation) [6]. Phần còn lại tận dụng những ưu điểm của phương pháp phân rã Poisson phân cấp, cụ thể thuộc tính người dùng vẫn lấy dựa trên phân phối Gamma với 2 tham số đầu vào shape, rate. Hình dưới mô tả mô hình đồ thị xác suất của CTMP. Dưới đây em sẽ giới thiệu qua về ý tưởng chính của mô hình, mô hình sinh, các công thức cập nhật cho các biến. Một cách cụ thể và chi tiết được nhóm tác giả đề cập ở [20].



Hình 14: Mô hình đồ thị xác suất biểu diễn cho CTMP

Trong đó các thành phần của mô hình:

- Tập biểu diễn cho các từ mô tả của mỗi sản phẩm thứ i, ký hiệu: w_i
- Tập dữ liệu từ hệ thống bao gồm: giá trị tương tác giữa người dùng – sản phẩm r_{ui} và mô tả của sản phẩm đó, ký hiệu: $D = \{r_{ui}, a_i\}_{u=1,i=1}^{U,I}$
- $\mu = \{\mu_{kv}\}_{K \times V}$: là một biến toàn cục trong mô hình, biểu diễn phân phối của K chủ đề trên tập từ vựng V từ.
- vector K chiều biểu diễn nội dung của sản phẩm theo chủ đề, ký hiệu: $\pi_{1:I}$
- χ : chủ đề tương ứng của từng từ.

Mô hình sinh của CTMP:

- Mỗi người dùng u, lấy mẫu $\theta_u \sim Gamma(c, d)$
- Mỗi sản phẩm i:
  - Lấy phân phối chủ đề $\pi_i \sim Dirichlet(\alpha)$
  - Đối với từ thứ n trong mô tả của sản phẩm i:
    - Lấy chủ đề $\chi_{in} \sim Categorical(\pi_i)$
    - Lấy từ $w_{in} \sim Categorical(\mu_{\chi_{in}})$
  - Lấy vector thuộc tính sản phẩm: $\beta_i \sim N(\pi_i, \lambda^{-1} I_K)$
- Với mỗi cặp người dùng u – sản phẩm i, lấy giá trị tương tác: $\mathrm{r}_{ui} \sim Poisson(\beta_i^T \theta_u)$

Hàm mục tiêu:

Sử dụng biến phụ z như HPF, ta có hàm mục tiêu xác suất hợp như sau:

$$L = \log P(\beta, \pi, D \mid \alpha, \mu, c, d)$$

$$= \sum_{i=1}^{I} \log P(\pi_i, \mathrm{w}_i \mid \alpha, \mu) + \sum_{i=1}^{I} \log P(\beta_i \mid \pi_i, \lambda) + \sum_u \sum_i \log \int \sum_{r_{ui}} P(\mathrm{r}_{ui}, \mathrm{z}_{ui}, \theta_u \mid \beta_i, \mathrm{c}, \mathrm{d})\, \mathrm{d}\, \theta_u$$

Sử dụng phân phối biến phân:

$$q(\theta_u, z_{ui}) = q(z_{ui} \mid \mathrm{r}_{ui}, \phi_{ui}) \prod_k q(\theta_u \mid shp_{uk}, rte_{uk})$$

Với:

$$\mathrm{q}(\mathrm{z}_{ui} \mid \mathrm{r}, \phi_{ui}) \sim \mathrm{Multinomial}(\mathrm{z}_{ui} \mid \mathrm{r}, \phi_{ui})$$

$$q(\theta_{uk} \mid \mathrm{shp}_{uk}, \mathrm{rte}_{uk}) \sim \mathrm{Gamma}(\mathrm{shp}_{uk}, rte_{uk})$$

- Sử dụng biến phân, ta được công thức cập nhật:

$$\phi_{uik} \propto \exp\{\log \beta_{ik} + \psi(shp_{uk} - \log rte_{uk})\} \quad (3.10)$$

$$\mathrm{shp}_{uk} = c + \sum_i r_{ui} \phi_{uik} \quad (3.11)$$

$$rte_{uk} = d + \sum_i \beta_{ik} \quad (3.12)$$

- Hàm mục tiêu cho π:

$$-\frac{\lambda}{2} \| \beta_i - \pi_i \|^2 + (\alpha - 1) \sum_k \log \pi_{ik} + \sum_v a_j^v \log(\sum_k \pi_{ik} \mu_{kv}) \quad (3.13)$$

Nhóm tác giả đã đề xuất sử dụng phương pháp OPE trong [20] để tìm giá trị phù hợp cho π.

- Hàm mục tiêu cho β:

$$f(\beta_i) = -\frac{\lambda}{2} \| \beta_i - \pi_i \|^2 + \sum_{uk} r_{ui} \phi_{uik} \log \beta_{ik} - \sum_k \beta_{ik} \sum_u \frac{shp_{uk}}{rte_{uk}}$$

Công thức cập nhật:

$$\beta_{ik} = \frac{-\sum_u \frac{shp_{uk}}{rte_{uk}} + \lambda \pi_{ik} + \sqrt{\Delta}}{2\lambda} \quad (3.14)$$

Với $\Delta = \left(-\sum_u \frac{shp_{uk}}{rte_{uk}} + \lambda \pi_{ik} \right)^2 + 4\lambda \sum_k r_{ui} \phi_{uik}$

Thuật toán học cho mô hình CTMP

Input: Dữ liệu quan sát $w_i = \{a_i\}_{i=1}^I$,R và các siêu tham số α, λ, c, d .

Output: Ước lượng các giá trị $\pi, \mu, \beta, \phi, shp, rte, \theta$.

Khởi tạo bằng cách học LDA trên tập mô tả sản phẩm

Repeat

For i=1: I **do**

Cập nhật π_i bằng cách tối ưu hàm (3.13) theo phương pháp OPE

Cập nhật β theo công thức (3.15)

End for

For u=1: U, k=1: K **do**

Cập nhật ϕ_{uik} bằng công thức (3.10)

Cập nhật shp_{uk} bằng công thức (3.11)

Cập nhật rte_{uk} bằng công thức (3.12)

End for

$\mu_{kv} \propto \sum_i a_i^v \pi_{ik}$

Until hội tụ

Chương 4: Mô hình phân rã ma trận Poisson kết hợp bộ tri thức tiên nghiệm nhúng của từ

Ở chương này em sẽ đề cập đến mô hình phân rã ma trận Poisson kết hợp sử dụng bộ tri thức tiên nghiệm nhúng của từ - Poisson Matrix Factorization using Word Embedding Prior (PFEP). Tương tự với mô hình CTMP, ý tưởng xây dựng mô hình này trên cơ sở sử dụng phân rã ma trận Poisson và khai thác thông tin sản phẩm thông qua một bộ tri thức từ nhúng tiên nghiệm kết hợp với một mạng neural. Việc khai thác thông tin sản phẩm như vậy có thể phần nào khắc phục được một số nhược điểm của phương pháp Latent Dirichlet Allocation – LDA khi làm việc trên bộ dữ liệu ngắn. Ở chương này nội dung trình bày bao gồm:

- Mô hình sinh.
- Cập nhật tham số.

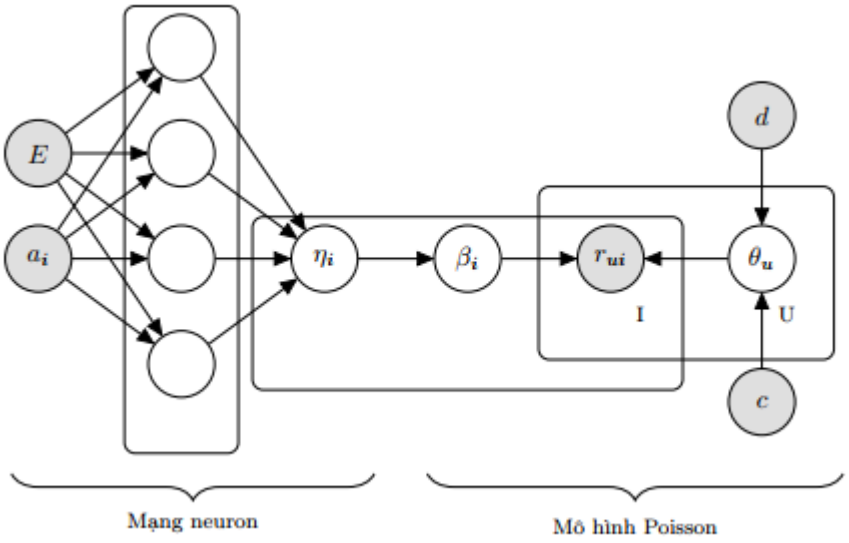
4.1 Mô hình sinh

Ký hiệu các thành phần tham gia vào mô hình:

- Ma trận biểu diễn nhúng của từ là nguồn tri thức có trước được học từ nguồn bên ngoài, trong đó có V chiều tương ứng với V từ trong từ điển và mỗi từ được biểu diễn dưới một vector K chiều, ký hiệu: $E = \{e_{lv}\}_{L \times V}$.
- Ma trận trọng số cho mạng neural dùng để khai thác bộ tri thức tiên nghiệm từ nhúng, ký hiệu: W.
- Một vector K chiều biểu diễn đầu ra của mỗi sản phẩm sau khi đi qua mạng neural, ký hiệu: η_i.
- Một vector thuộc tính K chiều biểu diễn cho mỗi sản phẩm, ký hiệu: β_i, được lấy dựa trên phân phối Gauss với kỳ vọng η_i.
- Một vector thuộc tính K chiều biểu diễn cho mỗi người dùng, ký hiệu: θ_u.

Như đã trình bày ở trên, mô hình phân rã ma trận Poisson sử dụng tri thức tiên nghiệm từ nhúng sẽ biểu diễn vector thuộc tính sản phẩm thông qua một mạng neural kết hợp với bộ tri thức từ nhúng đã huấn luyện từ trước. Các sản phẩm có mô tả gần giống nhau sẽ được biểu diễn tương tự nhau trong không gian biểu diễn. Mặt khác, việc khai thác như vậy có thể giúp khắc phục được việc không hiệu quả của phương pháp LDA khi sử dụng trên mô tả ngắn. Quá trình học bao gồm hai giai đoạn: giai đoạn thứ nhất là học để biểu diễn vector sản phẩm thông qua mạng neural, giai đoạn thứ hai là quá trình phân rã ma trận Poisson. Đầu ra của biểu diễn sản phẩm thông qua mạng neural của sản phẩm i được xác định bởi công thức $\eta_i = f(E, a_i; \mathrm{W})$, trong đó: mạng neural với ma trận trọng số W kích thước K x L và một hàm kích hoạt σ là hàm sigmoid. Một cách chi tiết, mô hình sinh của phương pháp này sẽ được em trình bày ở bên dưới.

Hình 15 biểu diễn mô hình đồ thị xác suất cho PFEP.



Hình 15: Mô hình đồ thị xác suất cho PFEP

Mô hình sinh PFEP như sau:

- Với mỗi sản phẩm i:
  - Vector biểu diễn mô tả thông tin sản phẩm: $\eta_i = f(E, a_i; \mathrm{W})$
  - Vector biểu diễn mô tả thuộc tính sản phẩm: $\beta_i = N(\eta_i, \lambda^{-1} I_K)$
- Với mỗi người dùng u:
  - Vector thuộc tính người dùng: $\theta_{uk} = Gamma(c, d)$
- Cho mỗi người dùng u và sản phẩm i, tương tác: $r_{ui} = Poisson(\beta_i^T \theta_u)$

4.2 Cập nhật tham số

Việc học mô hình sử dụng thuật toán coordinate ascent với sự kết hợp của thuật toán stochastic gradient ascent và suy diễn biến phân trong mô hình đồ thị xác suất. Sau cùng kết hợp thêm phương pháp học loại bỏ nhằm tăng chất lượng và cải thiện tốc độ học của mô hình.

4.2.1 Sử dụng gradient ascent cho PFEP

Xem xét hàm mục tiêu sau:

$$
\begin{aligned}
L &= \log P(\beta, R \mid E, a, c, d) \\
&= \log P(\beta \mid \mathrm{E}, \mathrm{a}) + \log P(\mathrm{R} \mid \beta, \mathrm{c}, \mathrm{d}) \\
&= \log P(\beta \mid \mathrm{f}(\mathrm{E}, \mathrm{a}; \mathrm{W})) + \log P(\mathrm{R} \mid \beta, \mathrm{c}, \mathrm{d}) \\
&= -\sum_{i=1}^{I} \frac{\lambda}{2} \| \beta_i - f(E, a_i; \mathrm{W}) \|^2 + \log P(R \mid \beta, c, d)
\end{aligned} \tag{4.1}
$$

Trên công thức (4.1) bao gồm hai thành phần: thành phần thứ nhất tương ứng với việc học biểu diễn nội dung sản phẩm và thành phần thứ hai tương ứng với việc học biểu diễn thuộc tính của người dùng, sản phẩm dựa trên ma trận tương tác. Để tối ưu thành phần thứ hai ta tiếp tục sử dụng biến phụ z như đã trình bày ở phần 2.5.3 với z thỏa mãn các điều kiện:

- $r_{ui} = \sum_{k=1}^{K} z_{uik}$
- $z_{uik} \sim Poisson(\beta_{ik}\theta_{uk})$

Biến đổi (4.1) sử dụng suy diễn biến phân kết hợp với bất đẳng thức Jensen, khi đó ta thu được hàm biên dưới l:

$$L = -\sum_{i=1}^{I}\frac{\lambda}{2}||\beta_i - f(E, a_i; \mathrm{W})||^2 + \log\int\sum_{z} P(R, z, \theta \mid \beta, c, d)d\theta$$
$$= -\sum_{i=1}^{I}\frac{\lambda}{2}||\beta_i - f(E, a_i; \mathrm{W})||^2 + \log\int\sum_{z} P(R, z, \theta \mid \beta, c, d)\frac{q(\theta, z)}{q(\theta, z)}d\theta$$
$$\geq -\sum_{i=1}^{I}\frac{\lambda}{2}||\beta_i - f(E, a_i; \mathrm{W})||^2 + E_{q(\theta,z)}\log P(\mathrm{R}, \mathrm{z}, \theta \mid \beta, \mathrm{c}, \mathrm{d}) - \mathrm{E}_{q(\theta,z)}\log q(\theta, \mathrm{z}) = l$$

Với phân phối biến phân:

$$Q(\theta, z) = \prod_{u=1,i=1}^{U,I} q(z_{ui} \mid r_{ui}, \phi_{ui})\prod_{u=1}^{U}\prod_{k=1}^{K} q(\theta_{uk} \mid shp_{uk}, rte_{uk})$$
$$q(z_{ui} \mid r_{ui}, \phi_{ui}) = Mult(z_{ui} \mid r_{ui}, \phi_{ui})$$
$$\mathrm{q}(\theta_{uk} \mid shp_{uk}, rte_{uk}) = Gamma(\theta_{uk} \mid shp_{uk}, rte_{uk})$$

Trong đó (shp_{uk}, rte_{uk}) là các tham số biến phân và $\phi_{ui} = \{\phi_{uik}\}_{K\times 1}$, $\sum_{k=1}^{K}\phi_{uik} = 1$

Xét hàm: $f = \log P(R, z, \theta \mid \beta, c, d) - \log Q(\theta, z)$

$$= \log P(R \mid \theta, \beta) + \log P(z \mid R, \beta, \theta) + \log P(\theta \mid c, d) - \log Q(\theta, z)$$
$$= \sum_{u=1,i=1}^{U,I}\log p(r_{ui} \mid \theta_u, \beta_i) + \sum_{u=1,i=1}^{U,I}\log p(\mathrm{z}_{ui} \mid \mathrm{r}_{ui}, \theta_u, \beta_i) + \sum_{u=1}^{U}\log p(\theta_u \mid \mathrm{c}, \mathrm{d})$$
$$- \sum_{u=1,i=1}^{U,I}\log q(z_{ui} \mid r_{ui}, \phi_{ui}) - \sum_{u=1}^{U}\log\prod_{k=1}^{K} q(\theta_{uk} \mid shp_{uk}, rte_{uk})$$

- $\sum_{u=1,i=1}^{U,I}\log p(r_{ui} \mid \theta_u, \beta_i) = \sum_{u=1,i=1}^{U,I}\left(r_{ui}\log(\beta_i^T\theta_u) - \log(r_{ui}!) - \beta_i^T\theta_u\right)$
- $\sum_{u=1,i=1}^{U,I}\log p(z_{ui} \mid r_{ui}, \theta_u, \beta_i) = \sum_{u=1,i=1}^{U,I}\log\frac{r_{ui}}{\prod_k z_{uik}}\prod_k\left(\frac{\beta_{ik}\theta_{uk}}{\beta_i^T\theta_u}\right)^{z_{uik}}$
- $\sum_{u=1}^{U}\log p(\theta_u \mid c, d) = \sum_{u=1}^{U}\log\prod_k\frac{d^c.\theta_{uk}^{c-1}.e^{-\theta_{uk}.d}}{\Gamma(c)}$

- $\sum_{u=1,i=1}^{U,I} \log q(z_{ui} \mid r_{ui}, \theta_{ui}) = \sum_{u=1,i=1}^{U,I} \log \frac{1}{\prod_k z_{uik}} \prod_k (\phi_{uik})^{z_{uik}}$
- $\sum_{u=1}^{U} \log \prod_{k=1}^{K} q(\theta_{uk} \mid \text{shp}_{uk}, \text{rte}_{uk}) = \sum_{u=1}^{U} \log \prod_k \frac{rte_{uk}^{\ shp_{uk}} . \theta_{uk}^{\ shp_{uk}-1} . e^{-\theta_{uk} . rte_{uk}}}{\Gamma(shp_{uk})}$

Cộng các thành phần tương ứng ta thu được:

$$
\begin{aligned}
f = & -\sum_{u=1,i=1}^{U,I} \beta_i^T \theta_u + \sum_{u=1,i=1}^{U,I} \sum_k z_{uik} \log \beta_{ik} \theta_{uk} - \sum_{u=1,i=1}^{U,I} \sum_k \log z_{uik} \\
& + \sum_{u=1}^{U} \sum_k (c-1) \log \theta_{uk} - \sum_{u=1}^{U} \sum_k d . \theta_{uk} \\
& + \sum_{u=1,i=1}^{U,I} \sum_k \log z_{uik} - \sum_{u=1,i=1}^{U,I} \sum_k z_{uik} \log \phi_{uik} \\
& - \sum_{u=1}^{U} \sum_k \left[(shp_{uk} - 1) \log \theta_{uk} + shp_{uk} . \log rte_{uk} - \theta_{uk} . rte_{uk} \right] + \sum_{u=1}^{U} \sum_k \log \Gamma(shp_{uk}) \\
= & -\sum_{u=1,i=1}^{U,I} \beta_i^T \theta_u + \sum_{u=1,i=1}^{U,I} \sum_k z_{uik} \log \beta_{ik} \theta_{uk} + \sum_{u=1}^{U} \sum_k (\text{c} - \text{shp}_{uk}) \log \theta_{uk} \\
& - \sum_{u=1}^{U} (d - rte_{uk}) \sum_k \theta_{uk} - \sum_{u=1,i=1}^{U,I} \sum_k z_{uik} \log \phi_{uik} - \sum_{u=1}^{U} \sum_k shp_{uk} \log rte_{uk} + \sum_{u=1}^{U} \sum_k \log \Gamma(shp_{uk}) + const
\end{aligned}
$$

Tiếp tục, ta có:

$$
\begin{aligned}
E_{q(\theta,z)}(\text{f}) = & -\sum_{u=1}^{U} \sum_{i=1}^{I} \beta_i \sum_k \frac{shp_{uk}}{rte_{uk}} + \sum_{u=1}^{U} \sum_{i=1}^{I} \sum_{k=1}^{K} r_{ui} \phi_{uik} \log \beta_{ik} \\
& - \sum_{u=1}^{U} \sum_{i=1}^{I} \sum_{k=1}^{K} r_{ui} \phi_{uik} \log(\phi_{uik}) - \sum_{u=1}^{U} \sum_{k=1}^{K} shp_{uk} . \log(rte_{uk}) + \sum_{u=1}^{U} \sum_{k=1}^{K} \log\left(\Gamma(shp_{uk})\right) \\
& + \sum_{u=1}^{U} \sum_{k=1}^{K} \left(\sum_{i=1}^{I} r_{ui} \phi_{uik} + c - shp_{uk} \right) \left(\Psi(shp_{uk}) - \log(rte_{uk}) \right) \\
& - \sum_{u=1}^{U} \sum_{k=1}^{K} (d - rte_{uk}) \frac{shp_{uk}}{rte_{uk}} \\
= & \sum_{u=1}^{U} \sum_{i=1}^{I} \sum_{k=1}^{K} r_{ui} \phi_{uik} \log \beta_{ik} - \sum_{u=1}^{U} \sum_{i=1}^{I} \sum_{k=1}^{K} r_{ui} \phi_{uik} \log(\phi_{uik}) + \sum_{u=1}^{U} \sum_{k=1}^{K} (rte_{uk} - d - \sum_{i=1}^{I} \beta_{ik}) \frac{shp_{uk}}{rte_{uk}} \\
& + \sum_{u=1}^{U} \sum_{k=1}^{K} \left(\sum_{i=1}^{I} r_{ui} \phi_{uik} + c - shp_{uk} \right) \left(\Psi(shp_{uk}) - \log(rte_{uk}) \right) \\
& - \sum_{u=1}^{U} \sum_{k=1}^{K} shp_{uk} \log(rte_{uk}) + \sum_{u=1}^{U} \sum_{k=1}^{K} \log(\Gamma(shp_{uk})) + const
\end{aligned}
$$

Khi đó, hàm l thu được là:

$$l(\beta,\phi,shp,rte,\mathrm{W}) = -\sum_{i=1}^{I}\frac{\lambda}{2}\|\beta_i - f(\mathrm{E},\mathrm{a}_i;\mathrm{W})\|_2^2 + \sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K} r_{ui}\phi_{uik}\log\beta_{ik} - \sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K} r_{ui}\phi_{uik}\log\left(\phi_{uik}\right)$$
$$+\sum_{u=1}^{U}\sum_{k=1}^{K}(rte_{uk} - d - \sum_{i=1}^{I}\beta_{ik})\frac{shp_{uk}}{rte_{uk}}$$
$$+\sum_{u=1}^{U}\sum_{k=1}^{K}\left(\sum_{i=1}^{I} r_{ui}\phi_{uik} + c - shp_{uk}\right)\left(\Psi\left(shp_{uk}\right) - \log\left(rte_{uk}\right)\right)$$
$$-\sum_{u=1}^{U}\sum_{k=1}^{K} shp_{uk}\log(rte_{uk}) + \sum_{u=1}^{U}\sum_{k=1}^{K}\log(\Gamma\left(shp_{uk}\right)) + const$$

Để tìm được cực trị của hàm l với các biến, ta sử dụng phương pháp coordinate ascent bằng cách tối ưu từng biến dựa trên cố định các biến còn lại.

- **Cập nhật** β

Đặt $\eta_i = f(E,a_i;\mathrm{W})$, khi đó ta có:

$$l(\beta) = -\sum_{i}^{I}\frac{\lambda}{2}\|\beta_i - \eta_i\|_2^2 + \sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K} r_{ui}\phi_{uik}\log\left(\beta_{ik}\right) - \sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K}\beta_{ik}\frac{shp_{uk}}{rte_{uk}}$$
$$= \sum_{i=1}^{I}(-\frac{\lambda}{2}\|\beta_i - \eta_i\|_2^2 + \sum_{u=1}^{U}\sum_{k=1}^{K} r_{ui}\phi_{uik}\log\left(\beta_{ik}\right) - \sum_{u=1}^{U}\sum_{k=1}^{K}\beta_{ik}\frac{shp_{uk}}{rte_{uk}})$$

Xét cho mỗi item i thu được: $\frac{\partial l(\beta_i)}{\partial(\beta_i)} = \lambda\left(\eta_i - \beta_i\right) + \sum_{u=1}^{U}\sum_{k=1}^{K}\frac{r_{ui}\phi_{uik}}{\beta_{ik}} - \sum_{u=1}^{U}\sum_{k=1}^{K}\frac{shp_{uk}}{rte_{uk}}$

Cho $\frac{\partial l(\beta_i)}{\partial(\beta_i)} = 0$

$$\Leftrightarrow \sum_{k=1}^{K}\left[\lambda\left(\eta_{ik} - \beta_{ik}\right) + \sum_{u=1}^{U}\frac{r_{ui}\phi_{uik}}{\beta_{ik}} - \sum_{u=1}^{U}\frac{shp_{uk}}{rte_{uk}}\right] = 0$$

Xét đối với mỗi thuộc tính k của β_i: $-\lambda{\beta_{ik}}^2 + (\lambda\eta_{ik} - \sum_{u=1}^{U}\frac{shp_{uk}}{rte_{uk}})\beta_{ik} + \sum_{u=1}^{U} r_{ui}\phi_{uik} = 0$

Giải phương trình bậc 2, ta được: $\beta_{ik} = \frac{\lambda\eta_{ik} - \sum_{u=1}^{U}\frac{shp_{uk}}{rte_{uk}} + \sqrt{\Delta}}{2\lambda}$ (4.2)

trong đó: $\Delta = \left(\lambda\eta_{ik} - \sum_{u=1}^{U}\frac{shp_{uk}}{rte_{uk}}\right)^2 + 4\lambda\sum_{k=1}^{K} r_{ui}\phi_{uik}$

- **Cập nhật ϕ**

Xét hàm: $l(\phi)=\sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K}r_{ui}\phi_{uik}\log(\beta_{ik})-\sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K}r_{ui}\phi_{uik}\log(\phi_{uik})$

$$+\sum_{u=1}^{U}\sum_{k=1}^{K}\left(\sum_{i=1}^{I}r_{ui}\phi_{uik}\right)\left(\Psi(shp_{uk})-\log(rte_{uk})\right)$$

$$=\sum_{u=1}^{U}\sum_{i=1}^{I}\sum_{k=1}^{K}\left[r_{ui}\phi_{uik}\log(\beta_{ik})-r_{ui}\phi_{uik}\log(\phi_{uik})+r_{ui}\phi_{uik}\left(\Psi(shp_{uk})-\log(rte_{uk})\right)\right]$$

Với điều kiện: $\sum_{k=1}^{K}\phi_{uik}=1$, sử dụng phương pháp nhân từ Lagrange ta được:

$$\phi_{uik}=\begin{cases}0 \text{ nếu } r_{ui}=0\\ \dfrac{\exp\{\log\beta_{ik}+\psi(shp_{uk})-\log(rte_{uk})\}}{\sum_{k=1}^{K}\exp\{\log\beta_{ik}+\psi(shp_{uk})-\log(rte_{uk})\}} \text{ ngược lại.}\end{cases} \quad (4.3)$$

trong đó ψ là hàm Digamma.

- **Cập nhật shp, rte**

Xét hàm:

$$l(shp)=\sum_{u=1}^{U}\sum_{k=1}^{K}\left(rte_{uk}-d-\sum_{i=1}^{I}\beta_{ik}\right)\frac{shp_{uk}}{rte_{uk}}$$

$$+\sum_{u=1}^{U}\sum_{k=1}^{K}\left(\sum_{i=1}^{I}r_{ui}\phi_{uik}+c-shp_{uk}\right)\Psi(shp_{uk})+\sum_{u=1}^{U}\sum_{k=1}^{K}\log\left(\Gamma(shp_{uk})\right)$$

$$=\sum_{u=1}^{U}\sum_{k=1}^{K}\left((\sum_{i=1}^{I}r_{ui}\phi_{uik}+c-shp_{uk})\Psi(shp_{uk})+\left(rte_{uk}-d-\sum_{i=1}^{I}\beta_{ik}\right)\frac{shp_{uk}}{rte_{uk}}+\log\left(\Gamma(shp_{uk})\right)\right)$$

Và hàm:

$$l(rte)=\sum_{u=1}^{U}\sum_{k=1}^{K}\left(rte_{uk}-d-\sum_{i=1}^{I}\beta_{ik}\right)\frac{shp_{uk}}{rte_{uk}}-\sum_{u=1}^{U}\sum_{k=1}^{K}\left(\sum_{i=1}^{I}r_{ui}\phi_{uik}+c-shp_{uk}\right)\log(rte_{uk})$$

$$-\sum_{u=1}^{U}\sum_{k=1}^{K}shp_{uk}\log rte_{uk}$$

$$=\sum_{u=1}^{U}\sum_{k=1}^{K}\left(\left(rte_{uk}-d-\sum_{i=1}^{I}\beta_{ik}\right)\frac{shp_{uk}}{rte_{uk}}-\left(\sum_{i=1}^{I}r_{ui}\phi_{uik}+c-shp_{uk}\right)\log(rte_{uk})-shp_{uk}\log rte_{uk}\right)$$

Lấy đạo hàm lần lượt theo shp_{uk} , rte_{uk} và cho bằng 0 ta được:

$$\begin{cases} \dfrac{\partial l(shp_{uk})}{\partial(shp_{uk})} = \left(rte_{uk} - d - \sum_{i=1}^{I} \beta_{ik} \right) \dfrac{1}{rte_{uk}} + (\sum_{i=1}^{I} r_{ui}\phi_{uik} + c - shp_{uk}) \dfrac{\partial \Psi(shp_{uk})}{\partial(shp_{uk})} = 0 \\ \dfrac{\partial l(rte_{uk})}{\partial(\mathrm{rte}_{uk})} = \left(rte_{uk} - d - \sum_{i=1}^{I} \beta_{ik} \right) \dfrac{-shp_{uk}}{rte_{uk}^{\ 2}} - (\sum_{i=1}^{I} r_{ui}\phi_{uik} + c - shp_{uk}) \dfrac{1}{rte_{uk}} = 0 \end{cases}$$

Dễ nhận thấy một cặp nghiệm thỏa hai phương trình này là:

$$\begin{cases} shp_{uk} = c + \sum_{i=1}^{I} r_{ui}\phi_{uik} & (4.4) \\ rte_{uk} = d + \sum_{i=1}^{I} \beta_{ik} & (4.5) \end{cases}$$

- **Cập nhật trọng số W cho mạng neural**

Xét hàm:

$$l(W) = -\sum_{i=1}^{I} \frac{\lambda}{2} \| \beta_i - f(E, a_i; \mathrm{W}) \|_2^2$$

Bài toán này được cập nhật bằng cách sử dụng stochastic gradient ascent với đầu vào là $x_i = (E, a_i)$ và đầu ra là β_i

Như vậy ta có thuật toán học cho mô hình PFEP

Input: Dữ liệu quan sát được R, E, $\{a_i\}_{i=1}^{I}$ và các siêu tham số λ,c,d

Output: Ước lượng các giá trị W, β , ϕ , shp , rte , θ

Repeat

For i=1: I **do**

Cập nhật β_i bằng công thức (4.2)

End for

For u=1: U, k=1: K **do**

Cập nhật ϕ_{uik} bằng công thức (4.3)

Cập nhật shp_{uk} bằng công thức (4.4)

Cập nhật rte_{uk} bằng công thức (4.5)

End for

Cập nhật W bằng stochastic gradient ascent

Until hội tụ

4.2.2 Thuật toán học loại bỏ

Như em đã trình bày ở trên, việc áp dụng học loại bỏ sẽ giảm bớt đi các trường hợp thông tin không mang ý nghĩa mà mô hình phải mô hình hóa chúng. Cụ thể, trong ma trận tương tác người dùng sản phẩm thường rất thưa do trong thực tế số lượng sản phẩm mà một người dùng nào đó đánh giá sẽ chiếm số lượng rất nhỏ trên tổng số các sản phẩm có trong hệ thống. Như vậy đối với mỗi người dùng có rất nhiều sản phẩm có giá trị tương tác bằng 0, một là người dùng đó không thích sản phẩm đó và hai là sản phẩm đó chưa được người dùng đó biết đến. Việc loại bỏ một cách ngẫu nhiên sẽ làm giảm bớt các trường hợp không thực sự tốt mà mô hình cần phải mô hình hóa.

Ở mỗi vòng lặp sẽ loại bỏ tương tác của mỗi người dùng từ dữ liệu tương tác ban đầu với một tỷ lệ loại bỏ nhất định, khi một tương tác bị loại bỏ thì sẽ không được dùng vào việc học mô hình. Như vậy, em có thuật toán học loại bỏ PFEP-Dropout như sau:

Input: Dữ liệu quan sát được R, E, $\{a_i\}_{i=1}^{I}$ và các siêu tham số λ,c,d

Output: Ước lượng các giá trị W, $\beta, \phi, shp, rte, \theta$

Repeat

For u=1: U **do**

Loại bỏ ngẫu nhiên tương tác người dùng u từ danh sách các tương tác ban đầu với tỷ lệ loại bỏ drop_rate

End for

For i=1: I **do**

Cập nhật β_i bằng công thức (4.2)

End for

For u=1: U, k=1: K **do**

Cập nhật ϕ_{uik} bằng công thức (4.3)

Cập nhật shp_{uk} bằng công thức (4.4)

Cập nhật rte_{uk} bằng công thức (4.5)

End for

Cập nhật W bằng stochastic gradient ascent

Until hội tụ

Chương 5: Thử nghiệm và đánh giá

Trong chương này em sẽ thử nghiệm, đánh giá hiệu quả của các mô hình trên các bộ dữ liệu với mô tả ngắn và mô tả trung bình. Đưa ra các độ đo được sử dụng và một số nhận xét rút ra được sau khi đánh giá mô hình với các bộ dữ liệu khác nhau.

5.1 Thử nghiệm

Để đánh giá khả năng gợi ý cũng như hiệu quả của mô hình, em sử dụng bộ dữ liệu CiteUlike, MovieLens 1M, MovieLens 10M và MovieLens 20M. Đồng thời em cũng có so sánh hiệu quả của mô hình đối với các mô hình khác:

- WMF: Phân rã ma trận dựa trên phân phối Gauss.
- HPF: Mô hình phân cấp dựa trên phân rã Poisson.
- CTMP: Mô hình phân rã ma trận Poisson kết hợp LDA để học biểu diễn thông tin sản phẩm.

Thông tin cấu hình của thiết bị được sử dụng trong quá trình thử nghiệm:

Tên	Thông tin
HĐH	Ubuntu 16.04
CPU	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
RAM	8GB
HDD	500GB

Bảng 1: Cấu hình phần cứng sử dụng

Thông tin thêm về các bộ dữ liệu có thể xem tại:

- MovieLens 1M, 10M và 20M: https://grouplens.org/datasets/movielens/.
- CiteUlike: http://www.citeulike.org/faq/data.adp.

5.2 Tập dữ liệu sử dụng

Bộ dữ liệu	#người dùng	#sản phẩm	Độ dài mô tả
CiteUlike	5551	16980	66.6
MovieLens 1M	6040	3681	4.7
MovieLens 10M	69878	10681	5.3
MovieLens 20M	138493	26744	5.5

Bảng 2: Thông tin bộ dữ liệu sử dụng

Từ ma trận tương tác giữa người dùng và sản phẩm, đối với mỗi người dùng dữ liệu đánh giá được chia ngẫu nhiên làm hai phần dùng cho học và kiểm thử mô hình với tỷ lệ tương ứng là 7:3. Chia như vậy 5 lần, ta được 5 cặp dữ liệu học và kiểm thử cho mỗi

bộ dữ liệu. Việc học sẽ được đánh giá trên bộ dữ liệu train, mô hình thu được sau đó sẽ được kiểm thử trên bộ dữ liệu test. Độ đo cuối cùng của hệ thống được tính bằng trung bình trên 5 lần chạy.

5.3 Độ đo sử dụng

Để đánh giá hiệu năng của các mô hình trên các bộ dữ liệu khác nhau, em sử dụng hai độ đo là độ chính xác (Precision) và độ bao phủ (Recall), hai độ đo này được dùng khá phổ biến trong đánh giá hiệu quả của một mô hình gợi ý.

Đối với mỗi người dùng, em tính độ Precision top M (Precision@M) và Recall top M (Recall@M) như sau:

$$precision_M^u = \frac{số\ item\ trong\ tập\ test\ của\ user\ xuất\ hiện\ trong\ top\ M}{M}$$

$$recall_M^u = \frac{số\ item\ trong\ tập\ test\ xuất\ hiện\ trong\ top\ M}{tổng\ số\ item\ trong\ tập\ test\ của\ user}$$

Và độ **precision** và **recall** cho toàn bộ hệ thống được tính bởi trung bình của precision và recall trên toàn bộ tất cả các người dùng.

$$precision@M = \frac{\sum_{u=1}^{U} precision_M^u}{U}$$

$$recall@M = \frac{\sum_{u=1}^{U} recall_M^u}{U}$$

5.4 Thiết lập tham số

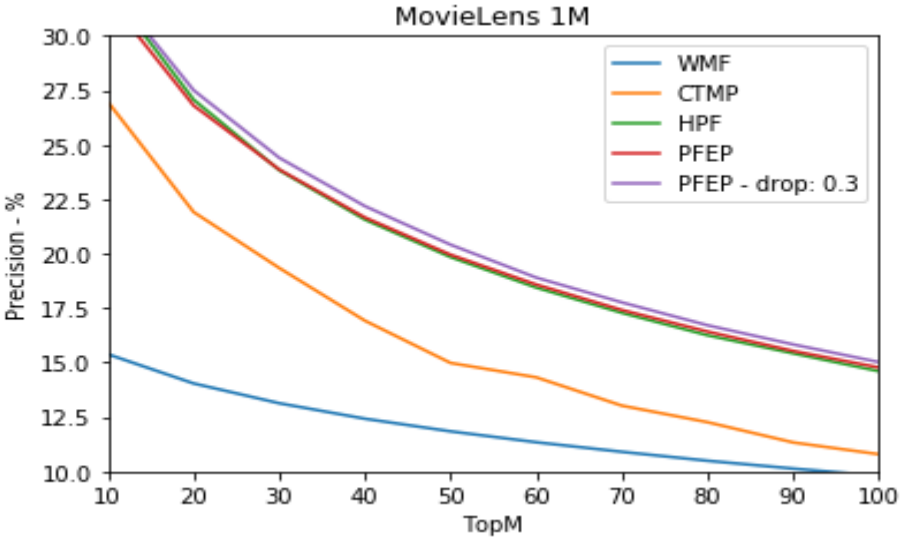
Để đánh giá hiệu quả của mô hình so với các mô hình khác, ở đây số lượng chiều vector ẩn biểu diễn cho người dùng sản phẩm em chọn K = 100. Các tham số của các mô hình khác được chọn sao cho đem lại kết quả tốt nhất đối với từng mô hình. Cụ thể, đối với từng mô hình như sau:

- WMF: các hệ số cho phân rã Gauss cho cả user và item: $\lambda_u = \lambda_i = 0.01$
- HPF: các tham số tiên nghiệm a = a’ = c = c’ = 0.3 và b’ = d’ = 1
- CTMP: tham số tiên nghiệm cho phân phối Poisson: c = d = 0.3 và hệ số Gauss cho phân biểu diễn nội dung sản phẩm là: 1
- PFEP: Hệ số Gauss cho biểu diễn nội dụng item là: 0.01 và hệ số sử dụng cho tiên nghiệm Poisson là 0.01, đồng thời với PFEP có khảo sát với drop-out là 0.3 khi so sánh với các mô hình khác.
- Top-K: top K để đánh giá mô hình: {10,20,30,40,50,60,70,80,90,100}

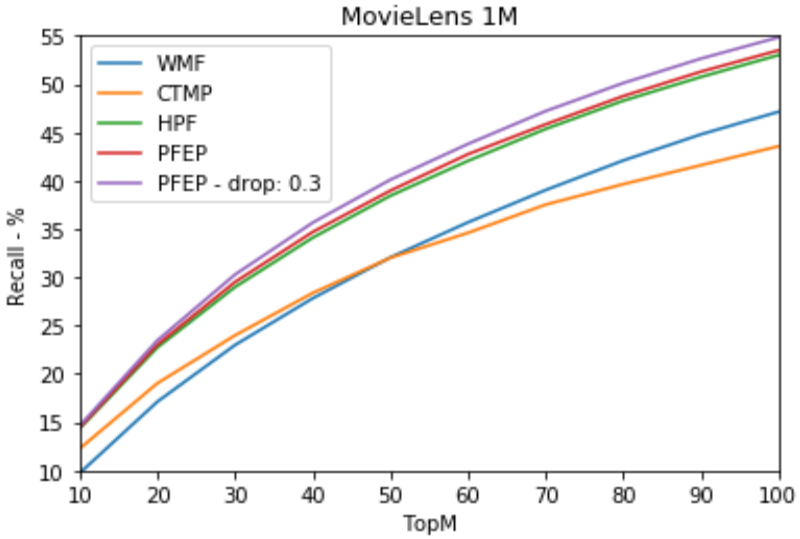
5.5 Kết quả thực nghiệm

5.5.1 Chất lượng mô hình trên bộ dữ liệu mô tả ngắn

Hình dưới mô tả chất lượng của mô hình trên các bộ mô tả ngắn bao gồm: MovieLens 1M, MovieLens 10M và MovieLens 20M. Hình 16, 17 biểu diễn chất lượng các mô hình trên bộ dữ liệu MovieLens 1M.

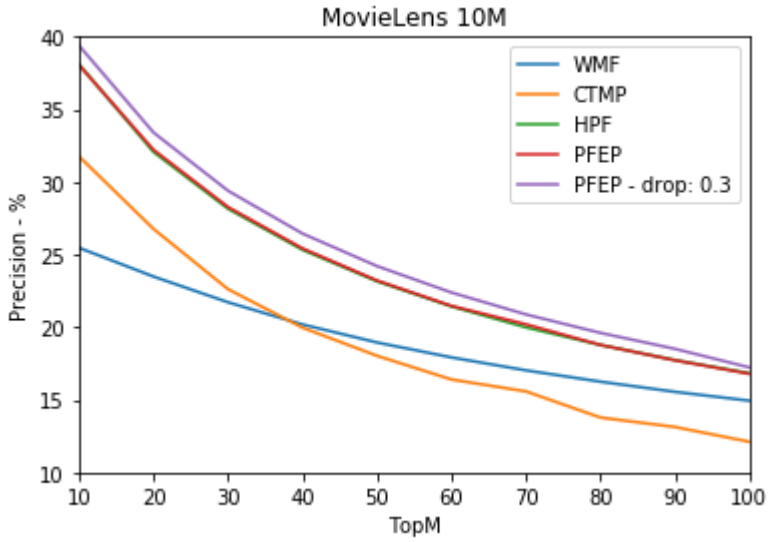


Hình 16: Độ chính xác của mô hình trên bộ dữ liệu MovieLens 1M

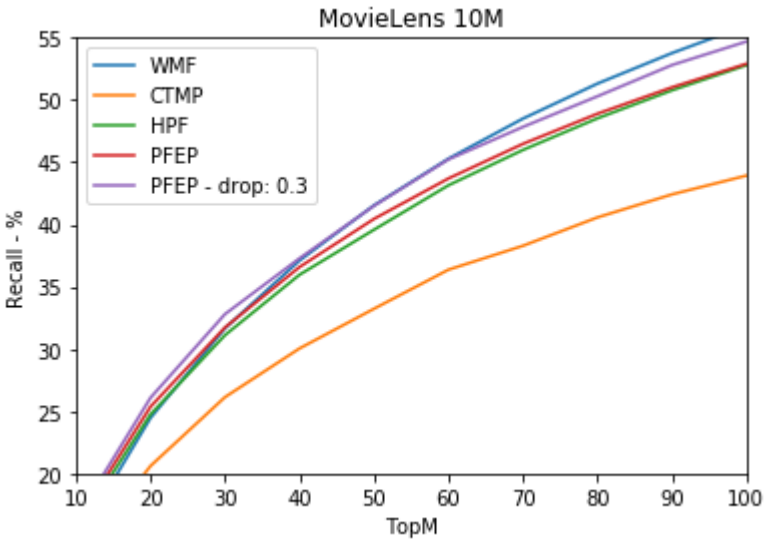


Hình 17: Độ bao phủ của mô hình trên bộ dữ liệu MovieLens 1M

Hình 18, 19 biểu diễn chất lượng của các mô hình trên bộ dữ liệu MovieLens 10M.

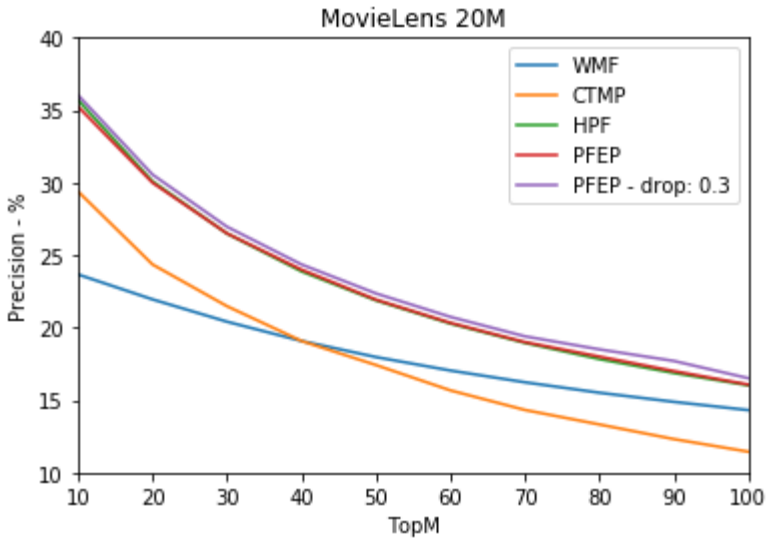


Hình 18: Độ chính xác của mô hình trên bộ dữ liệu MovieLens 10M

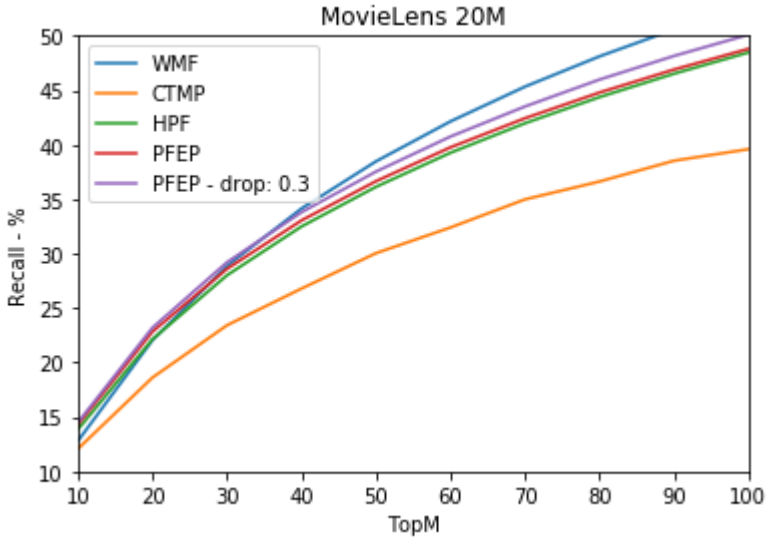


Hình 19: Độ bao phủ của mô hình trên bộ dữ liệu MovieLens 10M

Hình 20, 21 biểu diễn chất lượng của mô hình trên bộ dữ liệu MovieLens 20M.



Hình 20: Độ chính xác của mô hình trên bộ dữ liệu MovieLens 20M



Hình 21: Độ bao phủ của mô hình trên bộ dữ liệu MovieLens 20M

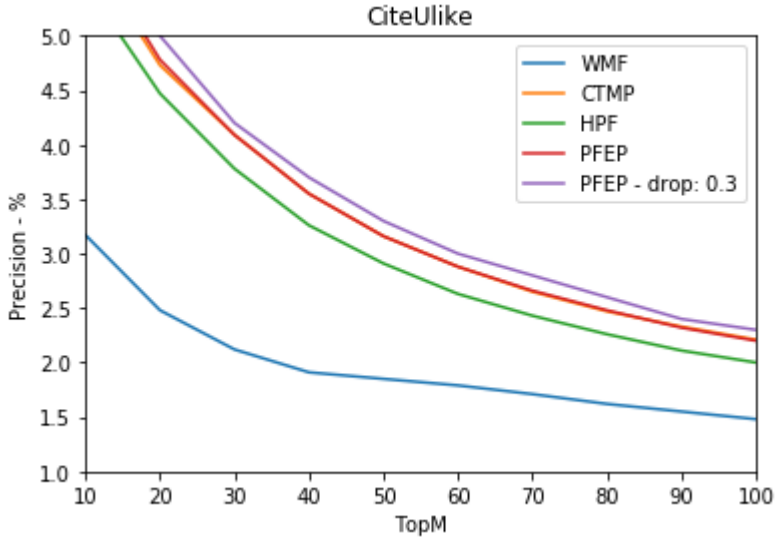
Như các hình trên ta có thể nhận thấy có sự khác biệt giữa các mô hình sử dụng phân rã ma trận Gauss và phân rã ma trận Poisson, tuy nhiên chưa có sự khác biệt rõ rệt giữa các phương pháp dựa trên phân rã mã trận Poisson. Ta có thể thấy rằng các mô hình phân rã Poisson đem lại hiệu quả khá tốt so với các mô hình dựa trên phân rã Gauss, điều này có thể chứng minh rằng việc coi giá trị tương tác người dùng, sản phẩm tuân theo phân phối rời rạc phù hợp hơn là tuân theo một phân phối liên tục và việc trọng số của biểu diễn các vector thuộc tính của người dùng, sản phẩm là các số nguyên dương sẽ hợp lý hơn là các số nguyên khi lấy theo phân phối Gauss.

Mặt khác, ta thấy rằng mô hình CTMP có sử dụng thêm thông tin mô tả item không đem lại kết quả tốt hơn so với mô hình chỉ dựa trên phân rã ma trận Poisson là HPF. Điều này có thể giải thích lý giải rằng LDA không thực sự hiệu quả khi biểu diễn cho dữ liệu văn bản ngắn.

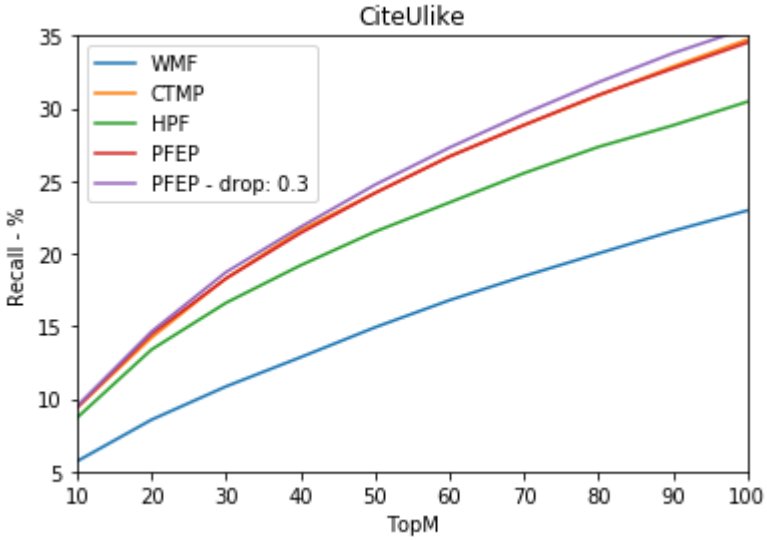
Nhưng nếu sử dụng thông tin của sản phẩm thông qua một mạng neural với bộ tri thức tiên nghiệm lại đem lại hiệu quả tốt hơn mặc dù đều dùng chung phân rã ma trận Poisson. Như vậy ta có thể kết luận rằng việc khai thác thông tin với mô tả ngắn một cách không hiệu quả sẽ làm giảm đi chất lượng gợi ý của mô hình.

5.5.2 Chất lượng mô hình trên bộ dữ liệu mô tả thông thường

Hình 22, 23 mô tả chất lượng của mô hình trên bộ dữ liệu với mô tả thông thường: CiteUlike.



Hình 22: Độ chính xác của mô hình trên bộ dữ liệu CiteUlike



Hình 23: Độ bao phủ của mô hình trên bộ dữ liệu CiteUlike

Hình trên mô tả kết quả gợi ý cho sản phẩm với nội dung mô tả thông thường, dễ dàng nhận thấy rằng việc khai thác thông tin sản phẩm trong trường hợp này đem lại hiệu quả tốt hơn nhiều so với mô hình không khai thác nội dung sản phẩm. Điều đó được

lý giải rằng việc mô tả của sản phẩm là một văn bản dài sẽ chứa nhiều thông tin quan trọng hơn và khi biết cách khai thác nó sẽ nâng cao chất lượng cho mô hình.

Cụ thể, mô hình CTMP lại có hiệu quả tốt hơn so với mô hình phân cấp Poisson HPF trái ngược lại đối với trường hợp mô tả ngắn, điều này có thể hiểu rằng LDA sẽ thực sự hiệu quả khi sử dụng trên văn bản dài so với văn bản ngắn. PFEP đem lại hiệu quả khá tốt, kết quả ngang với sử dụng LDA để biểu diễn thông tin sản phẩm trong mô hình CTMP. Đồng thời ta cũng nhận thấy học loại bỏ PFEP lại cải tiến thêm so với không sử dụng học loại bỏ, điều đó chứng tỏ rằng việc ngẫu nhiên loại bỏ đi các tương tác đối với mỗi người dùng thực sự đem lại kết quả tốt hơn.

5.5.3 Khảo sát mô hình với các giá trị siêu tham số thay đổi

Phần này em có khảo sát sự thay đổi chất lượng của mô hình vào kích thước vector ẩn, và sự ảnh hưởng của tỷ lệ học loại bỏ đến chất lượng và thời gian học của mô hình PFEP.

Sự phụ thuộc của mô hình vào kích thước chiều ẩn

Hình 25, 26, 27, 28 thể hiện sự thay đổi chất lượng của mô hình PFEP khi kích thước chiều ẩn K thay đổi trên bộ dữ liệu với mô tả ngắn MovieLens 1M và bộ dữ liệu với mô tả trung bình CiteUlike.

Ta thấy trên bộ dữ liệu với mô tả ngắn MovieLens 1M, chất lượng mô hình có sự thay đổi khi K tăng nhưng sự biến động là không đáng kể. Thậm chí với số chiều K quá lớn như tại K bằng 200 hoặc 300, chất lượng mô hình thậm chí bị giảm xuống.

Còn trên bộ dữ liệu với mô tả thông thường ta nhận thấy chất lượng mô hình tăng lên khi K tăng. Từ đó ta có thể nhận xét rằng, đối với dữ liệu có mô tả ngắn lượng thông tin ít nên giá trị mô hình không ảnh hưởng quá nhiều vào số chiều biểu diễn, nếu K lớn có thể gây ra overfitting làm giảm chất lượng mô hình. Còn đối với trên bộ dữ liệu có mô tả dài hơn, việc sử dụng K cao có thể biểu diễn tốt hơn thông tin trong sản phẩm từ đó giúp cải thiện chất lượng của mô hình.

Sự phụ thuộc của mô hình vào tỷ lệ loại bỏ drop_rate

Từ danh sách các bảng dưới cho ta thấy chất lượng mô hình PFEP có sự thay đổi khi tỷ lệ học loại bỏ drop_rate thay đổi. Đối với bộ dữ liệu mô tả ngắn, chất lượng mô hình biến thiên không đáng kể khi tỷ lệ drop_rate tăng dần. Nhưng đối với bộ dữ liệu mô tả trung bình, khi giá trị drop_rate tăng mô hình cũng cải thiện đáng kể, tuy nhiên ta thấy chất lượng mô hình sẽ giảm đi khi tỷ lệ drop_rate càng lớn, điều đó chứng tỏ trong dữ liệu với mô tả dài chứa nhiều thông tin quan trọng hơn và khi loại bỏ quá nhiều sẽ làm ảnh hưởng tới chất lượng của mô hình.

- **MovieLens 1M**

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	31.73	26.8	23.86	21.66	19.95	18.58	17.41	16.42	15.53	14.77
PFEP-0.1	32.15	27.29	24.16	21.94	20.02	18.79	17.61	16.59	15.69	14.92
PFEP-0.2	32.34	27.44	24.31	22	20.28	18.86	17.66	16.64	15.76	15
PFEP-0.3	32.36	27.5	24.4	22.18	20.41	18.9	17.76	16.71	15.83	15.02
PFEP-0.4	31.4	26.76	23.79	21.62	19.94	18.5	17.4	16.42	15.56	14.79

Bảng 3: Độ chính xác của mô hình trên bộ dữ liệu MovieLens 1M

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	14.46	23.04	29.52	34.69	38.97	42.74	45.83	48.75	51.28	53.49
PFEP-0.1	14.72	23.58	30.15	35.43	39.84	43.51	46.77	49.6	52.1	54.3
PFEP-0.2	14.8	23.65	30.21	35.5	39.79	43.64	46.96	49.79	52.34	54.64
PFEP-0.3	14.66	23.48	30.29	35.66	40.1	43.8	47.2	50.1	52.63	54.83
PFEP-0.4	14.44	23.27	29.73	34.96	39.44	43.14	46.54	49.49	52.1	54.33

Bảng 4: Độ bao phủ của mô hình trên bộ dữ liệu MovieLens 1M

- **MovieLens 10M**

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	38.05	32.22	28.28	25.43	23.22	21.47	20.02	18.78	17.72	16.79
PFEP-0.1	38.94	32.24	28.33	25.5	23.3	21.55	20.1	18.84	17.8	16.84
PFEP-0.2	37.9	32.02	28.11	25.26	23.08	21.34	19.89	18.67	17.61	16.69
PFEP-0.3	39.4	33.41	29.4	26.45	24.21	22.39	20.88	19.61	18.51	17.23

Bảng 5: Độ chính xác của mô hình trên bộ dữ liệu MovieLens 10M

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	15.92	25.36	31.73	36.58	40.45	43.7	46.48	48.9	51.02	52.91
PFEP-0.1	15.84	25.37	31.85	36.7	40.6	43.84	46.64	49.07	51.23	53.16
PFEP-0.2	15.97	25.39	31.86	36.74	40.72	40.03	46.88	49.36	51.53	53.47
PFEP-0.3	16.32	26.08	32.8	37.28	41.5	45.23	47.84	50.3	52.81	54.7

Bảng 6: Độ bao phủ của mô hình trên bộ dữ liệu MovieLens 10M

- **Movie Lens 20M**

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	35.25	30	26.5	24	21.93	20.33	19	18	17	16.07
PFEP-0.1	35.57	30.11	26.6	24.1	22.09	20.44	19.12	18.1	17.06	16.17
PFEP-0.2	35.87	30.42	26.79	24.18	22.2	20.59	19.25	18.11	17.14	16.28
PFEP-0.3	36.04	30.56	26.96	24.36	22.36	20.74	19.4	18.25	17.27	16.41

Bảng 7: Độ chính xác của mô hình trên MovieLens 20M

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	14.31	22.84	28.6	33.05	36.65	39.76	42.43	44.79	46.9	48.8
PFEP-0.1	14.27	22.68	28.63	33.2	36.86	40	42.7	45	47.07	49.02
PFEP-0.2	14.39	23.03	28.92	33.48	37.23	40.4	43.11	45.5	47.63	49.54
PFEP-0.3	14.5	23.15	29.13	33.75	37.53	40.74	43.51	45.94	48.12	50.1

Bảng 8: Độ bao phủ của mô hình trên MovieLens 20M

- **CiteUlike**

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	5.89	4.78	4.09	3.55	3.16	2.88	2.66	2.48	2.32	2.2
PFEP-0.1	6.01	4.9	4.21	3.64	3.23	2.94	2.71	2.53	2.37	2.23
PFEP-0.2	6.14	4.97	4.25	3.7	3.31	3.01	2.77	2.58	2.42	2.28
PFEP-0.3	5.9	4.77	4.11	3.59	3.2	2.91	2.68	2.49	2.35	2.22
PFEP-0.4	6	4.9	4.19	3.63	3.23	2.94	2.72	2.54	2.39	2.26

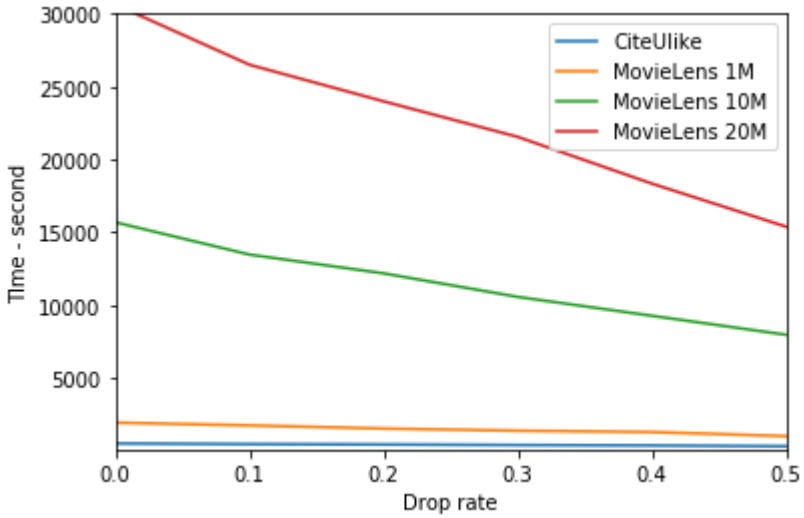
Bảng 9: Độ chính xác của mô hình trên CiteUlike

	Top 10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
PFEP	9.41	14.47	18.28	21.41	24.17	26.68	28.85	30.92	32.74	34.53
PFEP-0.1	9.24	14.37	18.36	21.37	24.11	26.7	29	31	32.78	34.38
PFEP-0.2	9.49	14.59	18.54	21.71	24.54	26.98	29.22	31.27	33.1	34.82
PFEP-0.3	9.22	14	17.8	21.14	23.82	26.3	28.34	30.3	32.18	33.88
PFEP-0.4	9.1	14.37	18.36	21.53	24.03	26.48	28.6	30.53	32.45	34.27

Bảng 10: Độ bao phủ của mô hình trên CiteUlike

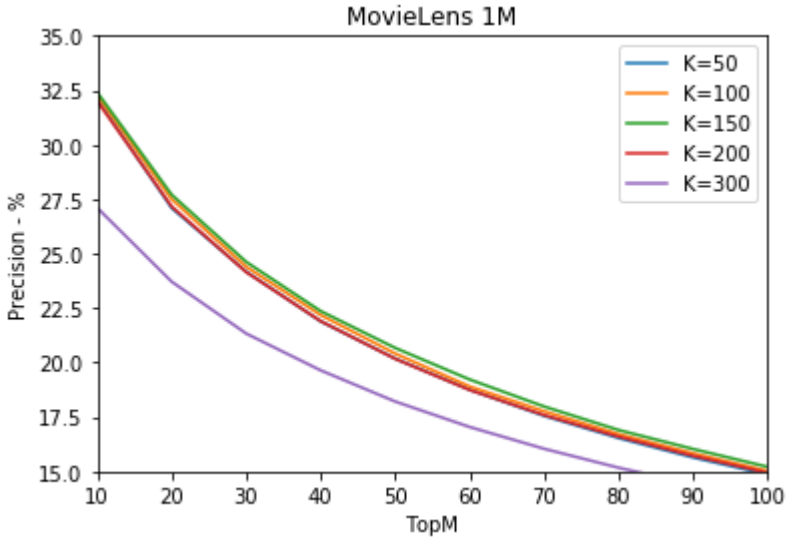
Sự phụ thuộc tốc độ học của mô hình vào tỷ lệ loại bỏ drop_rate

Phần này em sẽ khảo sát về thời gian học của mô hình với các tỷ lệ drop_rate khác nhau. Nhìn vào hình bên dưới ta có thể thấy rằng, khi việc tăng tỷ lệ loại bỏ cũng sẽ dẫn đến thời gian học cho mô hình sẽ nhanh hơn so với không sử dụng drop_rate. Và đặc biệt hữu ích với những bộ dữ liệu lớn như MovieLens 10M và MovieLens 20M.

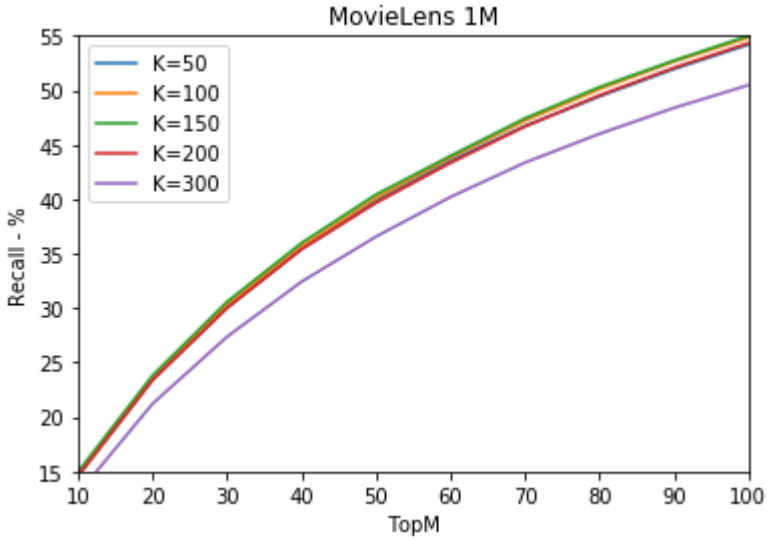


Hình 24: Thời gian học của mô hình trên các bộ dữ liệu với các tỷ lệ loại bỏ khác nhau

Hình 25, 26: Chất lượng gợi ý của mô hình với các biểu diễn số chiều ẩn khác nhau trên bộ dữ liệu với mô tả ngắn.

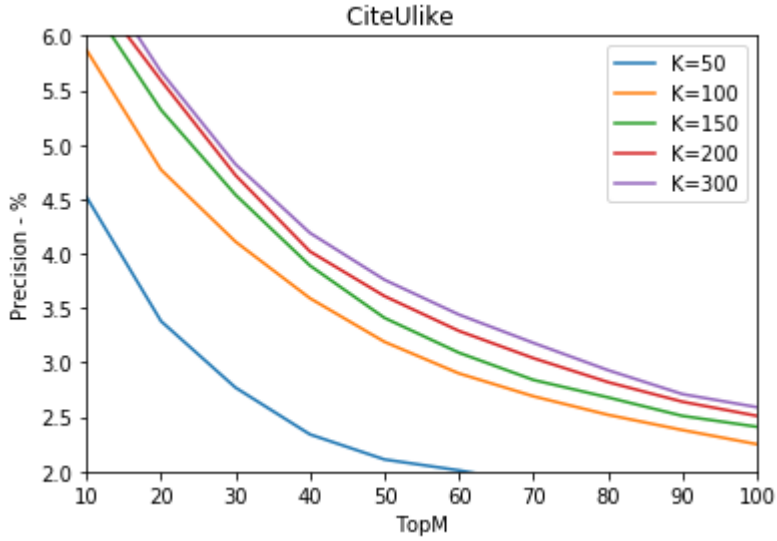


Hình 25: Độ chính xác mô hình với số chiều ẩn thay đổi trên MovieLens 1M

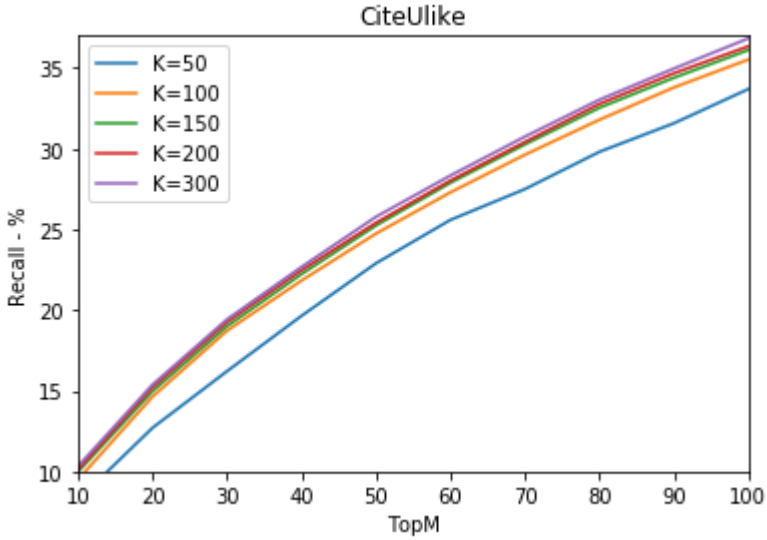


Hình 26: Độ bao phủ của mô hình đối với số chiều ẩn thay đổi trên MovieLens 1M

Hình 27, 28: Chất lượng gợi ý của mô hình với kích thước vector ẩn khác nhau trên bộ dữ liệu mô tả trung bình.



Hình 27: Độ chính xác của mô hình với số chiều ẩn thay đổi trên CiteUlike



Hình 28: Độ bao phủ của mô hình với số chiều ẩn thay đổi trên CiteUlike

Chương 6: Kết luận

Chương này tóm tắt lại kết quả đã đạt được, các kiến thức thu hoạch được khi thực hiện đồ án và các hướng nghiên cứu tiếp theo cho bài toán.

Thông qua đồ án này em có thể hiểu được các bài toán gợi ý nói chung, ưu và nhược điểm của các phương pháp nói riêng. Cụ thể đối với các mô hình phân rã Gauss sẽ đem lại hiệu quả thấp hơn so với các mô hình phân rã sử dụng Poisson, điều này chứng tỏ rằng phân phối Poisson là phù hợp hơn khi làm việc với bộ dữ liệu rời rạc và việc biểu diễn các vector thuộc tính người dùng, sản phẩm là các số nguyên dương trong phân rã Poisson sẽ hợp lý hơn đối với trong phân rã sử dụng phân phối Gauss.

Hiệu quả của việc sử dụng thông tin mô tả cho biểu diễn sản phẩm trước khi đưa vào phân rã ma trận, tuy nhiên việc khai thác bộ dữ liệu mô tả không hợp lý có thể dẫn đến làm giảm kết quả của mô hình. Ví dụ, mô hình CTMP khi sử dụng trên bộ dữ liệu mô tả ngắn đem lại hiệu quả thấp hơn so với mô hình phân cấp Poisson, nhưng mô hình PFEP khai thác nội dung mô tả sản phẩm thông qua một bộ tri thức tiên nghiệm kết hợp với một mạng neural đem lại hiệu quả cao hơn trong trường hợp này. Điều này chứng tỏ rằng việc khai thác thông tin cần lựa chọn một cách hợp lý, phù hợp với đặc trưng của từng bộ dữ liệu.

Việc sử dụng học loại bỏ thực sự đem lại kết quả tốt hơn so với học không loại bỏ, không chỉ vậy học loại bỏ còn giúp tiết kiệm tài nguyên tính toán và tăng tốc độ học cho mô hình. Tuy nhiên đối với từng bộ dữ liệu và đặc trưng của chúng mà cần lựa chọn một tỷ lệ thích hợp tránh làm mất mát quá nhiều thông tin trong bộ dữ liệu ảnh hưởng tới chất lượng mô hình.

Trong các mô hình này mới chỉ khai thác thông tin từ sản phẩm trước khi đưa vào phân rã ma trận, tuy nhiên chúng ta hoàn toàn có thể khai thác thêm các thông tin khác từ phía người dùng để xây dựng vector đặc trưng cho người dùng, các thông tin này có thể như: tuổi, sở thích, giới tính...

TÀI LIỆU THAM KHẢO

[1] Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009).

[2] Koren, Y., Bell, R., Volinsky, C., et al.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009).

[3] P. Gopalan, L. Charlin, D.M. Blei. *Content-based Recommendation with Poisson factorization*. NIPS 2014, Montreal, CA, Dec 2014.

[4] D. Agarwal and B. Chen. fLDA: *Matrix factorization through latent Dirichlet allocation*. In Proceedings of the third ACM international conference on web search and data mining, pages 91–100. ACM, 2010.

[5] Gopalan, P.K., Charlin, L., Blei, D.: Content-based recommendations with poisson factorization. In: Advances in Neural Information Processing Systems. pp. 3176–3184 (2014).

[6] D. Blei, A. Ng, and M. Jordan. *latent Dirichlet allocation*. Journal of Machine Learning Research, 3:993–1022, January 2003.

[7] A. Cemgil. *Bayesian inference for nonnegative matrix factorization models*. Computational Intelligence and Neuroscience, 2009.

[8] D. B Dunson and A. H. Herring. *Bayesian latent variable models for mixed discrete outcomes*. Biostatistics, 6(1):11–25, 2005.

[9] P. Gopalan, J.M. Hofman, and D. Blei. *Scalable recommendation with Poisson factorization*. ArXiv preprint arXiv:1311.1704, 2013.

[10] Y. Hu, Y. Koren, and C. Volinsky. *Collaborative filtering for implicit feedback datasets*. In Eighth IEEE International Conference on Data Mining., pages 263–272. IEEE, 2008.

[11] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. *Introduction to variational methods for graphical models*. Machine Learning, 37:183–233, 1999.

[12] H. Shan and A. Banerjee**.** *Generalized probabilistic matrix factorizations for collaborative filtering*. In Data Mining (ICDM), 2010 IEEE 10th International

Conference on, pages 1025–1030. IEEE, 2010.
[13] C. Wang and D. Blei. *Collaborative topic modeling for recommending scientific articles*. In Knowledge Discovery and Data Mining, 2011.

[14] Alexander Felfernig, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer, and Martin Stettinger. Basic approaches in recommendation systems. In *Recommendation Systems in Software Engineering*, 2014.

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009.

[16] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM ’10, pages 1445–1448, New York, NY, USA, 2010. ACM.

[17] Li C., Duan Y., Wang H., Zhang Z., Sun A., Ma Z.: Enhancing topic modeling for short texts with auxiliary word embeddings. ACM Transactions on Information Systems (TOIS) 36(2), 11 (2017).

[18] Lin C., He Y.: Joint sentiment/topic model for sentiment analysis. In: ACM Conference on Information and Knowledge Management. pp. 375–384 (2009).

[19] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014).

[20] Le H.M., Cong S.T., Van Linh N., Than K.: Collaborative topic model for poisson distributed ratings. International Journal of Approximate Reasoning 95, 62–76 (2018).