TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Nguyễn Duy Vũ
Nguyễn Chiêu Bản

GIẢI QUYẾT VẤN ĐỀ DỮ LIỆU BỊ THIÊN LỆCH TRONG BÀI TOÁN ĐỀ XUẤT SẢN PHẨM BẰNG MÔ HÌNH HỌC DỰA TRÊN ĐỘ ĐO IPS

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 06/2022

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Nguyễn Duy Vũ - 18120264
Nguyễn Chiêu Bản - 18120283

GIẢI QUYẾT VẤN ĐỀ DỮ LIỆU BỊ THIÊN LỆCH TRONG BÀI TOÁN ĐỀ XUẤT SẢN PHẨM BẰNG MÔ HÌNH HỌC DỰA TRÊN ĐỘ ĐO IPS

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN

Th.S. Trần Trung Kiên
TS. Nguyễn Ngọc Thảo

Tp. Hồ Chí Minh, tháng 06/2022

Nhận xét hướng dẫn

Nhận xét phản biện

Lời cảm ơn

Chúng em xin chân thành gửi lời cảm ơn sâu sắc đến thầy Trần Trung Kiên và cô Nguyễn Ngọc Thảo. Thầy và cô đã rất tận tâm, nhiệt tình hướng dẫn và chỉ bảo nhóm chúng em trong suốt quá trình thực hiện khóa luận. Cảm ơn tất cả những lời góp ý và nhận xét của thầy cô để giúp khóa luận của chúng em hoàn thành tốt nhất.

Chúng em cũng xin phép gửi lời cảm ơn đến quý thầy cô của trường Đại học Khoa học Tự nhiên nói chung và khoa Công nghệ Thông tin nói riêng vì đã tận tình chỉ dạy và truyền đạt những kiến thức, kinh nghiệm quý báu cho chúng em. Sự trưởng thành của chúng em có được hôm nay chính là nhờ phần lớn ở công dạy dỗ của các thầy cô.

Chúng em xin chân thành cảm ơn gia đình, bạn bè và mọi người xung quanh đã tạo điều kiện thuận lợi để nhóm chúng em hoàn thành khóa luận.

Lời cuối cùng, chúng em xin kính chúc quý thầy/cô có sức khỏe thật tốt, luôn bình an và hạnh phúc trong cuộc sống.

Nhóm chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 24 tháng 6 năm 2022

Nhóm sinh viên thực hiện

Nguyễn Duy Vũ

Nguyễn Chiêu Bản

fit@hcmus

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG KHÓA LUẬN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG GỢI Ý SẢN PHẨM DỰA TRÊN MÔ HÌNH HỌC MỐI QUAN HỆ NHÂN QUẢ

(Causal learning models for building recommendation system)

1 THÔNG TIN CHUNG

Người hướng dẫn:

- ThS. Trần Trung Kiên (Khoa Công nghệ Thông tin)
- TS. Nguyễn Ngọc Thảo (Khoa Công nghệ Thông tin)

Nhóm sinh viên thực hiện:

1. Nguyễn Duy Vũ (MSSV: 18120264)
2. Nguyễn Chiêu Bản (MSSV: 18120283)

Loại đề tài: Nghiên cứu

Thời gian thực hiện: Từ *1/2022* đến *6/2022*

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Bài toán gợi ý sản phẩm được phát biểu như sau:

1

- Đầu vào là đánh giá của người dùng đối với các sản phẩm trước đó trong hệ thống.
- Yêu cầu: đưa ra các sản phẩm trong hệ thống mà phù hợp với sở thích của người dùng.

Nếu giải quyết được bài toán gợi ý sản phẩm thì sẽ giúp cho trải nghiệm của người dùng được cá nhân hóa, giúp các công ty tạo ra được nhiều lợi thế cạnh tranh cũng như là kích thích nhu cầu mua sắm của người dùng. Người dùng thay vì hao phí thời gian vào việc lựa chọn sản phẩm thích hợp cho bản thân, thì họ sẽ tập trung vào việc trải nghiệm sản phẩm. Bài toán gợi ý sản phẩm là một bài toán không dễ vì nó phải đối mặt với vấn đề thiếu dữ liệu của người dùng để hệ thống hoạt động một cách hiệu quả, hoặc phải đối mặt với việc thay đổi sở thích của người dùng.

Trong thời gian gần đây, một hướng tiếp cận có nhiều tiềm năng phát triển trong bài toán gợi ý sản phẩm là sử dụng mô hình học nhân quả (causal learning model). Và đây là hướng tiếp cận nhóm chúng em chọn để tìm hiểu.

2.2 Mục tiêu đề tài

- Hiểu rõ tình hình nghiên cứu của bài toán gợi ý sản phẩm theo hướng tiếp cận học nhân quả (Các mô hình nào được đề xuất để giải quyết bài toán trong thời gian gần đây? Các ưu, nhược điểm của các mô hình? Các vấn đề lớn của bài toán mà mô hình giải quyết?). Từ đó, nhóm chúng em sẽ chọn ra một mô hình tốt, có tiềm năng phát triển trong tương lai (và khả thi để có thể hoàn thành trong thời lượng của khóa luận) để tiến hành tập trung tìm hiểu sâu hơn.
- Hiểu rõ lý thuyết của mô hình học mối quan hệ nhân quả đã chọn (trên cơ sở hiểu rõ lý thuyết nền tảng về học mối quan hệ nhân quả).
- Cài đặt lại mô hình mà bài báo đề xuất có được kết quả tương tự với kết quả trong bài báo. Nhóm có thể tiến hành thêm các thí nghiệm ngoài bài báo để thấy rõ hơn về ưu/nhược điểm của mô hình.

- Trên cơ sở đã hiểu rõ mô hình, nếu còn thời gian thì nhóm có thể tiến hành xem xét các cải thiện có thể có để nâng cao độ hiệu quả của mô hình.
- Rèn luyện được các kỹ năng mềm: làm việc nhóm, lên kế hoạch, kỹ năng viết và trình bày khóa luận,...

2.3 Phạm vi của đề tài

Đề tài sử dụng 2 tập dữ liệu lớn là MovieLens10M và Netflix, đây là 2 tập dữ liệu thường xuyên được sử dụng trong việc đánh giá các mô hình trong bài toán gợi ý sản phẩm. Tập dữ liệu bao gồm đánh giá của người dùng đối với các bộ phim khác nhau. Về cơ bản, đề tài chỉ tập trung vào việc tìm hiểu và cài đặt lại mô hình của một bài báo uy tín. Ngoài ra, nhóm chúng em có thể có thêm các thí nghiệm ngoài bài báo để thấy rõ hơn về ưu/nhược điểm của mô hình. Lý do chúng em giới hạn đề tài như vậy là vì:

(i) Mô hình học nhân quả là một mô hình khá mới, vì vậy chúng em thấy chỉ riêng việc hiểu rõ mô hình (và các kiến thức nền tảng bên dưới) và có thể tự cài đặt lại đã tốn rất nhiều thời gian.

(ii) Chúng em xác định là chỉ trên cơ sở hiểu rõ mô hình (và các kiến thức nền tảng bên dưới) thì mới có thể có được các cải tiến thật sự trong tương lai, cũng như là mới có thể vận dụng được mô hình cho các bài toán khác.

Tất nhiên, trong khóa luận, nếu có đủ thời gian nhóm chúng em sẽ thử đề xuất và cài đặt các cải tiến; tuy nhiên, đây không phải là mục tiêu chính của nhóm em.

2.4 Cách tiếp cận dự kiến

Hiện nay, để xây dựng một hệ thống gợi ý tự động, người ta thường sử dụng ba phương pháp cơ bản: lọc dựa trên nội dung (Content-base filtering - CB), lọc cộng tác (Collaborative Filtering - CF), và phương pháp kết hợp (Hybrid Filtering).

- Phương pháp lọc dựa trên nội dung (Content-base filtering - CB) học mối tương quan giữa các sản phẩm với sản phẩm bằng cách tạo cho mỗi người dùng một hồ sơ cá nhân dựa trên các đặc điểm của sản phẩm mà người dùng

đã đánh giá. Từ đó, mô hình sẽ chọn ra sản phẩm tương đồng nhất và gợi ý cho người dùng.

- Phương pháp lọc cộng tác (Collaborative Filtering - CF) học mối tương quan giữa người dùng với người dùng dựa vào những hành vi của người dùng trong quá khứ, sau đó gợi ý những sản phẩm mà được yêu thích bởi những người dùng tương đồng.
- Phương pháp kết hợp (Hybrid Filtering) sử dụng cả hai kỹ thuật lọc cộng tác và lọc dựa trên nội dung.

Điểm chung của các phương pháp truyền thống trên là dựa vào những đánh giá hoặc phản hồi của người dùng để suy diễn ra sở thích của họ. Các mô hình này được huấn luyện và đánh giá trên tập dữ liệu quan sát được và do đó chúng mang một giả định ngầm rằng các đánh giá bị thiếu thì không mang lại thông tin hữu ích, nói cách khác việc thiếu dữ liệu đánh giá của các sản phẩm là ngẫu nhiên (Missing At Random - MAR) [1]. Tuy nhiên, bài báo [2] cho thấy rằng những dữ liệu về đánh giá sản phẩm trực tuyến thường sẽ mắc phải một số thiên lệch. Ví dụ như trong dữ liệu về đánh giá phim, các phản hồi quan sát được là kết quả của việc người dùng tự lựa chọn phim để xem, và do đó, nó dựa trên sở thích và việc lựa chọn trước đó của người dùng nên số lượng đánh giá tốt về phim sẽ vượt trội so với các đánh giá tệ, ngoài ra những người dùng cực thích hoặc cực sẽ có khả năng đánh giá nhiều hơn, do đó có ít đánh giá phim ở mức trung bình. Vậy nên sự vắng mặt của những phản hồi cũng có thể mang đến thông tin hữu ích. Nếu dùng dữ liệu này để đánh giá mô hình thì sẽ không đúng với thực tế khi triển khai mô hình, hiểu được điều này, nhiều mô hình nhân quả đã được nghiên cứu để kiểm tra ảnh hưởng của các yếu tố gây nhiễu không được quan sát bằng cách đo lường ảnh hưởng của hệ thống gợi ý hay nói cách khác là những thay đổi trong hành động của người dùng dưới sự tác động của hệ thống gợi ý.

Một trong những nghiên cứu nổi bật là Recommendations as Treatments: Debiasing Learning and Evaluation của Schnabel, Tobias, Adith Swaminathan, Ashudeep Singh, Navin Chandak, và Thorsten Joachims đã được xuất bản trong tạp chí Pro-

ceedings of Machine Learning Research đồng thời cũng được công bố trong hội nghị International Conference on Machine Learning [3]. Trong nghiên cứu này, các tác giả đã đưa ra một phương pháp để giảm thiểu tác động của việc dữ liệu bị lệch trong quá trình học, qua đó tối ưu hóa mô hình, đồng thời, đánh giá mô hình sau khi học. Để thực hiện được điều này, đầu tiên, các tác giả đã cho thấy được cách ước lượng chất lượng của hệ thống gợi ý sử dụng phương pháp ước lượng điểm xu hướng theo trọng số (propensity-weighting) - một phương pháp thường được sử dụng trong suy diễn nhân quả. Tiếp theo, từ phương pháp ước lượng đó, các tác giả đề xuất phương pháp ERM (Empirical Risk Minimization) cho hệ thống gợi ý, có thể hiểu là giảm thiểu rủi ro trên tập dữ liệu quan sát được, phương pháp này giúp hệ thống gợi ý có thể học được trên dữ liệu bị thiên lệch. Kế đến, sử dụng phương pháp ERM trên để tìm ra phương pháp phân rã ma trận (Matrix Factorization - MF). Cuối cùng, tác giả chỉ ra các phương pháp để ước tính xu hướng trong dữ liệu quan sát được, trong đó, xu hướng là những thiên lệch lựa chọn do việc người dùng tự lựa chọn.

Sau đó, Bonner, Stephen và Vasile đã cải tiến phương pháp trên để tạo ra một hình mới trong bài báo Causal embeddings for recommendation[4], không những có khả năng giải quyết được vấn đề thiên lệch lựa chọn gây ra bởi người dùng, mà còn đánh giá và giải quyết thiên lệch chọn gây ra bởi chính hệ thống gợi ý. Để làm được điều này, nhóm tác giả đã sử dụng thêm kĩ thuật ước lượng ảnh hưởng tác động các nhân (Individual Treatment Effect - ITE) [5] vào trong nghiên cứu của họ.

Gần đây đã có các kĩ thuật cải tiến cho suy diễn nhân quả mang lại nhiều ý nghĩa trong lĩnh vực xây dựng hệ thống gợi ý. Tuy nhiên, do những giới hạn về thời gian và sự hiểu biết, nhóm em sẽ tập trung tìm hiểu và sử dụng phương pháp xây dựng hệ thống gợi ý dựa trên ITE được sử dụng trong bài báo [4]. Do những kĩ thuật trong suy diễn nhân quả và cơ sở lý thuyết của nó khá phức tạp, việc hiểu và áp dụng nó vào bài toán xây dựng hệ thống gợi ý là một nhiệm vụ không dễ dàng và vừa đủ cho phạm vi của một khóa luận tốt nghiệp.

2.5 Kết quả dự kiến của đề tài

Cả lĩnh vực hệ thống gợi ý và suy diễn nhân quả đều là những lĩnh vực rộng lớn đòi hỏi nhiều kiến thức, kỹ năng cũng như kinh nghiệm. Do vậy, trong phạm vi giới hạn của khóa luận này, nhóm sẽ tập trung giải thích một số cơ sở lý thuyết nền tảng mà tác giả đã sử dụng, qua đó chứng minh được bằng cách nào mà một hệ thống gợi ý được xây dựng dựa trên suy diễn nhân quả có thể mang lại hiệu quả đáng kể so với các phương pháp truyền thống. Tiếp theo, nhóm sẽ cài đặt lại từ đầu mô hình được đề xuất trong bài báo [4]. Từ đó, nhóm sẽ có được các kết quả thí nghiệm để cho thấy mô hình tự cài đặt ra được các kết quả như trong bài báo và thấy rõ về ưu/nhược điểm của mô hình. Nếu có thời gian thì có thể cài đặt và thí nghiệm thêm các cải tiến.

2.6 Kế hoạch thực hiện

Để đạt được kết quả như trên, nhóm sẽ thực hiện theo một kế hoạch như sau:

1. Tìm hiểu về lý thuyết và các thuật ngữ, tài liệu tham khảo liên quan đến bài báo. Giai đoạn này khá quan trọng nên nhóm dành 3 tháng (từ tháng 1/2022 đến hết tháng 3/2022). Trong đó sinh viên Nguyễn Duy Vũ sẽ tìm hiểu các kiến thức liên quan đến việc xây dựng một hệ thống gợi ý và một vài phương pháp hàng đầu (state of the art) trong lĩnh vực này, sinh viên Nguyễn Chiêu Bản tìm hiểu các kiến thức về suy diễn nhân quả được nhắc tới trong bài báo gốc, đồng thời giải thích tại sao cần có Suy diễn Nhân quả trong học máy và cách mà nó làm cho một mô hình học máy trở nên đáng tin cậy hơn.

2. Nhóm dành một tháng tiếp theo (từ đầu tháng 4/2022 đến cuối tháng 4/2022) để tìm hiểu chi tiết về bài báo từ những kiến thức tìm hiểu được trong phần trước đó. Tương tự như trên, những phần lý thuyết về hệ thống gợi ý sẽ được sinh viên Nguyễn Duy Vũ tìm hiểu, phần còn lại do sinh viên Nguyễn Chiêu Bản phụ trách. Sau đó, những phần thực hiện và tìm phần hiểu được sẽ được trao đổi cho thành viên còn lại.

3. Từ những hiểu biết trên, các thành viên trong nhóm sẽ cùng nhau thực hiện

lại thực nghiệm theo phương pháp đã tìm hiểu và một vài phương pháp truyền thống để đối chứng. Quá trình này dự kiến sẽ kéo dài một tháng rưỡi (từ đầu tháng 5/2022 đến nữa đầu tháng 6/2022).

4. Song song với việc thực hiện các thử nghiệm, nhóm sẽ bắt đầu viết báo cáo từ đầu tháng 5/2022 dự kiến hoàn thành và chỉnh sửa cho đến khi kết thúc thời gian thực hiện khóa luận.

Tài liệu

[1] D. F. Heitjan and S. Basu, "Distinguishing "missing at random" and "missing completely at random"," *The American Statistician*, vol. 50, no. 3, pp. 207–213, 1996.

[2] P. A. P. Hu Nan and J. J. Zhang, "On self-selection biases in online product reviews," *MIS Q.*, vol. 40, no. 2, pp. 449–471, 2017.

[3] A. S. N. C. T. J. Tobias Schnabel, Adith Swaminathan, "Recommendations as treatments: Debiasing learning and evaluation," *Proceedings of Machine Learning Research*, vol. 48, pp. 1670–1679, 2016.

[4] S. Bonner and F. Vasile, "Causal embeddings for recommendation," *Proceedings of the 12th ACM conference on recommender systems*, pp. 104–112, 2018.

[5] D. B. Rubin, "Estimating causal effects of treatments in randomized and non-randomized studies," *Journal of Educational Psychology*, vol. 66, no. 5, pp. 688–701, 1974.

XÁC NHẬN	*TP. Hồ Chí Minh, ngày 2 tháng 4 năm 2022*
CỦA NGƯỜI HƯỚNG DẪN	**NHÓM SINH VIÊN THỰC HIỆN**
(Ký và ghi rõ họ tên)	***(Ký và ghi rõ họ tên)***
	
TRẦN TRUNG KIÊN	**NGUYỄN DUY VŨ**
	
NGUYỄN NGỌC THẢO	**NGUYỄN CHIÊU BẢN**

Mục lục

Danh sách hình

Danh sách bảng

Tóm tắt

Hầu hết các bộ dữ liệu được sử dụng để huấn luyện các mô hình đề xuất sản phẩm ngày nay đều bị thiên lệch. Việc đánh giá và huấn luyện các mô hình dựa trên bộ dữ liệu thiên lệch này sẽ cho kết quả tệ trong thực tế. Vì vậy trong khóa luận này, nhóm chúng em sẽ trình bày về phương pháp "Inverse propensity scoring" (IPS) để giải quyết vấn đề thiên lệch dữ liệu này. Phương pháp này hoạt động bằng cách đánh lại trọng số của các mẫu dữ liệu trong quá trình đánh giá và huấn luyện từ đó khắc phục được vấn đề thiên lệch của dữ liệu. Phương pháp này cho thấy sự cải thiện đáng kể về mặt hiệu suất khi so sánh với phương pháp truyền thống trên bộ dữ liệu thế giới thực.

Chương 1

Giới thiệu

Trong chương này, đầu tiên nhóm chúng em phát biểu về bài toán đề xuất sản phẩm cũng như là ý nghĩa của nó đối với cuộc sống hiện nay. Sau đó, nhóm chúng em trình bày về vấn đề thiên lệch dữ liệu, một thách thức lớn của bài toán đề xuất sản phẩm. Từ đó dẫn đến phương pháp nhóm chúng em tìm hiểu "Inverse propensity scoring" (IPS) để khắc phục vấn đề thiên lệch dữ liệu này. Ngoài ra, ở cuối chương nhóm chúng em sẽ trình bày về cách tổ chức của khóa luận.

1.1 Phát biểu và ý nghĩa của bài toán

Ngày nay, các loại sản phẩm phục vụ cho đời sống con người được sinh ra ngày một nhiều. Điều này tạo nên thách thức cho người dùng là làm thế nào để có thể tìm thấy được các sản phẩm phù hợp với bản phân giữa một lượng sản phẩm khổng lồ như vậy. Do đó hệ thống đề xuất sản phẩm đã ra đời nhằm mục đích hỗ trợ cho người dùng trong quá trình tìm kiếm nội dung phù hợp cho bản thân.

Các hệ thống đề xuất sản phẩm giúp cho trải nghiệm của người dùng được cá nhân hóa, giúp tiết kiệm thời gian cho người dùng trong việc lựa chọn sản phẩm. Mặt khác, đối với các công ty nó còn tăng cao lợi thế cạnh tranh, kích thích nhu cầu mua sắm của người dùng bằng các gợi ý

sản phẩm. Có thể kể đến các hệ thống đề xuất sản phẩm nổi tiếng như: hệ thống đề xuất phim của Netflix, hay hệ thống đề xuất video của Youtube, hệ thống đề xuất nhạc của Spotify và nhiều loại hệ thống đề xuất khác nữa.

Bài toán hệ thống đề xuất sản phẩm sẽ được phát biểu như sau (hình 1.1 minh hoạ về cách thức hoạt động của hệ thống đề xuất phim):

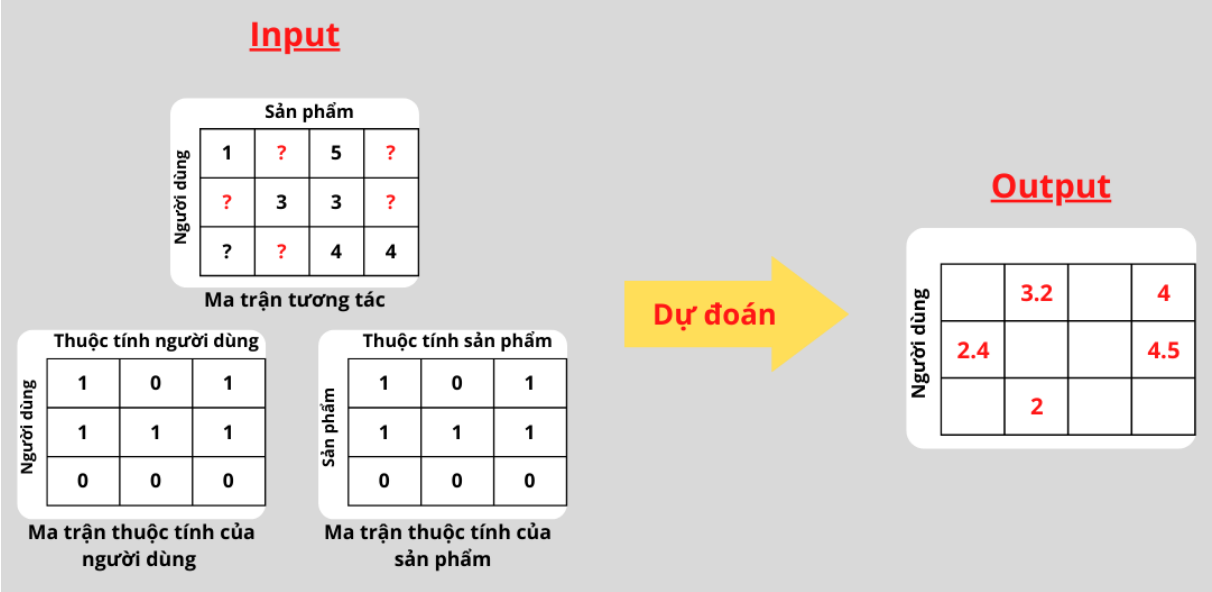
- Đầu vào là dữ liệu về điểm đánh giá của các người dùng đối với các sản phẩm trong hệ thống; dữ liệu này sẽ được đưa vào mô hình dưới dạng một ma trận tương tác thưa với mỗi dòng là các đánh giá của người dùng cho các sản phẩm. Ngoài ra, còn có thể có thêm dữ liệu về thông tin của mỗi người dùng và mỗi sản phẩm; dữ liệu này được đưa vào mô hình dưới dạng 2 ma trận thuộc tính của người dùng và sản phẩm, các điểm dữ liệu trong ma trận thuộc tính sẽ được tiền xử lý thành dạng số.

- Yêu cầu: đề xuất các sản phẩm trong hệ thống mà phù hợp với sở thích của mỗi người dùng (để có thể đề xuất thì một cách làm phổ biến là dự đoán điểm đánh giá của người dùng đối với các sản phẩm mà người dùng chưa đánh giá).

Trong phạm vi đề tài của khóa luận nhóm chúng em sẽ tiến hành xử lý trên dữ liệu explicit feedback. Explicit feedback là dữ liệu cho biết rõ ràng mức độ ưa thích của người dùng đối với sản phẩm, chẳng hạn như điểm đánh giá của người dùng cho 1 bộ phim từ 1 đến 5.

1.2 Thách thức của bài toán

Để huấn luyện được mô hình có thể đề xuất chính xác tất cả các sản phẩm cho người dùng, ta cần một bộ dữ liệu đầy đủ bao gồm đánh giá của tất cả người dùng cho tất cả sản phẩm. Tuy nhiên dữ liệu này không thể có được trong thực tế, do người dùng không thể nào xem và đánh giá

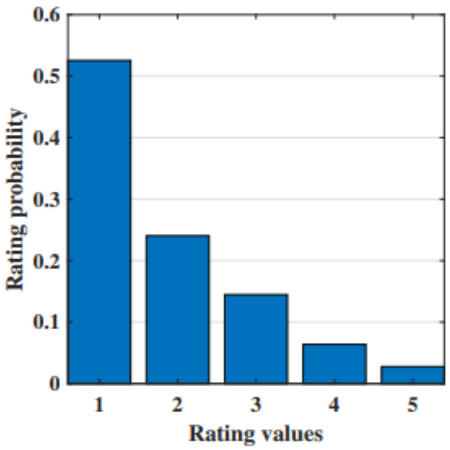


Hình 1.1: Cách thức hoạt động của hệ thống đề xuất sản phẩm.

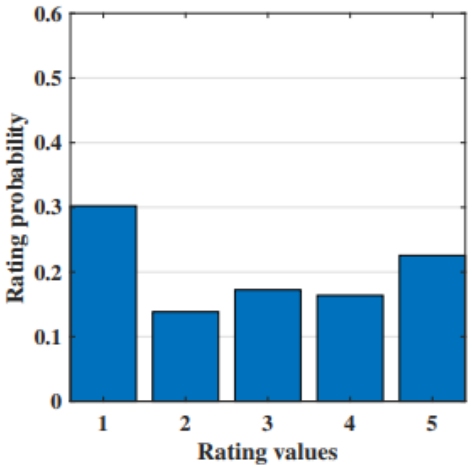
tất cả các sản phẩm trong hệ thống. Vì vậy để mô hình có thể huấn luyện được tốt nhất ta cần bộ dữ liệu quan sát được phải được phát sinh từ phân phối đều của bộ dữ liệu đầy đủ này, vì bộ dữ liệu quan sát được này sẽ đại diện cho bộ dữ liệu đầy đủ và nó được gọi là dữ liệu không bị thiên lệch.

Dữ liệu quan sát được trong bài toán đề xuất sản phẩm thường bị gặp vấn đề lớn về thiên lệch dữ liệu. Thiên lệch dữ liệu là dữ liệu quan sát được không được phát sinh từ phân phối đều, do đó nó không đại diện được cho bộ dữ liệu đầy đủ. Bắt nguồn từ 2 nguyên nhân chính trong việc thu thập dữ liệu sau:

- Do người dùng tự chọn các bộ phim để đánh giá, thay vì dựa trên việc người dùng đánh giá một tập các sản phẩm ngẫu nhiên. Việc người dùng đánh giá trên một tập các sản phẩm được đề xuất ngẫu nhiên sẽ đảm bảo dữ liệu thu được được phát sinh từ phân phối đều, giúp cho dữ liệu thu được có thể đại diện được cho tập dữ liệu đầy đủ. Một kết quả nghiên cứu của tác giả Marlin và các cộng sự [4] đã cho thấy tác động rõ ràng của thiên lệch dữ liệu. Họ đã tiến hành một cuộc khảo sát người dùng để thu thập dữ liệu điểm đánh giá của người dùng đối với một số sản phẩm được lựa chọn ngẫu nhiên, để



(a) Các sản phẩm được chọn ngẫu nhiên.



(b) Các sản phẩm được chọn do người dùng tự lựa chọn.

Hình 1.2: Phân phối điểm đánh giá sản phẩm, được thu thập bằng cách cho người dùng đánh giá các sản phẩm ngẫu nhiên và cho người dùng đánh giá các sản phẩm tự lựa chọn.

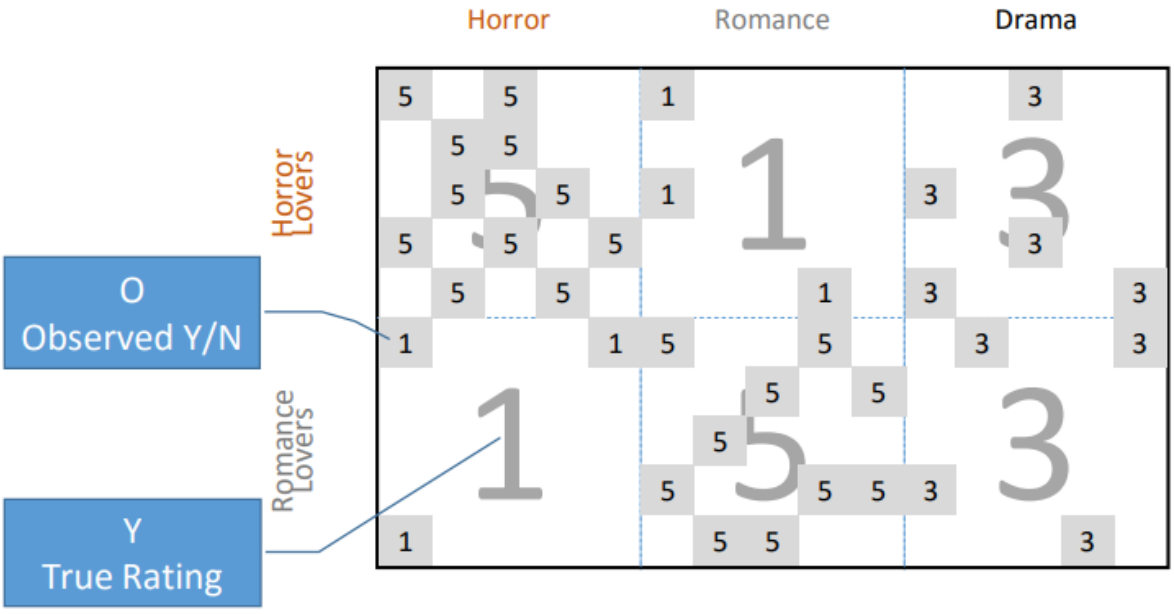
so sánh với các sản phẩm do người dùng tự lựa chọn. Hình 1.2 cho thấy rõ sự khác nhau trong phân phối dữ liệu của việc lựa chọn ngẫu nhiên và do người dùng đánh giá và đưa ra 2 phát hiện: 1) người dùng có xu hướng chọn và đánh giá các mặt hàng họ thích; và 2) người dùng có nhiều khả năng xếp hạng các mặt hàng đặc biệt xấu hoặc đặc biệt tốt.

- Do hệ thống đề xuất chỉ chọn các bộ phim nào đó để hiển thị cho người dùng, trong trường hợp hệ thống đề xuất sản phẩm đã triển khai từ trước. Với tác động của hệ thống đề xuất như vậy, người dùng chỉ có thể xem và đánh giá trên các bộ phim được hiển thị, điều này làm cho dữ liệu thu thập được chỉ tập trung vào một vài loại sản phẩm nào đó, không đại diện cho toàn bộ dữ liệu.

Để hiểu rõ hơn về vấn đề thiên lệch dữ liệu ta sẽ xem xét một ví dụ nhỏ trong hệ thống đề xuất phim, minh họa tác động của thiên lệch dữ liệu có thể gây ra cho việc đánh giá mô hình. Hình 1.3 minh họa cho thí nghiệm của ta trong đó:

- Ma trận Y là ma trận đầy đủ chứa đánh giá của tất cả người dùng đối với tất cả các sản phẩm, trong ma trận này bao gồm 2 nhóm nhỏ: 1) những người dùng yêu thích phim kinh dị, họ sẽ đánh giá 5 điểm cho tất cả các phim kinh dị và 1 điểm cho tất cả các phim lãng mãn mà họ đã xem; nhóm trái ngược 2) những người dùng yêu thích phim lãng mạn, họ sẽ đánh giá 1 điểm cho tất cả các phim kinh dị và 5 điểm cho tất cả các phim lãng mạn mà họ đã xem; và cả 2 nhóm người dùng đều đánh giá cho thể loại phim kịch là 3.

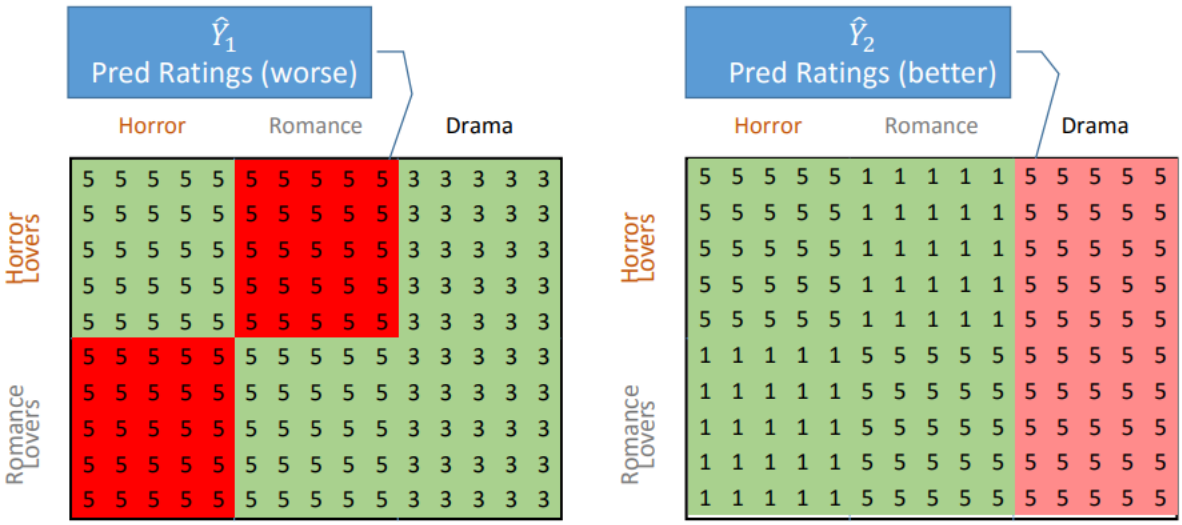
- Ma trận O là ma trận nhị phân chứa 2 giá trị 0 và 1, đại diện cho những sản phẩm người dùng đã đánh giá trong hệ thống, $[O_{u,i} = 1] \Leftrightarrow [Y_{u,i}$ được quan sát].



Hình 1.3: Ví dụ minh họa tác động của thiên lệch dữ liệu. Y là những điểm đánh giá thật của dữ liệu, minh họa bằng những con số nằm bên dưới ma trận. O là những điểm đánh giá quan sát được, minh họa bằng những điểm đánh giá của người dùng cho các sản phẩm.

Xét 2 cách dự đoán $\hat{Y}_1$ và $\hat{Y}_2$ sẽ được dùng để dự đoán giá trị cho các điểm đánh giá chưa quan sát được. Hình 1.4 sẽ minh họa cách 2 mô hình dự đoán như sau:

- Ở cách dự đoán $\hat{Y}_1$, những điểm đánh giá có giá trị thật là 1 sẽ được dự đoán là 5, các điểm đánh giá có giá trị thật là 3 và 5 sẽ được dự đoán đúng giá trị.
- Ở cách dự đoán $\hat{Y}_2$, những điểm đánh giá có giá trị thật là 3 sẽ được dự đoán là 5, các điểm đánh giá có giá trị thật là 3 và 5 sẽ được dự đoán đúng giá trị.



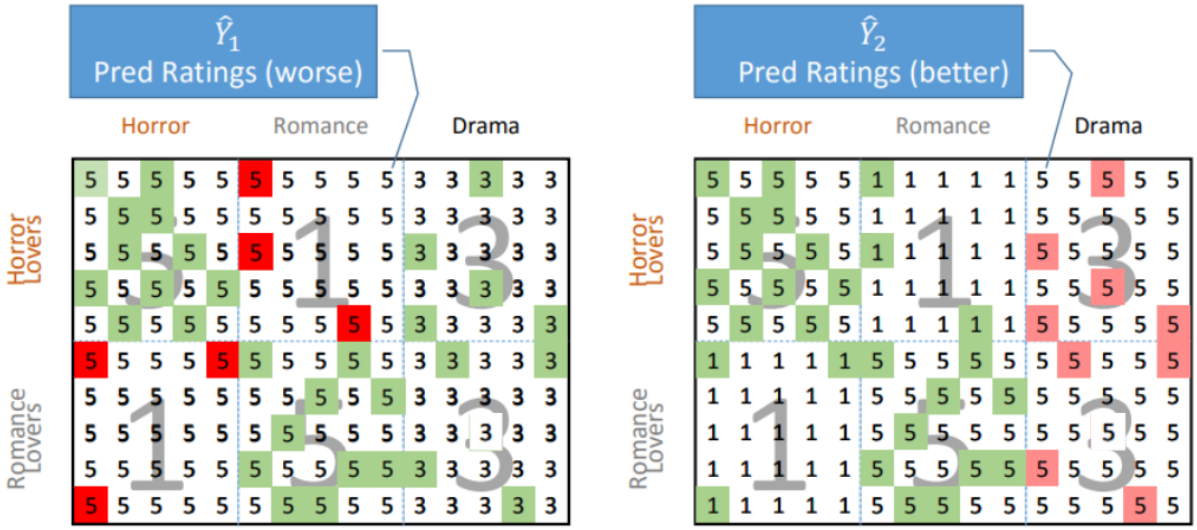
Hình 1.4: Các cách dự đoán $\hat{Y}_1$ và $\hat{Y}_2$ trên tập dữ liệu.

Nếu chúng ta có thể quan sát toàn bộ dữ liệu như trong hình 1.4, ta có thể thấy với cách dự đoán $\hat{Y}_1$ khi giá trị thật là 1 nhưng dự đoán là 5 thì độ lỗi của nó rất lớn. Khi so với cách dự đoán $\hat{Y}_2$ khi giá trị thật là 3 và dự đoán là 5 có độ lỗi không quá nghiêm trọng như cách dự đoán $\hat{Y}_1$.

Để đánh giá một cách dự đoán có tốt hay không, ta sẽ tiến hành đánh giá độ lỗi trên tập dữ liệu quan sát được. Tuy nhiên dữ liệu quan sát được ở đây lại không được phát sinh từ phân phối đều do đó nó có sự thiên lệch, cụ thể hơn là những điểm đánh giá là 1 xuất hiện ít hơn những điểm đánh giá là 3 (minh họa ở hình 1.5); điều này bắt nguồn từ việc người dùng tự lựa chọn các bộ phim họ thích để đánh giá.

Khi đánh giá cách dự đoán nào là tốt trên tập dữ liệu quan sát được bị thiên lệch này, ta sẽ cho rằng dự đoán $\hat{Y}_1$ tốt hơn $\hat{Y}_2$. Vì theo cách dự đoán

$\hat{Y}_1$ những điểm có độ lỗi lớn (điểm đánh giá thật là 1 nhưng dự đoán là 5) xuất hiện rất ít trong bộ dữ liệu quan sát được; trong khi đó theo cách dự đoán $\hat{Y}_2$ những điểm có độ lỗi nhỏ hơn (điểm đánh giá thật là 3 nhưng dự đoán là 5) xuất hiện nhiều hơn trong bộ dữ liệu quan sát được; điều này làm cho độ lỗi của cách dự đoán $\hat{Y}_1$ thấp hơn cách dự đoán $\hat{Y}_2$ mặc dù cách dự đoán $\hat{Y}_1$ tệ hơn.



Hình 1.5: Đánh giá 2 mô hình dự đoán $\hat{Y}_1$, $\hat{Y}_2$ dựa trên các mâu quan sát được.

Để khắc phục vấn đề thiên lệch dữ liệu tác giả Tobias Schnabel và các cộng sự [9] trong bài báo "Recommendations as Treatments: Debiasing Learning and Evaluation" tại hội nghị "ICML 2016", đã đề xuất việc áp dụng phương pháp "Inverse propensity scoring" (IPS) vào quá trình huấn luyện và đánh giá mô hình. IPS hoạt động bằng cách đánh lại trọng số của các mẫu dựa trên propensity, theo cách giảm trọng số của các mẫu thường quan sát được, trong khi tăng trọng số của các mẫu hiếm gặp. Điều này sẽ giúp kiểm soát được vấn đề thiên lệch của dữ liệu.

1.3 Bố cục

Phần còn lại của khóa luận sẽ được trình bày như sau:

- Chương 2 trình bày kiến thức nền tảng về "Matrix factorization", "Gradient descent", "Naive bayes" và "Logistic regression".
- Chương 3 trình bày về độ đo IPS và ứng dụng của nó trong việc đánh giá và huấn luyện mô hình; đây là phần chính của khóa luận. Trong phần này gồm có hai phần nhỏ:
  - Độ đo "Self normalized inverse propensity scoring" (SNIPS).
  - Ước lượng propensity: nhóm chúng em trình bày về cách ước lượng ma trận propensity thông qua 2 mô hình "Naive bayes" và "Logistic regression".
- Chương 4 trình bày về các thí nghiệm và các kết quả đạt được.
- Cuối cùng, tổng kết và các hướng phát triển sẽ được trình bày ở chương 5.

Chương 2

Kiến thức nền tảng

Trong chương này, đầu tiên nhóm chúng em trình bày về thuật toán "Matrix factorization" - thuật toán đề xuất sản phẩm bằng cách phân rã ma trận tương tác. Sau đó nhóm chúng em sẽ trình bày về thuật toán "Gradient descent" - thuật toán mà nhóm chúng em sẽ sử dụng để cực tiểu hóa hàm chi phí của "Matrix factorization". Ngoài ra, nhóm chúng em còn trình bày về "Naive bayes" và "Logistic regression" - hai mô hình phân lớp mà nhóm chúng em sẽ sử dụng để ước lượng propensity trong IPS. Chương này, đặc biệt là về phần "Matrix factorization" cung cấp những kiến thức nền tảng để có thể hiểu rõ về những cải tiến mà nhóm em tìm hiểu ở chương kế tiếp.

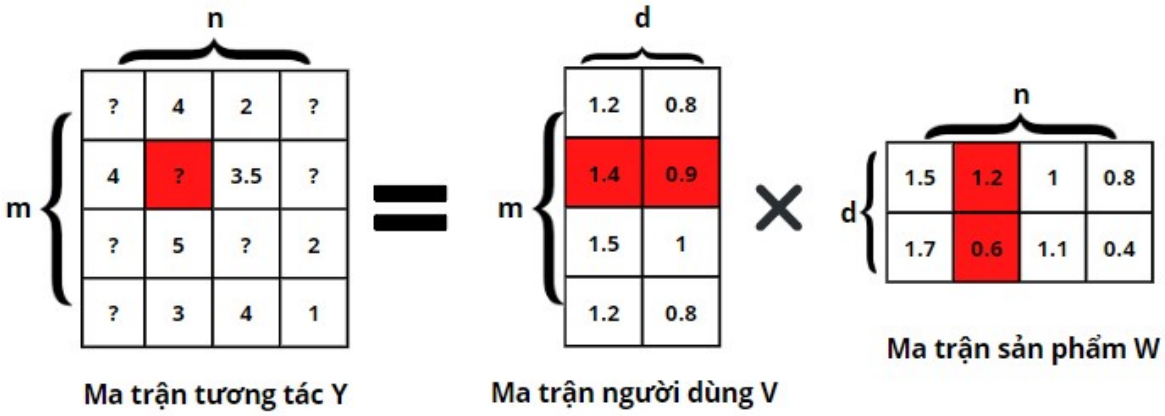
2.1 "Matrix factorization"

"Matrix factorization" là một phương pháp thuộc nhóm lọc cộng tác (collaborative filtering), một nhóm các phương pháp tập trung vào mối quan hệ giữa các người dùng, dựa trên đánh giá của các người dùng trước đó trong hệ thống. Các phương pháp lọc cộng tác này sẽ dựa trên ý tưởng những người dùng có cùng sở thích đối với một số sản phẩm nhất định, thì cũng có cùng sở thích đối với sản phẩm khác; do đó nó sẽ đề xuất sản phẩm cho người dùng dựa trên các sản phẩm mà người dùng giống với họ đã thích.

2.1.1 Hàm dự đoán của "Matrix factorization"

Phương pháp "Matrix factorization" sẽ phân tích ma trận tương tác của người dùng và sản phẩm Y thành tích của hai ma trận V và W, sao cho từ hai ma trận V và W này ta có thể xây dựng lại được ma trận Y càng chính xác càng tốt. Một cách cụ thể, ma trận tương tác dự đoán được của ta có thể ước lượng như sau (hình 2.1 minh họa phương pháp "Matrix factorization"):

$$Y \approx \hat{Y} = V \times W^T \tag{2.1}$$



Hình 2.1: Matrix Factorization. Ma trận tương tác Y sẽ được phân rã thành ma trận đại diện cho người dùng V và đại diện cho sản phẩm W.

Trong đó:

- Ma trận tương tác Y là ma trận thưa có kích thước là $m \times n$ tương ứng với m người dùng đánh giá cho n sản phẩm. Ma trận $\hat{Y}$ là ma trận dự đoán có cùng kích thước với Y.
- d là hyperparameter được điều chỉnh trong quá trình huấn luyện mô hình, đại diện cho số lượng các đặc trưng tiềm ẩn. Các đặc trưng tiềm ẩn mô tả sự liên quan giữa các người dùng và các sản phẩm. Ví dụ như trong hệ thống đề xuất phim, các đặc trưng tiềm ẩn có thể

là thể loại, ngôn ngữ, diễn viên hay bất kì các đặc trưng nào khác; hoặc có thể là bất cứ sự liên quan khác giữa người dùng và sản phẩm mà ta không cần đặt tên cụ thể.

- Ma trận V có kích thước $m \times d$ và là ma trận đại diện cho người dùng. Trong ma trận này, người dùng thứ u sẽ tương ứng với hàng thứ u trong ma trận và sẽ được kí hiệu là v_u. Các hệ số trong véc-tơ v_u sẽ đo lường mức độ yêu thích của người dùng u cho các đặc trưng ẩn, hệ số càng cao tương ứng với người dùng u càng thích các sản phẩm mang đặc trưng đó.

- Tương tự, ma trận W có kích thước $d \times n$ và là ma trận đại diện cho sản phẩm. Trong ma trận này, sản phẩm thứ i sẽ tương ứng với hàng thứ i trong ma trận và sẽ được kí hiệu là w_i. Các hệ số trong véc-tơ w_i sẽ đo lường mức độ sản phẩm i mang các đặc trưng ẩn, hệ số càng cao tương ứng với sản phẩm i mang đặc trưng đó càng lớn.

Mục tiêu của chúng ta là đề xuất cho người dùng u các sản phẩm i mang đặc trưng mà người dùng u thích, tương ứng với giá trị của v_u và w_i đều cao dẫn đến giá trị của $v_u^T \times w_i$ càng cao. Khi đó, đánh giá của người dùng u cho sản phẩm i sẽ được tính toán bằng tích của 2 véc-tơ v_u và w_i. Cụ thể như công thức sau:

$$\hat{Y}_{u,i} = v_u \times w_i^T \tag{2.2}$$

Trong thực tế, một số người dùng sẽ có thiên hướng đánh giá cao hơn các người dùng khác, hay một số sản phẩm sẽ bị đánh giá thấp hơn trung bình. Do đó đánh giá của người dùng u cho sản phẩm i cần cộng thêm các offset cho riêng từng người dùng a_u từng sản phẩm b_i và giá trị trung bình cho toàn bộ đánh giá c. Khi đó công thức 2.2 được viết lại như sau:

$$\hat{Y}_{u,i} = v_u \times w_i^T + a_u + b_i + c \tag{2.3}$$

2.1.2 Tìm các tham số của hàm dự đoán của "Matrix Factorization"

Mục tiêu chính của việc huấn luyện mô hình "Matrix Factorization" là tìm giá trị của hai ma trận V và W. Vì vậy hai ma trận này sẽ được ước lượng bằng cách cực tiểu hóa trung bình bình phương sai số giữa đánh giá dự đoán và đánh giá thực trên các đánh giá quan sát được. Với A là ma trận chứa các offset, hàm mục tiêu được định nghĩa cụ thể như sau:

$$\underset{V,W,A}{\operatorname{argmin}} \sum_{(u,i)\in\kappa} (Y_{u,i} - \hat{Y}_{u,i})^2 + \lambda(||V||_F^2 + ||W||_F^2 + A) \tag{2.4}$$

Ở đây:

- κ là tập dữ liệu chứa các đánh giá quan sát được.
- λ là tham số regularization ($0 \leq \lambda \leq 1$) và $\lambda(||V||_F^2 + ||W||_F^2)$ là regularization để tránh hiện tượng overfitting của mô hình.
- $||\cdot||_F$ là ký hiệu của chuẩn Frobenius, bằng căn bậc hai của tổng bình phương tất cả các phần tử của ma trận.

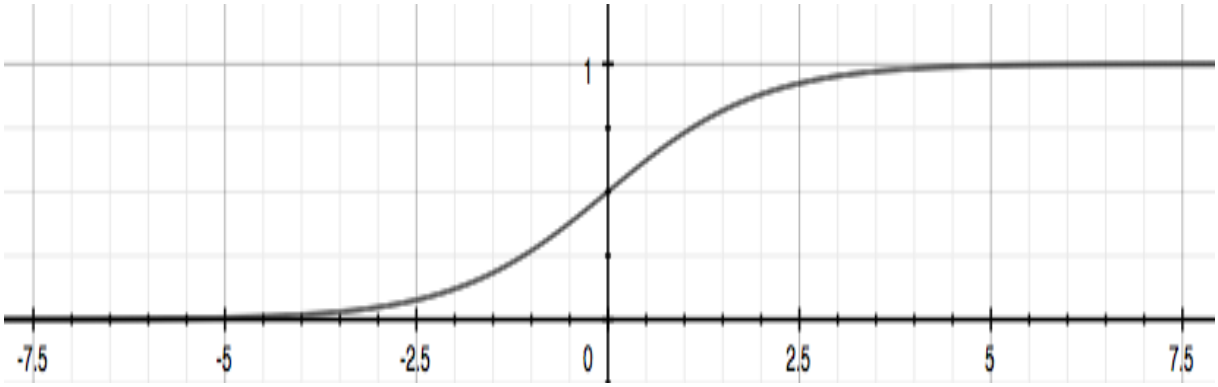
Để cực tiểu hóa hàm mục tiêu này, ta có thể sử dụng thuật toán "Gradient Descent" sẽ được trình bày ở phần 2.4.

2.2 "Logistic regression"

"Logistic regression" là một mô hình học có giám sát và được sử dụng trong bài toán phân lớp nhị phân. Mô hình này hoạt động bằng cách sử dụng hàm "sigmoid" để phân lớp các véc-tơ đầu vào. Trong khóa luận của nhóm chúng em, "Logistic regression" được sử dụng để ước lượng propensity (tiền đề cho việc tính toán độ đo IPS) thông qua các thuộc tính đặc trưng của người dùng và sản phẩm.

2.2.1 Hàm "sigmoid"

Hàm "sigmoid" là một hàm số liên tục có đạo hàm không âm tại mọi điểm. Hàm "sigmoid" sẽ ánh xạ tất cả số thực vào khoảng (0,1), điều này làm cho nó trở thành một hàm phù hợp cho bài toán phân lớp. Hình 2.2 minh họa đồ thị hàm số "sigmoid". Thông qua đồ thị ta có thể thấy khi đầu vào càng tiến đến $+\infty$ giá trị khi đi qua hàm "sigmoid" càng tiến gần 1, ngược lại, khi đầu vào càng tiến đến $-\infty$ giá trị khi đi qua hàm "sigmoid" càng tiến gần về 0.



Hình 2.2: Đồ thị hàm số số "Sigmoid" (hình vẽ được lấy từ bài giảng của GS. Andrew Ng trong khóa học "Machine Learning" ở trang coursera.org).

Hàm "sigmoid" sẽ được tính toán thông qua công thức:

$$\sigma(x) = \frac{1}{1 + e^{x}} \tag{2.5}$$

2.2.2 Hàm dự đoán của "Logistic regression"

Với một véc-tơ đầu vào $x \in \mathbb{R}^{D \times 1}$, với D là số lượng đặc trưng của véc-tơ đầu vào, hàm dự đoán $h(x)$ của "Logistic regression" sẽ trả về một giá trị nằm trong khoảng $(0, 1)$. Giá trị trả về này cho biết xác suất $p(y = 1|x)$, với y là đầu ra của véc-tơ đầu vào x. Khi đó, xác suất đầu ra y là 0 sẽ được tính toán bằng $[p(y = 0|x) = 1 - p(y = 1|x)]$. Như vậy, ta có thể biết được véc-tơ đầu vào x thuộc lớp nào dựa trên một một ngưỡng bằng 0.5,

tức là $h(x) \geq 0.5$ thì véc-tơ đầu vào x thuộc lớp 1 và ngược lại $h(x) \leq 0.5$ thì véc-tơ đầu vào x sẽ thuộc lớp 0.

Cụ thể, hàm dự đoán h(x) của "Logistic regression" như sau:

$$\begin{aligned} h(x) &= p(y = 1|x) \\ &= \sigma(W^T x) \\ &= \frac{1}{1 + e^{W^T x}} \end{aligned} \tag{2.6}$$

với W là các tham số của hàm dự đoán.

2.2.3 Tìm các tham số của hàm dự đoán của "Logistic regression"

Xác suất $p(y|x)$

Mục đích của chúng ta là tìm các tham số của hàm dự đoán W sao cho $h(x)$ càng gần với 1 càng tốt đối với các điểm dữ liệu thuộc lớp 1 và càng gần với 0 với các điểm dữ liệu thuộc lớp 0. Như đã đề cập ở trước:

$$p(y = 1|x) = \sigma(W^T x) \tag{2.7}$$

$$p(y = 0|x) = 1 - \sigma(W^T x) \tag{2.8}$$

Đặt giá trị dự đoán của mô hình là $\alpha = \sigma(W^T x)$, theo phân phối Bernoulli ta gộp 2 phương trình 2.7 và 2.8 lại thành:

$$p(y|x) = \alpha^y (1 - \alpha)^{1-y} \tag{2.9}$$

Tìm tham số W

Cho tập huấn luyện $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}))$, với N là số lượng các véc-tơ đầu vào. Để tìm ra được tham số W của hàm dự đoán của "Logistic regression" ở công thức 2.6, ta sẽ dùng phương pháp "maximum likelihood".

Với giả định các mẫu dữ liệu trong tập huấn luyện được phát sinh một cách độc lập, ta có hàm "likelihood" sau:

$$\begin{aligned} L(W) &= p(Y|X) \\ &= \prod_{i=1}^{N} p(y^{(i)}|x^{(i)}) \\ &= \prod_{i=1}^{N} (\alpha^{(i)})^{y^i} (1 - (\alpha^{(i)}))^{1-y^i} \end{aligned} \tag{2.10}$$

Trong đó:

- $X = \{x^{(1)}, \ldots, x^{(N)}\}$ và $Y = \{y^{(1)}, \ldots, y^{(N)}\}$.

Ta tìm W sao cho hàm "likelihood" $L(W)$ đạt cực đại. Cực đại $L(W)$ tương đương với cực tiểu hàm "negative log-likelihood" $-log(W)$.

Như vậy, ta sẽ tìm các tham số W của hàm dự đoán của "Logistic regression" sao cho hàm chi phí sau đạt cực tiểu:

$$\begin{aligned} C(W) &= -log(L(W)) \\ &= -\sum_{i=1}^{N} (y^i log(\alpha^{(i)}) + (1 - y^i) log(1 - \alpha^{(i)})) \end{aligned} \tag{2.11}$$

Để cực tiểu hàm này, ta có thể sử dụng thuật toán "Gradient Descent" (sẽ được trình bày ở các phần sau).

2.3 "Naive bayes"

"Naive bayes" là một mô hình học có giám sát và thường được sử dụng trong bài toán phân lớp. Mô hình này hoạt động bằng cách sử dụng định lý "Bayes" để phân lớp các véc-tơ đầu vào, tuân theo một giả định là các biến đầu vào độc lập với nhau. Trong khóa luận của nhóm chúng em, "Naive

bayes" được sử dụng để ước lượng propensity bằng cách sử dụng ma trận tương tác và một phần nhỏ dữ liệu kiểm tra tuân theo phân phối đều.

2.3.1 Định lý "Bayes"

Định lý "Bayes" là một định lý về xác suất có điều kiện. Nó được sử dụng để tính toán xác suất một biến cố A xảy ra khi biết biến cố B xảy ra, được tính toán thông qua công thức sau:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.12}$$

Trong đó:

- $p(A|B)$ là xác suất biến cố A xảy ra với điều kiện biến cố B xảy ra. Tương tự, $p(B|A)$ là xác suất biến cố B xảy ra với điều kiện biến cố A xảy ra.
- $p(A)$ và $p(B)$ là xác suất xảy ra 2 biến cố A và B độc lập, không bị tác động bởi biến cố khác.

2.3.2 Hàm dự đoán của "Naive bayes"

Với một véc-tơ đầu vào $x \in \mathbb{R}^{D\times 1}$, với D là số lượng đặc trưng của véc-tơ đầu vào, hàm dự đoán $h(x)$ của "Naive Bayes" sẽ trả về một véc-tơ gồm có K phần tử (ứng với K lớp), trong đó phần tử thứ k cho biết xác suất $p(y = k|x)$ với $y \in \{1, 2, \ldots, K\}$ là nhãn lớp của véc-tơ đầu vào x.

Như vậy, véc-tơ đầu vào x sẽ thuộc lớp có xác suất lớn nhất.

$$\begin{aligned} h(x) &= \begin{bmatrix} p(y=1|x) \\ p(y=2|x) \\ \vdots \\ p(y=k|x) \end{bmatrix} \\ &= \begin{bmatrix} \frac{p(x|y=1)p(y=1)}{p(x))} \\ \frac{p(x|y=2)p(y=2)}{p(x)} \\ \vdots \\ \frac{p(x|y=k)p(y=k)}{p(x))} \end{bmatrix} \\ &= \begin{bmatrix} p(x|y=1)p(y=1) \\ p(x|y=2)p(y=2) \\ \vdots \\ p(x|y=k)p(y=k) \end{bmatrix} \end{aligned} \tag{2.13}$$

Trong đó:

- $p(y=k)$ là xác suất thuộc lớp k.
- $p(x|y=k)$ là xác suất véc-tơ đầu vào x thuộc lớp k.

2.3.3 Tìm các xác suất của hàm dự đoán của "Naive Bayes"

Xác suất p(y)

Gọi N là số lượng các mẫu huấn luyện, N_k ($k \in 1, 2, \ldots, K$) là số lượng các véc-tơ đầu vào thuộc lớp k. Xác suất $p(y=k)$ có thể được tính toàn bằng số điểm dữ liệu trong tập huấn luyện thuộc lớp này chia cho tổng số điểm dữ liệu trong tập huấn luyện, được thể hiện cụ thể trong công thức sau:

$$p(y=k) = \frac{N_k}{N} \tag{2.14}$$

Xác suất p(x|y)

Gọi x là một mẫu bất kỳ trong tập huấn luyện, $\{x_1, x_2, \ldots, x_d\}$ các thuộc tính của mẫu x, với d là số lượng các thuộc tính của mẫu x. Giả xử các mẫu trong tập huấn luyện được phát sinh một cách độc lập với nhau. Khi đó, xác suất đầu vào x thuộc lớp k là $p(x|y = k)$ sẽ được tính toán theo công thức:

$$\begin{aligned} p(x|y=k) &= p(x_1, x_2, \ldots, x_d|y=k) \\ &= \prod_{I=1}^{d} p(x_i|y=k) \quad \text{với } i \in (1, \ldots, d) \end{aligned} \tag{2.15}$$

Với $N_{x_i,k}$ là số lần đặc trưng x_i thuộc lớp k. Xác suất thuộc tính x_i thuộc lớp k sẽ được tính toán bằng số lượng thuộc tính của công thức sau:

$$p(x_i|y=k) = \frac{N_{x_i,k}}{N_k} \tag{2.16}$$

2.4 "Gradient descent"

"Gradient descent" là một thuật toán dùng để ước lượng các tham số trong hàm chi phí, nhằm mục đích tìm ra các tham số mà tại đó hàm chi phí đạt giá trị nhỏ nhất (hay nói cách khác là tìm điểm cực tiểu của hàm chi phí). Trong phạm vi khóa luận, thuật toán "Gradient descent" được sử dụng để tối ưu hàm chi phí của 2 mô hình "Matrix factorization" và "Logistic regression".

2.4.1 Ý tưởng chính

Thông thường khi tìm kiếm điểm cực tiểu của một hàm số $f(x)$ nào đó ta sẽ tìm đạo hàm của hàm số đó $f^{'}(x)$ và điểm cực tiểu của hàm số sẽ là điểm có đạo hàm bằng 0. Đường tiếp tuyến của một điểm trong đồ thị của hàm số đó sẽ có hệ số góc bằng với đạo hàm của điểm đó, hệ số góc

của đường tiếp tuyến tại điểm cực tiểu sẽ có giá trị bằng 0. Tuy nhiên, trong các bài toán học máy khi các hàm chi phí có thể rất phức tạp, hay các điểm dữ liệu có số chiều lớn thì việc tìm kiếm điểm cực tiểu bằng cách giải phương trình đạo hàm bằng 0 trở nên bất khả thi. Vì vậy thuật toán "Gradient descent" ra đời như là một phương pháp để giải quyết vấn đề này.

Thuật toán "Gradient descent" được minh họa trong hình 2.3 hoạt động theo ý tưởng như sau:

- Khởi tạo các tham số của hàm chi phí ngẫu nhiên, thông thường khởi tạo bằng 0 (đây có thể xem như là một điểm ngẫu nhiên nằm trên đồ thị hàm số chi phí).

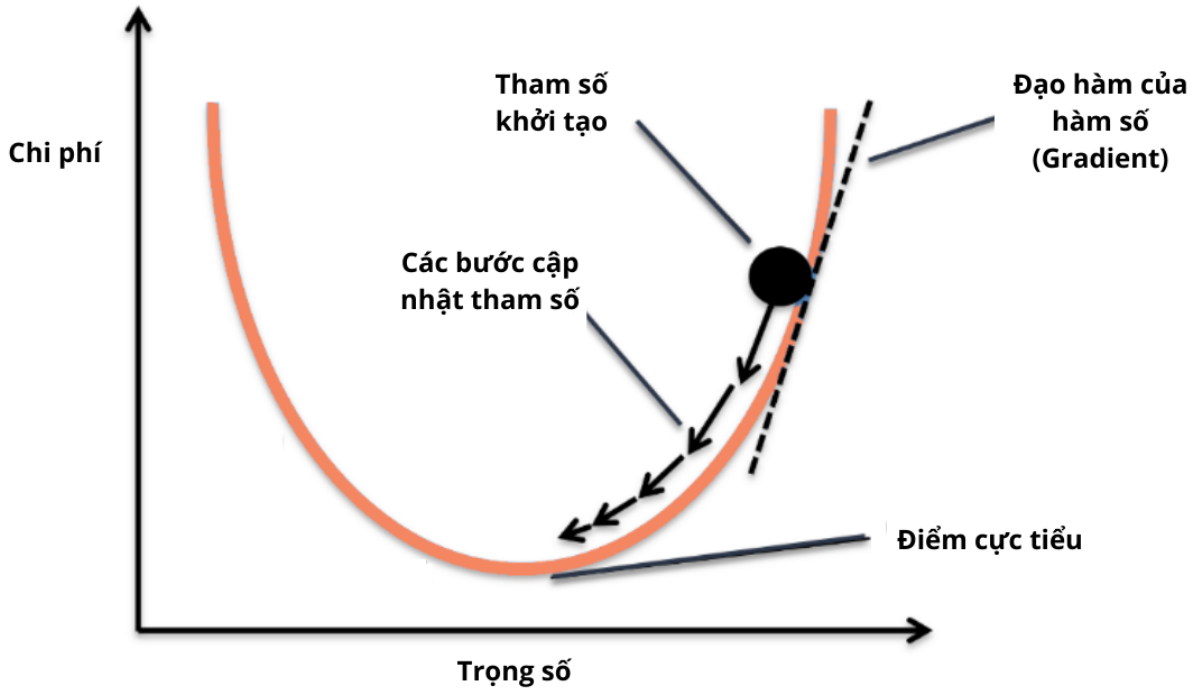
- Sau đó ta thực hiện quá trình cập nhất tham số như sau: tính toán đạo hàm của hàm chi phí tại điểm đó, hệ số góc của tiếp tuyến chính là đạo hàm tại điểm đó, thông qua hệ số góc ta sẽ biết được phương hướng di chuyển. Cập nhật giá trị bộ tham số theo hướng giảm giá trị của hàm chi phí, tức là theo hướng có góc dốc nhất (đây có thể xem như là di chuyển đến điểm hướng xuống dốc).

- Lặp lại quá trình cập nhật tham số cho đến khi thỏa điều kiện dừng. Điều kiện dừng có thể là giới hạn số vòng lặp của việc cập nhật tham số,; có thể khi giá trị của gradient trong hai lần cập nhật liên tiếp cách nhau rất nhỏ; hoặc cũng có thể là khi giá trị của hàm chi phí trong hai lần cập nhật liên tiếp cách nhau rất nhỏ.

Với $W = W_0, W_1, \ldots, W_k$ là véc-tơ chứa các tham số của hàm chi phí, k là số lượng các tham số; $C(W)$ là hàm chi phí của mô hình dựa trên bộ tham số W. Cụ thể, thuật toán "Gradient descent" sẽ được thực hiện bằng cách lặp lại quá trình cập nhật bộ tham số W cho đến khi thỏa điều kiện dừng như sau:

$$W_j = W_j - \alpha \frac{\partial C(W)}{\partial W_j} \tag{2.17}$$

D



Hình 2.3: Minh họa quá trình tìm điểm cực tiểu của hàm chi phi thông qua thuật toán "Gradient descent".

Trong đó:

- α là siêu tham số đại diện cho learning rate của thuật toán, nó quyết định độ lớn của mỗi lần cập nhật. Khi *alpha* nhỏ sẽ dẫn đến mỗi lần cập nhật sẽ có độ lớn nhỏ hơn và khi *alpha* lớn thì mỗi lần cập nhật sẽ có độ lớn lớn hơn. Trong hình 2.3 *alpha* là độ lớn của mỗi mũi tên trong quá trình cập nhật tham số.
- $j \in \{0, 1, \ldots, k\}$ là chỉ số của các tham số trong hàm chi phí.
- $\frac{\partial C(W)}{\partial W_j}$ là đạo hàm riêng của hàm chi phí, nó được sử dụng để xác định hướng cập nhật của tham số. Trong hình 2.3 đạo hàm riêng của hàm chi phí được sử dụng để xác định hướng của mũi tên trong việc cập nhật tham số.

Siêu tham số α là một tham số cực kì quan trọng trong thuật toán "Gradient descent". Vì α đại diện cho learning rate của thuật toán, nên

khi α quá nhỏ thuật toán "Gradient descent" sẽ tốn rất nhiều thời gian trong việc đạt đến điểm cực tiểu của hàm chi phí (hội tụ). Ngược lại, khi α quá lớn mặc dù làm độ lớn của các tham số được cập nhật qua mỗi vòng lặp lớn hơn, tuy nhiên nó có thể dẫn đến việc các tham số cập nhật có thể bị vượt quá điểm cực tiểu, thậm chí có thể sẽ dẫn đến tình trạng thuật toán không thể đạt đến điểm cực tiểu của hàm chi phí (không thể hội tụ).

Chương 3

Phương pháp tìm hiểu

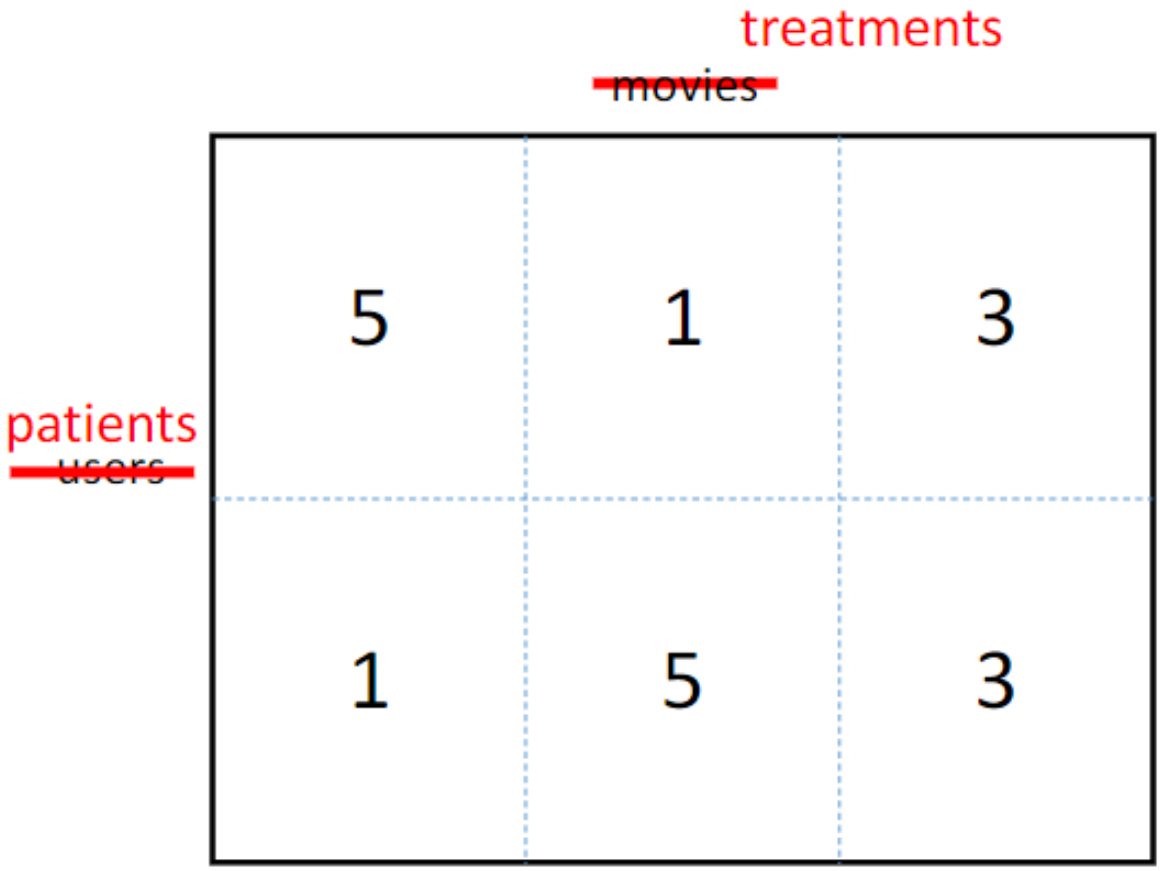
Chương này nhóm chúng em trình bày về những đóng góp của bài báo mà nhóm chúng em tìm hiểu được. Ở đây, nhóm chúng em tập trung vào việc xử lý vấn đề thiên lệch dữ liệu; bằng cách kết nối giữa bài toán xây dựng hệ thống gợi ý và suy luận nhân quả, nhóm em sử dụng phương pháp Inverse propensity scoring (IPS), một phương pháp thường được sử dụng trong suy diễn nhân quả, cho quá trình đánh giá và huấn luyện mô hình gợi ý. Sau đó, nhóm chúng em trình bày về hai phương pháp ước lượng propensity được trình bày ở phần trước là "Naive Bayes" và "Logistic Regression". Tiếp theo, nhóm em trình bày phương pháp "Matrix factorization" dựa trên propensity vừa tìm được. Sau cùng, nhóm em trình bày về các vấn đề của phương pháp IPS và một phương pháp khác có thể khắc phục được các vấn đề của IPS là "Self Normalized Inverse Propensity Scoring" (SNIPS).

3.1 Xem hệ thống gợi ý như một tác động điều trị

Như đã trình bày ở ví dụ về thiên lệch dữ liệu, ta có thể thấy rằng việc người dùng đánh giá hoặc không đánh giá một sản phẩm có thể bị ảnh hưởng bởi rất nhiều yếu tố tiềm ẩn. Giống như việc khi ta muốn xem xét một loại thuốc hay một phương pháp điều trị nhất định có hiệu quả như

thể nào đối với bệnh nhân, bất kể kết quả có tích cực hay tiêu cực thì cũng có thể bị ảnh hưởng bởi rất nhiều yếu tố tiềm ẩn. Cụ thể, giả sử ta tiến hành thử nghiệm phương pháp điều trị đó trên một nhóm bệnh nhân, sau đó ta theo dõi tình trạng bệnh của các bệnh nhân đó, và ta quan sát được sức khỏe của bệnh nhân có chuyển biến tích cực, liệu ta có thể kết luận được rằng phương pháp điều trị đó có thật sự hiệu quả không? Nếu xem xét kỹ lưỡng, có thể rằng phương pháp điều trị của ta có lẽ khá đắt tiền nên chỉ những người có điều kiện kinh tế ổn định mới có xu hướng dễ tiếp xúc với phương pháp điều trị đó. Mà những người như vậy thì sẽ nhiều điều kiện thuận lợi để chăm sóc sức khỏe bản thân hơn, dẫn đến việc tình trạng bệnh của họ có chuyển biến tích cực hơn. Do đó việc kiểm tra được độ hiệu quả một phương pháp điều trị không hề đơn giản, người ta thường sử dụng các mô hình nhân quả. Nếu ta xem mỗi người dùng như một bệnh nhân, mỗi bộ phim ta gợi ý giống như một phương pháp điều trị hoặc một loại thuốc được thể hiện trong hình 3.1, ta sẽ quan tâm đến việc người dùng có phù hợp với bộ phim mà ta gợi ý hay không, phim được gợi ý có khiến người dùng thích thú hay không. Từ đó ta có thể thấy bài toán gợi ý và bài toán được đưa ra khá tương đồng nhau, do đó ta cũng sử dụng ý tưởng giải quyết của các phương pháp nhân quả đối với bài toán xem xét hiệu quả điều trị để gợi ý sản phẩm.

Trong ví dụ về xem xét hiệu quả của một phương pháp điều trị, ta thấy được rằng ta không thể có được kết quả tình trạng bệnh của tất cả các bệnh nhân với tất cả các phương pháp điều trị như hình 3.1 mà chỉ có kết quả của các bệnh nhân tham gia điều trị hoặc tham gia thử nghiệm. Vậy nên, ta cần phải kiểm soát cả những biến làm ảnh hưởng đến khả năng bệnh nhân có thể tiếp cận được với điều trị đó để mô hình trở nên chính xác hơn, ví dụ như những đặc trưng về nhân khẩu học, thu nhập, mức sống, vị trí sống,... Tuy nhiên sẽ có rất nhiều biến ẩn như vậy mà ta khó có thể kiểm soát được. Do đó phương pháp Inverse propensity scoring (IPS) được nghiên cứu với ý tưởng rằng ta không cần phải kiểm soát trực tiếp những biến ẩn, mà chỉ cần kiểm soát được xu hướng nhận được điều

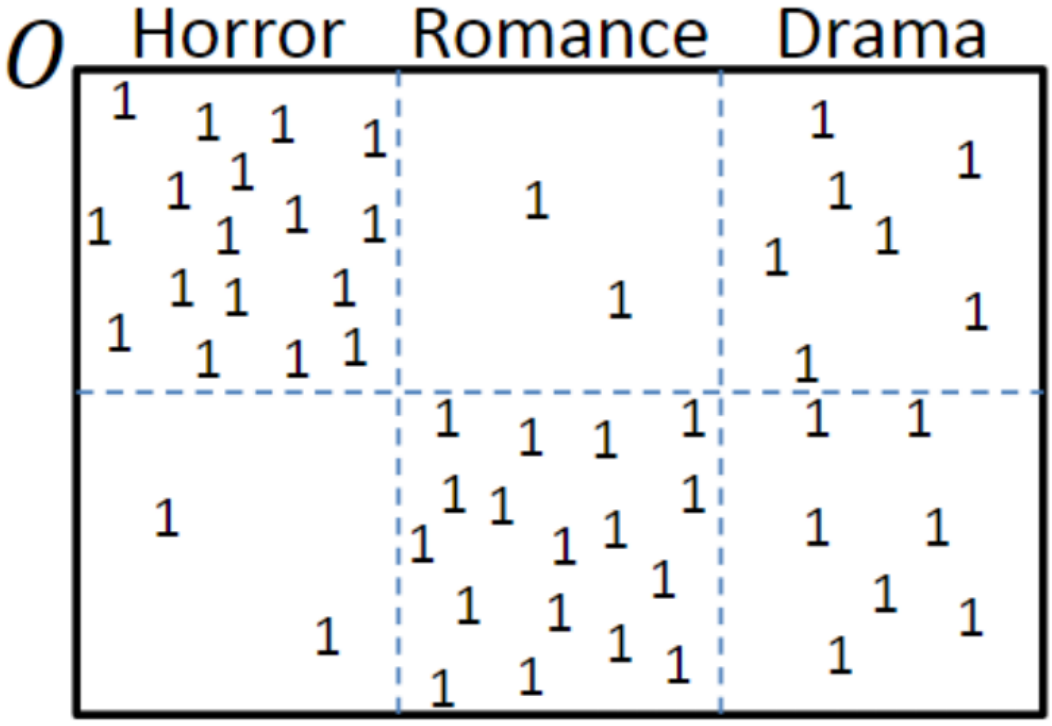


Hình 3.1: Mối liên hệ giữa hệ thống gợi ý và tác động điều trị.

trị của những bệnh nhân. Điều này vẫn đúng với bài toán xây dựng hệ thống gợi ý, vì ta không thể kiếm biết được kết quả đánh giá của toàn bộ người dùng dành cho toàn bộ các bộ phim, cũng không thể kiểm soát những yếu tố tiềm ẩn, ví dụ như bộ phim đó có được gợi ý bởi bạn bè của người dùng hay không... Cho nên việc ta sử dụng một phương pháp phổ biến của suy diễn nhân quả cho bài toán xây dựng hệ thống gợi ý là hoàn toàn có cơ sở.

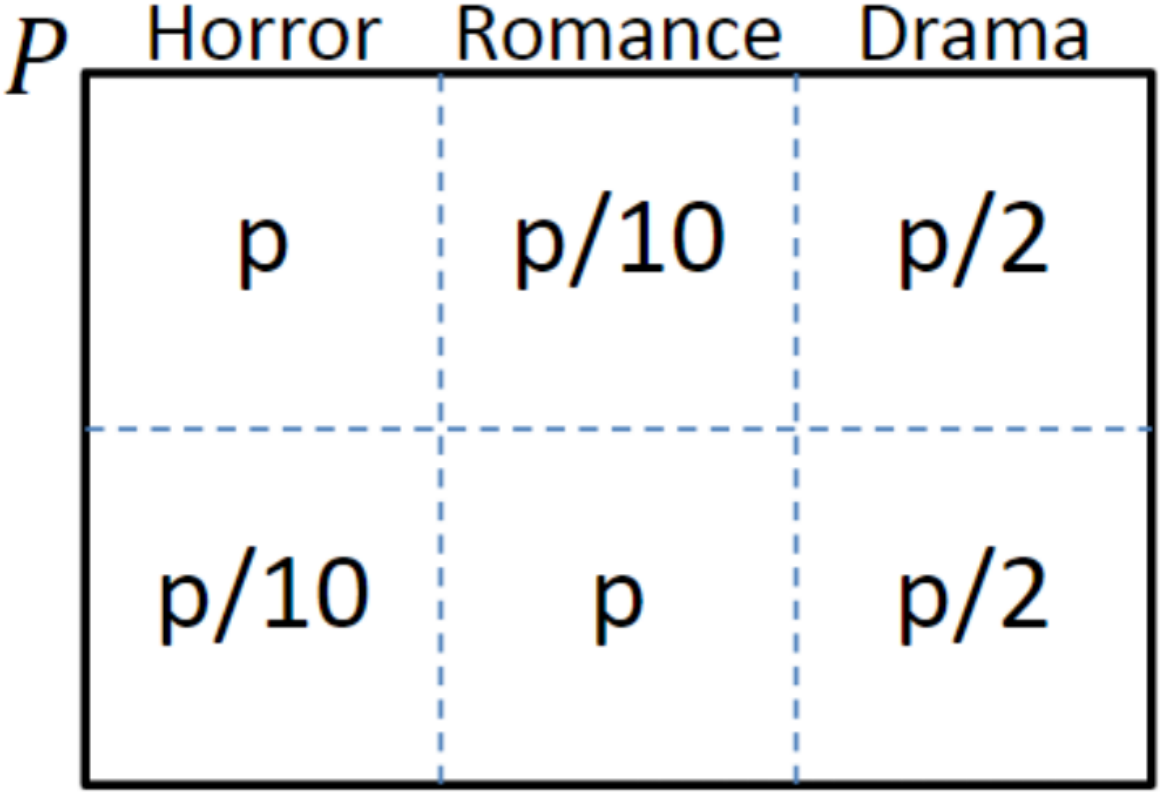
Trong bài toán gợi ý, với giả định rằng người dùng có thể đánh giá hoặc không đánh giá một phim mà họ đã xem, một đánh giá của người dùng dành cho một sản phẩm có thể xuất hiện hoặc không xuất hiện trong tập dữ liệu mà ta qua sát được. Ta có thể biểu diễn điều này thông qua ma trận quan sát O như trong hình 3.2, trong đó các giá trị $O_{u,i} = 1$ tương ứng

với đánh giá cho bộ phim i từ người dùng u được cung cấp tới hệ thống. Ta đặt $P_{u,i}$ là xác suất mà đánh giá $Y_{u,i}$ được quan sát, hay $P_{u,i} = P(O_{u,i} = 1)$. $P_{u,i}$ được gọi là điểm propensity (propensity) của đánh giá $Y_{u,i}$. Từ các đặc trưng mà ta đã quan sát được về các người dùng, các bộ phim và ma trận quan sát O, ta có thể dự đoán được propensity của ma trận đánh giá Y. Phương pháp này sẽ được giới thiệu ở phần tiếp theo của khóa luận này.



Hình 3.2: Hình ảnh minh họa ma trận quan sát O (hình ảnh được lấy từ bài báo của tác giả Tobias Schnabel [9]).

Hình ảnh 3.3 và 3.2 minh họa về việc biểu diễn ma trận quan sát O và ma trận propensity P. Trong đó, ma trận quan sát O các vị trí có giá trị là 1 đại diện cho đánh giá của người dùng với sản phẩm xuất hiện trong hệ thống, ma trận propensity P chứa propensity của các đánh giá tương ứng với khả năng ta quan sát được các đánh giá, những phim có số lượt đánh giá nhiều sẽ có propensity cao.



Hình 3.3: Hình ảnh minh họa ma trận quan sát P (hình ảnh được lấy từ bài báo của tác giả Tobias Schnabel [9]).

3.2 "Inverse propensity scoring" (IPS)

3.2.1 Các hàm độ lỗi truyền thống

Nhắc lại về phương pháp tính độ lỗi mà ta thường sử dụng, ta ký hiệu ma trận đánh giá mà ta quan sát được là Y, ma trận đánh giá mà ta dự đoán là $\hat{Y}$. Với Y là ma trận thưa chứa đánh giá của các người dùng cho các sản phẩm trong hệ thống, do người dùng không thể nào xem được tất cả sản phẩm, vì vậy đây là một ma trận thưa với rất nhiều đánh giá bị thiếu. Ma trận đánh giá mà ta dự đoán $\hat{Y}$ là ma trận đã được được điền đầy đủ các đánh giá bị thiếu từ mà trận Y; cụ thể, đây là ma trận chứa đánh giá của tất cả người dùng cho tất cả sản phẩm. Mục tiêu của phương pháp tính độ lỗi là xem xét liệu ma trận dự đoán $\hat{Y}$ có phải là ma trận $\hat{Y}$

khi đã được điền đầy đủ các giá trị thiếu hay không. Vì vậy hàm tính độ lỗi của ta sẽ tính toán độ lỗi dựa trên sự khác biệt về điểm đánh giá trên ma trận $\hat{Y}$ khi so sánh với các đánh giá trên ma trận Y. Thông thường, hàm tính độ lỗi sẽ được biểu diễn như sau:

$$R(\hat{Y}) = \frac{1}{U \cdot I} \sum_{u=1}^{U} \sum_{i=1}^{I} \delta_{u,i}(Y, \hat{Y}) \tag{3.1}$$

Trong đó, δ là một hàm tính độ lỗi bất kì. Nhưng vì ta chỉ có thể quan sát được một phần của toàn bộ đánh giá, do đó ta chỉ tính trung bình độ lỗi của các đánh giá quan sát được, ta tạm gọi hàm lỗi này là hàm lỗi ngây thơ (Naive), hàm lỗi này có công thức như sau:

$$R_{Naive}(\hat{Y}) = \frac{1}{|\{(u,i) : O_{u,i} = 1\}|} \sum_{(u,i):O_{u,i}=1}^{I} \delta_{u,i}(Y, \hat{Y}) \tag{3.2}$$

Sự "ngây thơ" của hàm lỗi này đã dẫn đến việc đánh giá mô hình bị sai ở ví dụ về thiên lệch lựa chọn trong phần 1.2. Do mẫu dữ liệu mà ta quan sát được không được phát sinh ngẫu nhiên theo phân phối đều từ dữ liệu thực tế mà bị thiên lệch, bắt nguồn từ việc người dùng tự lựa chọn các sản phẩm để đánh giá. Đó là lý do tại sao hàm lỗi ngây thơ lại có giá trị khác biệt so với độ lỗi thực tế, người ta gọi đây là một hàm lỗi bị lệch (bias), hay nói cách khác, kỳ vọng của hàm lỗi này khác với độ lỗi thực tế.

$$E_O[R_{Naive}(\hat{Y})] \neq R(\hat{Y})$$

3.2.2 Hàm tính độ lỗi IPS

Hiểu được vấn đề của hàm lỗi ngây thơ, nhóm tác giả đã đưa ra một hàm lỗi thay thế giúp giải quyết được vấn đề dữ liệu bị lệch. Phương pháp dựa trên một phương pháp thường được sử dụng trong các mô hình nhân quả, gọi là phương pháp Inverse propensity scoring (IPS).

Áp dụng IPS trong nghiên cứu của Imbens và Rubin [2] để sử dụng cho hàm lỗi của hệ thống gợi ý, ta định nghĩa công thức tính độ lỗi IPS như sau:

$$R_{IPS}(\hat{Y}|P) = \frac{1}{U \cdot I} \sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(Y, \hat{Y})}{P_{u,i}} \tag{3.3}$$

Về cơ bản, propensity này luôn luôn lớn hơn 0 ở mọi cặp người dùng - sản phẩm để chắc chắn mỗi phần tử trong ma trận đánh giá Y đều có thể được quan sát; và tổng nghịch đảo các propensity của các đánh giá mà ta quan sát được sẽ bằng với số lượng đánh giá của toàn bộ người dùng dành cho toàn bộ sản phẩm, hay nói cách khác:

$$\mathbb{E}_O \left[\sum_{(u,i):O_{u,i}=1} \frac{1}{P_{u,i}} \right] = U \cdot I \tag{3.4}$$

Bổ đề 3.2.1 *(Sự không thiên lệch của IPS) Với mọi người dùng u và sản phẩm i, nếu propensity* $P_{u,i} \in (0, 1)$ *và* $P_{u,i} > 0$, *thì* $R_{IPS}(\hat{Y}|P)$ *là một độ đo không thiên lệch của* $R(\hat{Y})$, *có nghĩa là kỳ vọng của* $R_{IPS}(\hat{Y}|P)$ *bằng với* $R(\hat{Y})$.

Chứng minh:

$$\begin{aligned} E_O \left[R_{IPS}(\hat{Y}|P) \right] &= \frac{1}{U \cdot I} \cdot \sum_{u=1}^{U} \sum_{i=1}^{I} \mathbb{E}_O \left[\frac{\delta_{u,i}(Y, \hat{Y}}{P_{u,i}} O_{u,i} \right] \\ &= \frac{1}{U \cdot I} \cdot \sum_{u=1}^{U} \sum_{i=1}^{I} \delta_{u,i} \\ &= R(\hat{Y}) \end{aligned} \tag{3.5}$$

Trong phạm vi của bài báo mà nhóm em tìm hiểu, tác giả tiến hành hai loại nghiên cứu là nghiên cứu quan sát và nghiên cứu thực nghiệm:

- Nghiên cứu thực nghiệm: trong nghiên cứu này, ta có thể điều khiển hệ thống gợi ý của ta bằng cách quyết định những sản phẩm nào sẽ

được hiển thị đến người dùng, từ đó ta có thể biết được propensity của nó.

- Nghiên cứu quan sát: trong nghiên cứu này, ta không biết được propensity từ trước mà cần phải tiến hành ước lượng nó thông qua các thuộc tính của người dùng và sản phẩm; hoặc có thể thông qua dữ liệu đánh giá. Phương pháp ước lượng này sẽ được trình bài cụ thể trong phần tiếp theo thông qua các một trong hai mô hình "Naive Bayes" và "Logistic Regression".

3.3 Ước lượng propensity

Trước tiên, để hiểu được các phương pháp ước lượng propensity của một đánh giá hay nói các khác là xác suất mà một đánh giá được quan sát, ta cần hiểu được các loại mất mát dữ liệu. Đầu tiêu là MAR (Mising At Random), có nghĩa là sự mất mát dữ liệu này là ngẫu nhiên. Kiểu mất mát này là kiểu ta thường giả định trong học máy. Ở kiểu mất mát này, ta có thể ước lượng được giá trị bị thiếu thông qua các giá trị quan sát được. Ví dụ như trong nghiên cứu về thông tin nhân khẩu học, nếu ta giả định rằng giá trị thu nhập bị thiếu là ngẫu nhiên thì ta có thể ước lượng thu nhập bị thiếu dựa vào các thông tin ta quan sát được như độ tuổi, nghề nghiệp, nơi sống,..Tuy nhiên, một kiểu mất mát dữ liệu khác là MNAR (Missing Not At Random), ở kiểu mất mát này ta sẽ khó ước lượng được thu nhập bị thiếu vì các giá trị bị thiếu thường có một nguyên nhân nào đó, và nó mang một ý nghĩa nhất định.

Trong trường hợp trên, có thể những người dùng có thu nhập cao có xu hướng hạn chế công khai thu nhập của bản thân, do đó những thu nhập bị thiếu có thể cao hơn phần thu nhập ta quan sát được rất nhiều do đó không thể ước lượng bằng các mô hình học máy thông thường. Còn một kiểu mất mát khác là MCAR (Missing Completely At Random), có nghĩa là sự mất mát dữ liệu là hoàn toàn ngẫu nhiên, các mẫu mà ta quan sát

được có thể xem như những mẫu đại diện, do đó ta có thể xóa những mẫu có dữ liệu bị thiếu. Ta có thể tìm hiểu kĩ hơn về các vấn đề mất mát dữ liệu trong nghiên cứu của [5].

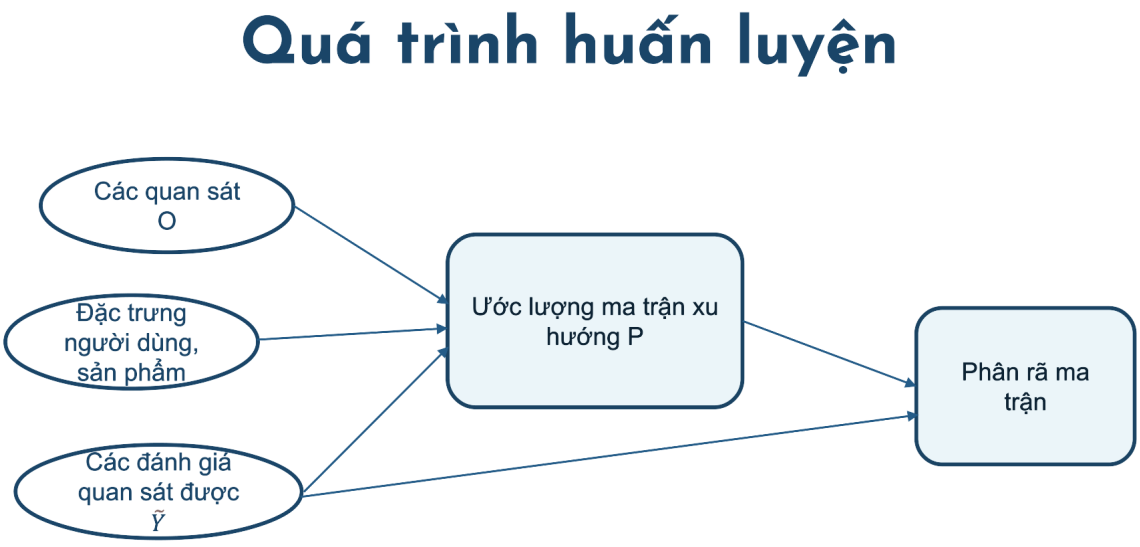
Xét nghiên cứu quan sát trên tập dữ liệu ML100K được cung cấp bởi Grouplens, ta dễ thấy các đánh giá bị thiếu không phải do ngẫu nhiên (MNAR) mà do thiên lệch lựa chọn của người dùng và của hệ thống gợi ý đã sử dụng. Do đó ta không biết được propensity của các đánh giá mà cần ước lượng propensity $P_{u,i}$ của mỗi đánh giá của người dùng u dành cho sản phẩm i sẽ được quan sát. Nói chung, xác suất một đánh giá ta có thể quan sát được có thể phụ thuộc vào các đặc trưng có thể quan sát được X (ví dụ như đặc trưng của người dùng, đặc trưng của sản phẩm mà ta thu thập được), các đặc trưng không thể quan sát được X^{hid} (ví dụ như sản phẩm đó có được giới thiệu bởi bạn bè của người dùng hay không), và đánh giá Y:

$$P_{u,i} = P(O_{u,i} = 1|X, X^{hid}, Y) \tag{3.6}$$

Do đó khi các đặc trưng có thể quan sát được đã được sử dụng để tính toán, ta có cơ sở để giả định rằng $O_{u,i}$ độc lập với ma trận dự đoán mới $\hat{Y}$ nên nó độc lập với $\delta_{u,i}(Y, \hat{Y})$.

Tiếp theo, ta làm rõ ý nghĩa của việc ước lượng ma trận propensity và hai phương pháp ước lượng ma trận propensity.

Hình 3.4 cho thấy rằng ta sẽ sử dụng các thông tin về những đặc trưng có thể quan sát được X, ma trận quan sát O và ma trận đánh giá Y để ước lượng ma trận propensity. Cụ thể, phương pháp "Naive Bayes" sử dụng ma trận đánh giá Y và ma trận quán sát O, trong khi phương pháp "Logistic Regression" sử dụng các đặc trưng có thể quan sát X và ma trận quan sát O. Sau khi ước lượng ma trận propensity, ta sẽ sử dụng ma trận này kết hợp với ma trận đánh giá Y để sử dụng cho phương pháp MF-IPS sẽ được trình bày cụ thể trong 3.4.



Hình 3.4: Tóm tắt quá trình học.

3.3.1 Ước lượng ma trận propensity thông qua "Naive Bayes"

Để thực hiện được phương pháp "Naive Bayes" cho ước lượng ma trận propensity, ta cần phải giả định rằng sự phụ thuộc giữa các biến X, X_{hid} và các đánh giá khác là không đáng kể. Do đó công thức 3.6 được đơn giản thành $P(O_{u,i}|Y_{u,i})$ tương tự như nghiên cứu của nhóm tác giả Marlin và Zemel [3]. Ta có thể xem $Y_{u,i}$ như là những mẫu đánh giá mà ta quan sát được, khi đó ta chỉ cần ước lượng các propensity cho những đánh giá quan sát được để tính toán IPS và SNIPS [10]. Ta sử dụng mô hình "Naive Bayes" để ước lượng propensity này như sau:

$$P(O_{u,i} = 1|Y_{u,i} = r) = \frac{P(Y = r|O = 1)P(O = 1)}{P(Y = r)} \tag{3.7}$$

Trong đó việc ước lượng maximum $P(Y = r|O = 1)$ và $P(O = 1)$ có thể được tính toán thông qua đếm số đánh giá quan sát được trong dữ liệu MNAR. Tuy nhiên, khi muốn ước lượng $P(Y = r) = P(Y = r|O = 1) + P(Y = r|O = 0)$, ta cần phải có một mẫu nhỏ MCAR. Phương pháp

cụ thể để tìm tập nhỏ MNAR sẽ được trình bày ở phần thực nghiệm.

3.3.2 Ước lượng ma trận propensity thông qua "Logistic Regression"

Một hướng tiếp cận khác để giải quyết việc ước lượng ma trận propensity là sử dụng hồi quy Logistic. Phương pháp này có ưu điểm là không yêu cầu một tập mẫu nhỏ MCAR. Cũng dựa trên công thức 3.6, nhưng mục tiêu của ta là tìm một bộ tham số ϕ sao cho ma trận quan sát O có thể độc lập với ma trận ma trận đặc trưng không quan sát được X^{hid} và Y. Nói cách khác, $P(O_{u,i} = 1|X, X^{hid}, Y) = P(O_{u,i}|X, \phi)$. Ở phương pháp này, ta giả định rằng tồn tại một bộ tham số $\phi = (\omega, \beta, \gamma)$ sao cho:

$$P_{u,i} = \sigma(\omega^T X_{u,i} + \beta_i + \gamma_u) \tag{3.8}$$

Trong đó, $X_{u,i}$ là vector được vector hóa từ những thông tin quan sát được về các cặp người dùng, sản phẩm (ví dụ như thông tin nhân khẩu học của người dùng, bộ phim có được quảng cáo hay không), $\sigma(\cdot)$ là hàm sigmoid, β_i, γ_u lần lượt là offset của người dùng và sản phẩm.

3.4 "Matrix factorization" kết hợp với IPS

Ở phần trước, ta đã tìm hiểu về phương pháp IPS ngoài ra ta còn tìm hiểu về cách ước lượng propensity của các đánh giá hay còn gọi là xác suất để các đánh giá xuất hiện trong tập dữ liệu mà ta quan sát được. Trong phần này, dựa vào những kiến thức đã trình bày trước đó, nhóm em sẽ sử dụng thay thế hàm lỗi của mô hình "Matrix factorization" bằng độ đo IPS dựa trên công thức 3.3.

Phát biểu bài toán: Với ma trận quan sát O, ma trận đánh giá Y, ma trận propensity P, không gian giả thuyết $\mathcal{H}$ của các dự đoán $\hat{Y}$ và hàm lỗi

$\delta_{u,i}(Y, \hat{Y})$. Mục tiêu của ta là tìm $\hat{Y}$ sao cho:

$$\hat{Y} = \underset{\hat{Y} \in \mathcal{H}}{argmin} \left\{ \hat{R}_{IPS}(\hat{Y}|P \right\} \tag{3.9}$$

Tiếp theo, để giải quyết bài toán dự đoán đánh giá $\hat{Y}$, ta sử dụng phương pháp "Matrix factorization" như đã trình bày ở chương 2.1. Giả sử ta xem mô hình "Matrix factorization" với nhiệm vụ dự đoán mỗi đánh giá $\hat{Y}_{u,i} = \upsilon_u^T \omega_i + a_u + b_i + c$ với a_u là offset tương ứng của từng người dùng, b_i là offset của sản phẩm, c là offset toàn cục của không gian giả thuyết $\mathcal{H}$. Từ đó ta có thể xem mục tiêu huấn luyện của ta là:

$$\underset{V,W,A}{argmin} \left[\sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(Y, V^T W + A)}{P_{u,i}} + \lambda(||V||_F^2 + ||W||_F^2) \right] \tag{3.10}$$

Ta có thể thấy rằng, các phương pháp "Matrix factorization" truyền thống là một trường hợp đặc biệt của công thức 3.10, với tất cả propensity $P_{u,i}$ đều bằng nhau. Hay nói cách khác, khả năng xuất hiện của các đánh giá là như nhau, điều này xuất phát từ một giả định không chính xác của người dùng là dữ liệu đánh giá mà ta thu thập được là MCAR. Đây chính là điểm đặc biệt của mô hình mà nhóm em tìm hiểu. Mô hình này có vẻ không có quá nhiều khác biệt so với các mô hình thông thường được xây dựng trước đó, chỉ khác nhau một phần nhỏ ở hàm mục tiêu. Tuy nhiên, thách thức của việc tìm được ma trận propensity P là khá lớn, và khó để tìm chính xác được một ma trận propensity P để ta có thể thấy được toàn bộ dữ liệu thật sự dựa vào dữ liệu quan sát được.

3.5 "Self normalized inverse propensity scoring" (SNIPS)

Quay lại với nghiên cứu quan sát, khi mà ta chưa biết được giá trị của các propensity mà cần phải ước lượng chúng, việc ước lượng chính xác các propensity để tập dữ liệu ta quan sát được có thể đại diện cho toàn bộ tập tất cả dữ liệu gần như là một nhiệm vụ khó khăn. Tuy nhiên mục tiêu của ta không nhất thiết phải dự ước lượng một propensity hoàn hảo như vậy mà chỉ cần một propensity có tốt hơn so với việc xem tất cả các propensity là bằng nhau, tương tự như cách ta giả định rằng dữ liệu ta đang quan sát được là MCAR. Do đó, ma trận propensity mà ta ước lượng có thể có khác biệt lớn so với ma trận propensity hoàn hảo. Điều này dẫn đến việc ta cần xem xét một vài tính chất của ma trận propensity ước lượng được gây ảnh hưởng đến mô hình "Matrix factorization".

Bổ đề 3.5.1 *Độ lệch của độ đo IPS với những propensity không chính xác: Đặt P là xác suất biên của việc quan sát được một đánh giá trong ma trận đánh giá Y, và đặt $\hat{P}$ là propensity được ước tính sao cho $\hat{P}_{u,i} > 0$ với mọi u,i. Độ lệch của độ đo IPS ở công thức 3.3 được tính theo công thức:*

$$bias\left(\hat{R}_{IPS}(\hat{Y}|\hat{P})\right) = \sum_{u,i} \frac{\delta(Y,\hat{Y})}{U \cdot I}\left[1 - \frac{P_{u,i}}{\hat{P}_{u,i}}\right] \tag{3.11}$$

Ngoài vấn đề về độ lệch, ta còn cần phải quan tâm đến độ lỗi tổng quát của quá trình học, độ lỗi này bị ảnh hưởng như thế nào do tác động của propensity được ta ước lượng thông qua định lý được phát biểu trong [9], như sau:

Định lý 3.5.2 (Giới hạn của độ lỗi tổng quát với phương pháp IPS) *Với mọi không gian giả thuyết hữu hạn của các dự đoán $\mathcal{H} = \{\hat{Y}_1, ..., \hat{Y}_{\mathcal{H}}\}$ và độ lỗi $0 \leqslant \delta_{u,i}(Y,\hat{Y}) \leqslant \Delta$, sử dụng độ đo IPS với ma trận propensity được ước lượng $\hat{P}(\hat{P}_{u,i} > 0)$ và cho ma trận quan sát O được huấn luyện*

từ Y với ma trận propensity P độc lập theo phân phối Bernoulli, được giới hạn bởi:

$$R(\hat{Y}) \leqslant \hat{R}_{IPS}(\hat{Y}|P) + \sum_{u,i} \frac{\delta(Y,\hat{Y})}{U \cdot I} \left| 1 - \frac{P_{u,i}}{\hat{P}_{u,i}} \right| + \frac{\Delta}{U \cdot I} \sqrt{\frac{log(2|\mathcal{H}|/\eta}{2})} \sqrt{\sum_{u,i} \frac{1}{P_{u,i}^2}} \tag{3.12}$$

Trong đó:

- $\hat{R}_{IPS}(\hat{Y}|P)$ là độ lỗi IPS.
- $\sum_{u,i} \frac{\delta(Y,\hat{Y})}{U \cdot I} \left| 1 - \frac{P_{u,i}}{\hat{P}_{u,i}} \right|$ là độ lệch của độ lỗi IPS với ma trận propensity được ước lượng không hoàn hảo.
- $\frac{\Delta}{U \cdot I} \sqrt{\frac{log(2|\mathcal{H}|/\eta}{2})} \sqrt{\sum_{u,i} \frac{1}{P_{u,i}^2}}$ là phương sai của độ lỗi IPS với ma trận propensity được ước lượng không hoàn hảo.

Định lý trên cho thấy được sự đánh đổi giữa độ lệch và phương sai khác với các phương pháp học thông thường. Ở đây, ta có thể thấy rằng việc đánh giá cao những propensity nhỏ có thể có lợi nếu như việc giảm phương sai lớn hơn việc làm tăng độ lệch. Quay lại với công thức 3.3, nếu ta thay thế propensity thực tế $P_{u,i}$ bằng propensity được ước lượng $\hat{P}_{u,i}$, ta được $\frac{1}{U \cdot I} \sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(Y,\hat{Y}}{\hat{P}_{u,i}}$. Ở đây ta có thể thấy hai vấn đề có thể xảy ra với hàm lỗi IPS, đó là:

- Ta đã biết rằng ta cố gắng ước lượng một ma trận propensity sao cho nó tốt hơn các propensity với phân phối đều, và có thể có sự khác nhau rất nhiều giữa các giá trị trong ma trận propensity. Nếu propensity của một đánh giá nào đó nhỏ hơn propensity của một đánh giá khác hàng trăm lần, thì độ lỗi của nó sẽ được khuếch đại lên gấp hàng trăm lần so với đánh giá khác.

- Ngoài ra, việc ta chỉ có thể ước lượng một ma trận propensity không hoàn hảo có thể khiến giá trị của propensity còn có nhiều khác biệt hơn.

Chính vì hai vấn đề trên, ta có thể thấy rằng độ lỗi IPS có thể biến thiên rất nhiều qua các lần huấn luyện, hay nói cách khác là độ lỗi IPS có phương sai lớn. Điều này khiến mô hình của chúng ta không được ổn định, đây là thứ ta phải đánh đổi cho một độ đo không thiên lệch.

Để giải quyết vấn đề phương sai lớn của độ đo IPS, người ta đã áp dụng một kĩ thuật đó là sử dụng biến kiểm soát. Biến kiểm soát là một biến ngẫu nhiên mà ta biết được kỳ vọng của nó - là một công cụ được sử dụng để giảm phương sai của xấp xỉ Monte Carlo [7]. Đặt $V(X)$ là một biến kiểm soát với kỳ vọng được biết trước $\mathbb{E}_X[V(X)] = \upsilon \neq 0$ và đặt $\mathbb{E}_X[W(X)]$ là kỳ vọng mà ta muốn ước lượng dựa trên các mẫu độc lập của X. Ta có $\mathbb{E}_X[W(X)] = \frac{\mathbb{E}[W(X)]}{\mathbb{E}[V(X)]}\upsilon$. Từ đó ta có độ đo sau:

$$\hat{W}^{SN} = \frac{\sum_{i=1}^{n} W(X_i)}{\sum_{i=1}^{n} V(X_i)}\upsilon \tag{3.13}$$

Độ đo trên được gọi là độ đo tự chuẩn hóa (Self-Normalized) trong các tài liệu về đánh lại trọng số của mẫu [11]. Độ đo này được chứng minh rằng có phương sai giảm hơn đáng kể khi mà biến $W(X)$ và biến $V(X)$ có tương quan với nhau.

Dựa vào độ đo tự chuẩn hóa trên, nếu ta sử dụng $\sum_{(u,i):O_{u,i}=1} \frac{1}{P_{u,i}}$ như một biến kiểm soát cho độ đo IPS, ta biết được giá trị kỳ vọng của biến kiểm soát này theo công thức 3.4 và ta dễ thấy biến ngẫu nhiên này có tương quan với độ đo IPS. Ta được độ đo SNIPS (Self-Normalized IPS) như sau:

$$\hat{R}_{SNIPS}(\hat{Y}|P) = \frac{\frac{1}{U \cdot I} \sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(Y,\hat{Y}}{P_{u,i}})}{\sum_{(u,i):O_{u,i}=1} \frac{1}{P_{u,i}}} U \cdot I \tag{3.14}$$

$$= \frac{\sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(Y,\hat{Y}}{P_{u,i}})}{\sum_{(u,i):O_{u,i}=1} \frac{1}{P_{u,i}}}$$

Độ do SNIPS thường có phương sai nhỏ hơn IPS nhưng phải đánh đổi một chút độ lệch.

Chương 4

Các kết quả thí nghiệm

4.1 Các thiết lập thí nghiệm

4.1.1 Các tập dữ liệu

Nhóm chúng em sẽ tiến hành các thí nghiệm về hiệu năng của mô hình "Matrix factorization" truyền thống (MF) so với "Matrix factorization" sử dụng độ đo IPS (MF-IPS). Các thí nghiệm về hiệu năng này sẽ được đánh giá trên 2 tập dữ liệu Coat Shopping và Yahoo!R3, 2 tập dữ liệu này đều có tập training bị thiên lệch dữ liệu do người dùng tự lựa chọn chọn sản phẩm để đánh giá; và có một tập test không bị thiên lệch do người dùng đánh giá một tập các sản phẩm ngẫu nhiên. Thông tin cụ thể của từng tập dữ liệu được mô tả như sau:

- Tập dữ liệu Coat Shopping: tập dữ liệu này chứa đánh giá của các người dùng cho các áo khoác, được thu thập bằng cách mô phỏng dữ liệu bị thiên lệch của những người dùng mua áo khoác trong cửa hàng trực tuyến. Những người dùng được yêu cầu phải đánh giá 24 chiếc áo khoác họ tự chọn và 16 chiếc áo khoác được chọn ngẫu nhiên theo phân phối đều dựa trên thang điểm từ 1 đến 5. Tập dữ liệu chứa đánh giá từ 290 người dùng cho 300 sản phẩm. Các điểm đánh giá do người dùng tự lựa chọn sẽ được sử dụng làm tập huấn luyện, đồng

thời, các điểm đánh giá trên tập các sản phẩm ngẫu nhiên sẽ được dùng để kiểm tra.

- Tập dữ liệu Yahoo!R3: tập dữ liệu này chứa đánh giá của các người dùng cho các bài hát. Tập dữ liệu huấn luyện bị thiên lệch cung cấp hơn 300 nghìn đánh giá cho các bài hát, các bài hát này được tự lựa chọn bởi 15400 người dùng. Tập dữ liệu kiểm tra chứa đánh giá của 5400 người dùng cho 10 bài hát được chọn ngẫu nhiên theo phân phối đều.

Ngoài ra, nhằm đánh giá ảnh hưởng của việc dữ liệu bị thiên lệch tới việc học của MF và MF-IPS nhóm chúng em còn tiến hành thí nghiệm trên tập dữ liệu MovieLens 100k. Tập dữ liệu này chứa đánh giá của các người dùng cho các bộ phim. Tập dữ liệu MovieLens 100k sẽ chứa 100 000 đánh giá (từ 1 đến 5) của 943 người dùng trên 1682 bộ phim, trong đó mỗi người dùng sẽ đánh giá ít nhất 20 bộ phim. Các bộ phim được đánh giá trong tập dữ liệu này đều do người dùng tự lựa chọn, vì vậy đây là một tập dữ liệu bị thiên lệch. Do đó để thí nghiệm về ảnh hưởng của dữ liệu bị thiên lệch nhóm chúng em sẽ tiến hành tạo một bộ dữ liệu MovieLens 100k giả lập dựa trên bộ dữ liệu gốc, cách tạo bộ dữ liệu giả lập này sẽ được nhóm chúng em giới thiệu cụ thể trong phần 4.3.2.

4.1.2 Các thiết lập về huấn luyện và kiểm tra

Trong tất cả các thí nghiệm, nhóm chúng em sẽ tiến hành lựa chọn các siêu tham số regularization λ và tham số đặc trưng tiềm ẩn d cho mô hình "Matrix factorization" thông qua phương pháp "k-fold Cross-Validation" với $k = 4$. Cụ thể, phương pháp "k-fold Cross-Validation" sẽ tách tập dữ liệu quan sát được MNAR thành 4 phần bằng nhau, trong đó 3 phần sẽ được sử dụng như là tập training và 1 phần còn lại sẽ được sử dụng như là tập validation để đánh giá mô hình huấn luyện được. Do tập dữ liệu được chia thành 4 phần vì vậy các propensity cũng cần được điều chỉnh

cho phù hợp; cụ thể ở tập training, propensity sẽ được nhân với $\frac{3}{4}$ và ở tập validation propensity sẽ được nhân với $\frac{1}{4}$. Các tham số cho độ lỗi nhỏ nhất khi đánh giá trên tập validation sẽ được sử dụng để huấn luyện lại trên toàn bộ tập dữ liệu quan sát được.

Để đánh giá hiệu năng của mô hình, nhóm chúng em sẽ sử dụng độ đo Mean square error (MSE) để đánh giá độ lỗi của giá trị dự đoán của mô hình so với tập test. Độ lỗi trên 2 tập dữ liệu Coat và Yahoo sẽ được tính toán thông qua tập test được cung cấp sẵn. Còn đối tập dữ liệu Movielens 100k, độ lỗi sẽ được tính toán dựa trên dữ liệu giả lập.

4.2 Kết quả cài đặt của nhóm chúng em so với kết quả cài đặt của bài báo

Đầu tiên, nhóm em tiến hành thử nghiệm để xem hiệu quả của phương pháp học MF-IPS so với phương pháp MF thông thường trên hai bộ dữ liệu thế thới thực là Coat và Yahoo!R3. Trên bộ dữ liệu Coat, nhóm em sử dụng phương pháp "Logistic Regression" để ước lượng ma trận propensity như đã trình bày ở phần 3.3.2, trên bộ dữ Yahoo!R3, nhóm em sử dụng phương pháp "Naive Bayes" để ước lượng ma trận propensity như đã trình bày ở phần 3.3.1. Kết quả được hiển thị trong bảng 4.1

	Yahoo		Coat	
	MAE	MSE	MAE	MSE
MF-IPS	**0.796**	**0.976**	**0.904**	**1.197**
MF	1.183	1.899	0.916	1.205

Bảng 4.1: Độ lỗi trên tập test của hai mô hình MF và MF-IPS khi học với hai bộ dữ liệu Coat và Yahoo!R3.

So với kết quả bài báo được thể hiện trong hình 4.1. Thử nghiệm của nhóm em và kết quả của bài báo đều có cùng kết luận là phương pháp MF-IPS sẽ có kết quả trên tập test tốt hơn so với phương pháp MF thông thường, ngoài ra phương pháp MF-IPS sử dụng trên bộ dữ liệu Yahoo!R3

	YAHOO		COAT	
	MAE	MSE	MAE	MSE
MF-IPS	**0.810**	**0.989**	**0.860**	**1.093**
MF-Naive	1.154	1.891	0.920	1.202

Hình 4.1: Kết quả thí nghiệm của bài báo trên tập test với bộ dữ liệu Yahoo và Coat

cải thiện được độ lỗi nhiều hơn so với phương pháp MF-IPS được sử dụng trên bộ dữ liệu Coat. Điều này có thể do bộ dữ liệu Yahoo!R3 bị lệch nhiều hơn bộ dữ liệu Coat hoặc cũng có thể do phương pháp ước lượng ma trận propensity bằng "Naive Bayes" sử dụng một phần thông tin được tiết lộ từ tập test nên kết quả thu được tốt hơn so với phương pháp "Logistic Regression".

Tiếp theo, nhóm em sẽ tiến hành các thí nghiệm để làm rõ những vấn đề trên.

4.3 Ảnh hưởng của việc dữ liệu bị thiên lệch tới việc học của MF và MF-IPS

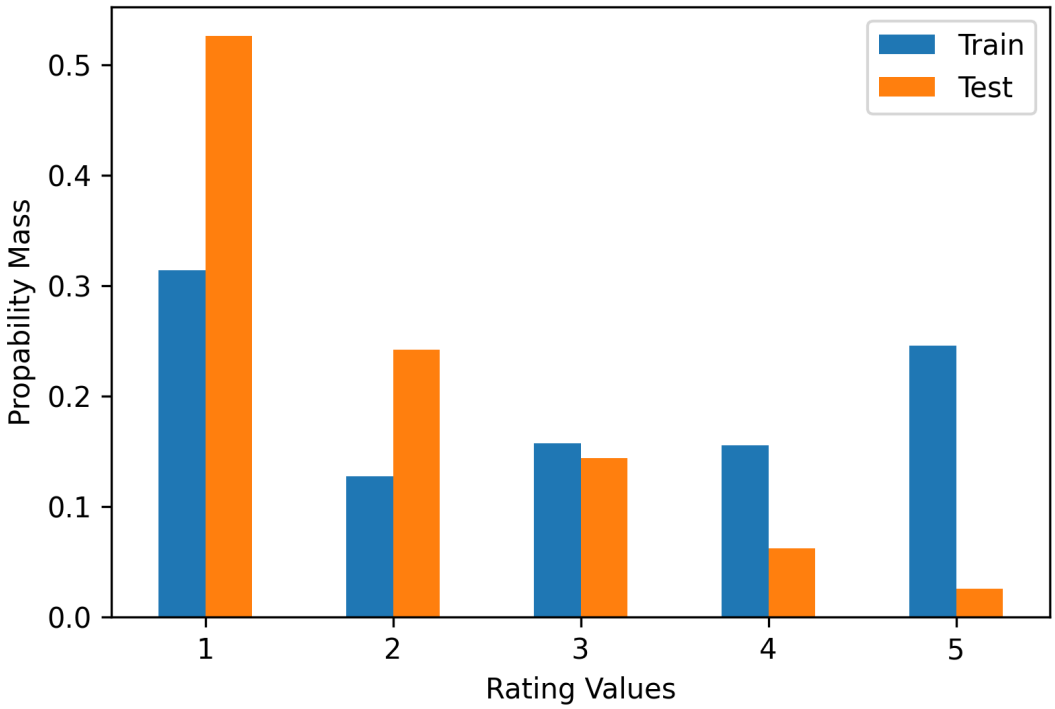
4.3.1 Mức độ cải thiện của MF-IPS với MF khi học trên tập COAT với tập YAHOO

Mức độ thiên lệch của tập dữ liệu Coat và tập dữ liệu Yahoo

Ta biết rằng bộ dữ liệu Coat và bộ dữ liệu Yahoo!R3 là những tập dữ liệu với tập test là MCAR trong khi tập training của chúng bị lệch. Nhóm em có tiến hành so sánh độ lệch của tập training với tập test trên hai bộ dữ liệu bằng độ đo KL - div hay còn gọi là độ đo "Kullback-Leibler

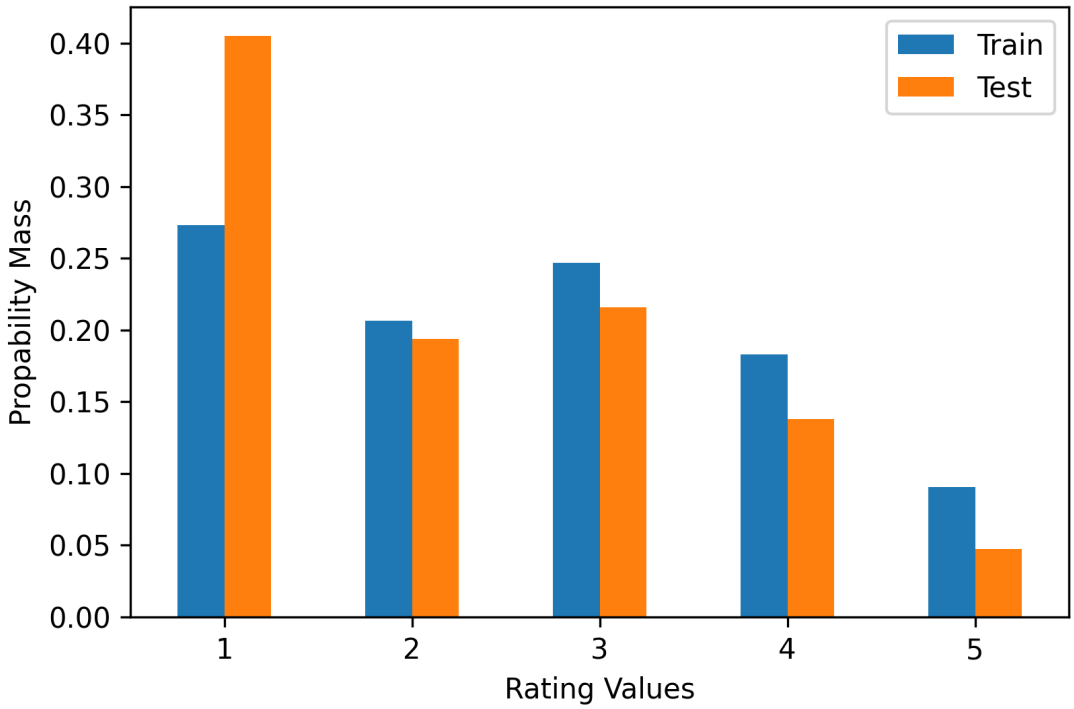
divergence" tương tự như [8], giá trị của độ đo này càng lớn chứng tỏ tập training và tập test càng có phân phối khác biệt nhau. Cụ thể, các kết quả nhóm chúng em thu được như sau:

- Đối với tập dữ liệu Coat, giá trị của độ đo KL - div là 0.049, phân phối của nó được thể hiện trong hình 4.3.
- Đối với tập dữ liệu Yahoo, giá trị của độ đo KL - div là 0.469, phân phối của nó được thể hiện trong hình 4.2.



Hình 4.2: Phân phối dữ liệu của tập training và tập test trên bộ dữ liệu Yahoo.

Qua hình 4.2, hình 4.3 và kết quả của độ đo KL - div ta có thể thấy chênh lệch về phân phối dữ liệu trong tập training và tập test của tập Yahoo lớn hơn tập Coat rất nhiều. Điều này có nghĩa là mức độ thiên lệch trong tập dữ liệu Yahoo cao hơn tập dữ liệu Coat nhiều, làm cho độ hiệu quả của phương pháp MF-IPS khi so với phương pháp MF sẽ được thể



Hình 4.3: Phân phối dữ liệu của tập training và tập test trên bộ dữ liệu Coat.

hiện nổi bật hơn ở trong tập Yahoo. Trong thí nghiệm tiếp theo nhóm em sẽ cho thấy rõ vấn đề này.

Cách tiến hành

Để biết được hiệu quả của phương pháp MF-IPS so với với phương pháp MF thông thường, nhóm em tiến hành thí nghiệm phương pháp MF-IPS và MF truyền thống trên cả hai bộ dữ liệu Coat và Yahoo. Để công bằng nhóm em sẽ sử dụng cùng một phương pháp ước lượng ma trận propensity đó là "Naive bayes" vì bộ dữ liệu Yahoo!R3 không có thông tin về người dùng - sản phẩm để thực hiện phương pháp "Logistic regression". Sau đó hai mô hình sẽ đánh giá độ lỗi MSE trên 95% dữ liệu tập test và 5% dữ liệu tập test còn lại sẽ được sử dụng để ước lượng propensity thông qua phương pháp "Naive bayes".

Kết quả

Bảng 4.2 cho thấy kết quả thí nghiệm của nhóm chúng em về độ lỗi MSE của 2 phương pháp MF và MF-IPS (với propensity được ước lượng thông qua "Naive bayes") trên 2 tập dữ liệu Yahoo và Coat.

	Yahoo	Coat
MF-IPS	0.976	1.080
MF	1.899	1.197

Bảng 4.2: Độ lỗi MSE của phương pháp MF và MF-IPS trên 2 tập dữ liệu Coat và Yahoo. Trong đó IPS được ước lượng thông qua "Naive bayes".

Có thể thấy độ lỗi của phương pháp MF-IPS thấp hơn độ lỗi của phương pháp MF trên cả 2 tập dữ liệu Coat và Yahoo. Đặc biệt, tập dữ liệu Yahoo có mức độ thiên lệch dữ liệu nhiều hơn làm cho mức độ cải thiện của MF-IPS so với MF lớn hơn nhiều khi so sánh với mức độ cải thiện trên tập Coat.

4.3.2 Mức độ cải thiện của MF-IPS với MF khi học trên tập MovieLens giả lập với mức độ thiên lệch của dữ liệu giảm dần

Tập MovieLens 100k giả lập

Như đã trình bày ở phần trước, ở thí nghiệm này nhóm chúng em sẽ tiến hành thí nghiệm trên tập dữ liệu MovieLens 100k. Vì thí nghiệm này yêu cầu kiểm soát được mức độ thiên lệch của dữ liệu, nên nhóm chúng em sẽ tiến hành tạo một tập dữ liệu MovieLens 100k giả lập.

Tập dữ liệu MovieLens 100k giả lập sẽ được tạo bằng cách sử dụng mô hình MF được huấn luyện trên toàn bộ tập dữ liệu MovieLens 100k, để điền toàn bộ giá trị còn thiếu trong ma trận tương tác giữa người dùng và sản phẩm MovieLens 100k. Với $[p_1, p_2, p_3, p_4, p_5]$ là phân bố của dữ liệu trong tập MovieLens 100k giả lập, p_1 là tỉ lệ đánh giá 1 trong toàn bộ dữ

liệu và tương tự với các đánh giá còn lại, thì phân bố dữ liệu sau khi hoàn thành ma trận giả lập là $[0.001, 0.027, 0.601, 0.364, 0.007]$. Có thể thấy các đánh giá trong ma trận giả lập này bị lệch nghiêm trọng, với số lượng đánh giá 3 và 4 chiếm số lượng rất lớn khi so với các đánh giá 1, 2 và 5. Điều này rất vô lý trong thực tế, vì vậy nhóm chúng em tiến hành điều chỉnh lại ma trận giả lập này tuân theo phân bố dữ liệu của tập test trong tập Yahoo!R3 là $[0.526, 0.242, 0.144, 0.062, 0.026]$.

Ngoài ra, dữ liệu quan sát được sẽ được ước lượng như sau: Với các đánh giá 4 và 5, propensity để quan sát được các đánh giá này k. Đối với các đánh giá < 4, propensity để quan sát được các đánh giá này là $k\alpha^{4-r}$, với r là các đánh giá và $\alpha \in [0, 1]$ là tham số điều chỉnh mức độ thiên lệch dữ liệu. Với mỗi α. k được đặt sao cho số lượng dữ liệu quan sát được chỉ chiếm 5% trên toàn bộ ma trận. Bằng cách thay đổi giá trị α, chúng ta có thể thay đổi được mức độ thiên lệch dữ liệu. Khi $\alpha = 1$ dữ liệu sẽ được phát sinh hoàn toàn ngẫu nhiên theo phân phối đều (vì khi đó propensity của toàn bộ đánh giá đều bằng k). Khi $\alpha \to 0$ thì dữ liệu quan sát được chỉ chứa các giá trị 4 và 5 (vì khi đó propensity của các đánh giá 1,2 và 3 bằng 0). Lưu ý là với $\alpha = 0.25$ thì phân bố của dữ liệu quan sát được sẽ giống với phân bố của dữ liệu MovieLens 100k ban đầu ([0.06, 0.11, 0.27, 0.35, 0.21] trong tập dữ liệu gốc và [0.06, 0.10, 0.25, 0.42, 0.17] trên tập dữ liệu quan sát được)

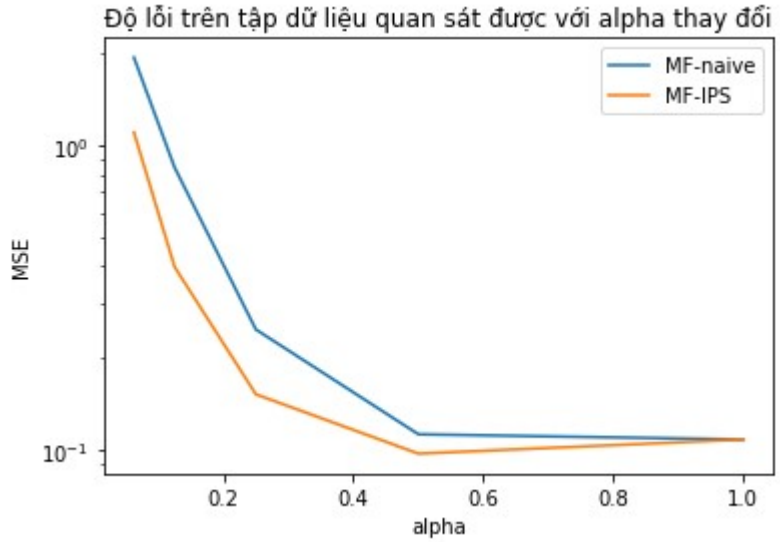
Cách tiến hành

Sau khi có bộ dữ liệu quan sát được ta sẽ tiến hành huấn luyện 2 mô hình MF và MF-IPS trên tập dữ liệu này. Hai mô hình này sẽ được cross-validation để chọn tham số λ khác nhau và tham số d sẽ được cố định với $d = 20$. Ta sẽ tiến hành thay đổi giá trị α từ [0.0625, 0.125, 0.25, 0.5, 1] để điều chỉnh mức độ thiên lệch dữ liệu trên tập dữ liệu quan sát được được sử dụng để huấn luyện mô hình. Tập dữ liệu quan sát được sẽ được lấy từ tập dữ liệu MovieLens 100k giả lập tuân theo propensity của mỗi đánh giá, sao cho dữ liệu quan sát được chỉ chiếm 5% tập dữ liệu MovieLens

100k giả lập.

Kết quả

Hình 4.4 mô tả kết quả thu được của nhóm em với độ lỗi MSE trên 2 mô hình MF và MF-IPS khi tiến hành thay đổi mức độ thiên lệch dữ liệu bằng cách thay đổi giá trị của α.



Hình 4.4: Hình ảnh minh họa sự cải thiện về độ lỗi của MF và MF-IPS khi thay đổi mức độ thiên lệch của dữ liệu

Khi α nhỏ mức độ thiên lệch của dữ liệu cao (trong dữ liệu quan sát được đa phần chỉ chứa điểm đánh giá 4 và 5) thì độ lỗi trên 2 mô hình MF và MF-IPS rất cao. Mặc dù vậy, độ lỗi của mô hình MF-IPS vẫn có độ lỗi thấp hơn nhiều mô hình MF truyền thống.

Khi α lớn mức độ thiên lệch dữ liệu của dữ liệu giảm dần (phân phối của các điểm đánh giá đều nhau hơn) độ lỗi trên 2 mô hình này có sự cải thiện đáng kể và gần như xê xích nhau không nhiều. Mặc dù vậy mô hình MF-IPS vẫn cho độ lỗi thấp hơn mô hình MF.

Tóm lại, khi dữ liệu bị thiên lệch nhiều phương pháp MF-IPS đã cho thấy độ hiệu quả của nó trong việc kiểm soát sự thiên lệch và cho kết quả cải thiện đáng kể trên tập dữ liệu bị thiên lệch này. Mặt khác, khi dữ liệu

không bị thiên lệch (dữ liệu được phát sinh theo phân phối đều) phương pháp MF-IPS sẽ cho kết quả giống với phương pháp MF truyền thống, bởi vì lúc này dữ liệu không còn bị thiên lệch nên propensity của nó là như nhau trong mọi trường hợp làm cho MF và MF-IPS lúc này không có gì khác nhau.

4.4 Ảnh hưởng của việc ước lượng ma trận propensity tới việc học của MF-IPS

Trong phần này, nhóm em sẽ thiết kế các thí nghiệm để làm rõ ảnh hưởng của việc ước lượng ma trận propensity đến việc học của MF-IPS. Việc học của MF-IPS có thể bị ảnh hưởng bởi hiệu quả của việc ước lượng ma trận propensity, ta sẽ tiến hành xem xét mức độ hiệu quả của việc ước lượng ma trận propensity thông qua hai thiết lập sau:

- Phương pháp ước lượng sẽ ảnh hưởng đến hiệu quả của việc ước lượng. Do đó nhóm sẽ tiến hành ước lượng ma trận propensity bằng cả phương pháp "Logistic Regression" và phương pháp "Naive Bayes" trên cùng bộ dữ liệu Coat, vì chỉ bộ dữ liệu này mới chứa thông tin của người dùng và sản phẩm để có thể tiến hành ước lượng ma trận propensity bằng phương pháp "Logistic Regression".

- Khi sử dụng phương pháp "Naive Bayes" để ước lượng lượng ma trận propensity, số lượng mẫu MCAR được sử dụng có thể ảnh hưởng đến hiệu quả của việc ước lượng này và từ đó, ảnh hưởng đến quá trình học của MF-IPS.

4.4.1 So sánh mức độ cải thiện của MF-IPS với MF bằng các phương pháp ước lượng ma trận propensity khác nhau

Như đã trình bày ở trên, thí nghiệm này nhóm em sẽ sử dụng bộ dữ liệu Coat. Với phương pháp ước lượng ma trận bằng propensity bằng "Naive Bayes", nhóm em sẽ sử dụng 5% của tập test làm mẫu MCAR và chỉ đo độ lỗi của việc học MF-IPS trên 95% còn lại của tập test. Với phương pháp ước lượng ma trận propensity bằng phương pháp "Logistic Regression", nhóm sẽ sẽ sử dụng thông tin của người dùng và sản phẩm để ước lượng. Để công bằng, nhóm cũng sẽ đo độ lỗi MF-IPS trên cùng một tập test được sử dụng bên trên. Ngoài ra nhóm cũng sử dụng phương pháp MF thông thường và dĩ nhiên cũng sẽ đo lại độ lỗi trên cùng một tập test để xem xét mức độ cải thiện của MF-IPS với MF bằng các phương pháp ước lượng ma trận propensity khác nhau. Kết quả mà nhóm em thu được như sau:

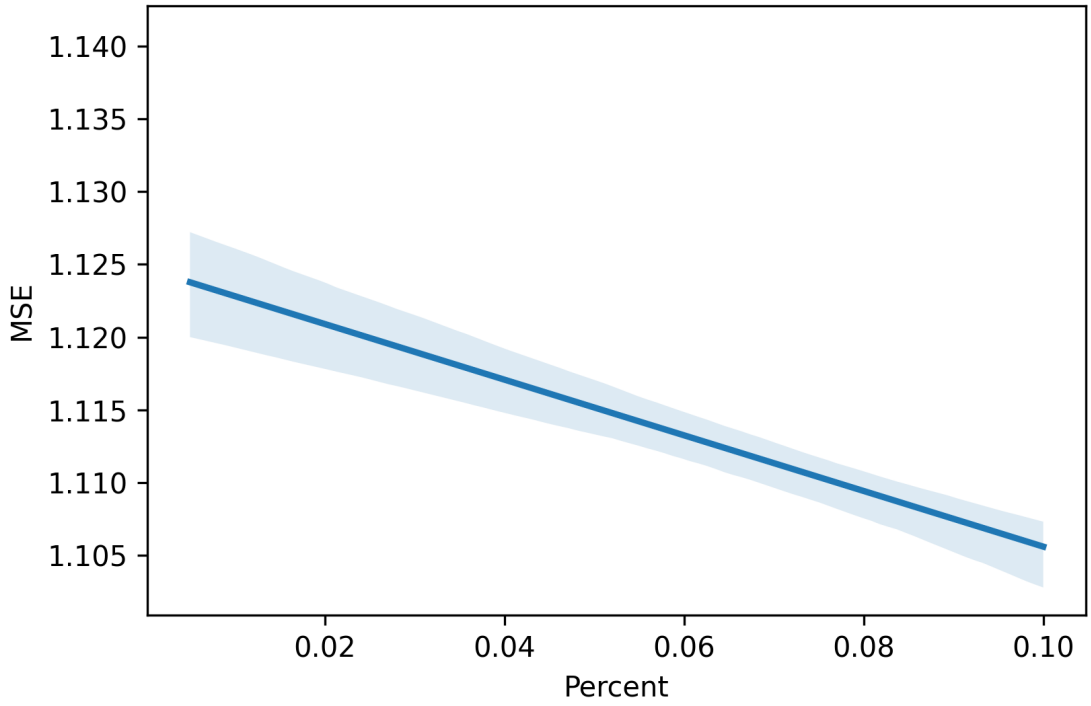
	MAE	MSE
MF	0.9138	1.1973
MF-IPS-LR	0.9037	1.1967
MF-IPS-NB	**0.8529**	**1.0795**

Bảng 4.3: Độ lỗi MSE và MAE của hai mô hình MF và MF-IPS với hai phương pháp ước lượng ma trận propensity khác nhau trên tập dữ liệu Coat.

Bảng 4.3 cho ta thấy rằng phương pháp ước lượng ma trận propensity có ảnh hưởng đáng kể đến quá trình học của MF-IPS. Phương pháp "Naive Bayes" cho kết quả ước lượng tốt hơn phương pháp "Logistic Regression" vì theo như ví dụ ban đầu và theo nghiên cứu của Hu Nan [6], giá trị của mỗi đánh giá sẽ phần lớn quyết định khả năng đánh giá đó xuất hiện, hơn nữa việc sử dụng được phần nào phân phối của tập test cũng sẽ làm tăng hiệu quả của phương pháp ước lượng mà trận propensity bằng "Naive Bayes".

4.4.2 So sánh mức độ cải thiện của MF-IPS với MF khi ước lượng ma trận propensity bằng "Naive Bayes" với số lượng dữ liệu MCAR khác nhau

Ở thí nghiệm này, nhóm em sẽ sử dụng phương pháp "Naive Bayes" để ước lượng ma trận propensity. Tuy nhiên nhóm chỉ đo độ lỗi trên 90% của tập test, phần còn lại sẽ được sử dụng làm mẫu MCAR cho quá trình ước lượng ma trận propensity, lượng tập test được tiết lộ lần lượt sẽ là [0.5%, 1%, 5%, 10%]. Nhóm em tiến hành thử nghiệm và kết quả thu được như hình 4.5, trong đó vùng bóng mờ là khoảng tin cậy 95% trong 30 lần thử nghiệm.



Hình 4.5: Hình ảnh minh họa sự cải thiện về độ lỗi của MF và MF-IPS với lượng tiết lộ tập test khác nhau.

Nhận xét: ta thấy rằng khi tăng lượng dữ liệu MCAR cho quá trình ước lượng ma trận propensity, việc học của MF-IPS sẽ có cải thiện. Điều

này cũng khá dễ hiểu vì khi sử dụng càng nhiều lượng dữ liệu MCAR thì việc tính toán xác suất $P(Y = r)$ càng chính xác, giúp cho việc ước lượng propensity ít bị sai lệch hơn.

Chương 5

Tổng kết và hướng phát triển

5.1 Tổng kết

Trong khóa luận này, nhóm em đã tìm hiểu về một phương pháp hiệu quả và mạnh mẽ để giải quyết vấn đề dữ liệu bị lệch dựa trên propensity. Phương pháp này mang lại nhiều ưu điểm như:

- **Đơn giản để thực hiện**: về cơ bản, ta chỉ cần sử dụng một số phương pháp đơn giản như "Naive bayes" hay "Logistic regression" để ước lượng propensity, sau đó ta đơn thuần áp dụng một thay đổi nhỏ ở hàm mục tiêu bằng cách nhân với nghịch đảo của propensity.

- **Hiệu quả**: bằng các thí nghiệm trong chương 4, ta thấy được phương pháp MF-IPS cải thiện đáng kể độ lỗi của mô hình trên các tập dữ liệu thế giới thực so với phương pháp MF. Trong các tập dữ liệu này, tập training bị lệch do người dùng tự chọn sản phẩm để đánh giá, còn tập test không bị lệch do các sản phẩm được hiển thị ngẫu nhiên tới người dùng theo phân phối đều, và người dùng sẽ đánh giá các sản phẩm được hiển thị ngẫu nhiên này. Do đó, tập test thể hiện được chính xác sở thích tự nhiên của người dùng. Chính vì vậy, việc giảm độ lỗi trên tập test so với phương pháp MF thông thường phần nào cho thấy rằng phương pháp MF-IPS mô hình hóa sở thích của người dùng tốt hơn.

- **Không cần sử dụng đến các biến ẩn**: ta thấy được rằng việc người dùng tự đánh giá sẽ phụ thuộc vào nhiều yếu tố tiềm ẩn như sản phẩm có được gợi ý bởi bạn bè của người dùng hay không, tâm trạng của người dùng lúc xem phim hay sử dụng sản phẩm đó có tích cực hay tiêu cực... Với phương pháp MF-IPS này, ta không cần phải kiểm soát các biến ẩn đó mà vẫn có thể phản ánh chính xác được phần nào sở thích của người dùng.

- **Ít giả định về mô hình hơn**: Với các mô hình học máy thông thường, ta cần phải có nhiều giả định, ví dụ như tập dữ liệu quan sát được và dữ liệu thực tế phải cùng phân phối với nhau. Nói cách khác là ta cần giả định dữ liệu mà ta quan sát được mang tính đại diện cho dữ liệu thực tế. Tuy nhiên ta có thể thấy rằng ta có thể học được tốt trên tập dữ liệu MNAR - tập dữ liệu không được phát sinh ngẫu nhiên theo phân phối đều từ tập dữ liệu thực tế, bằng cách đánh lại trọng số trên tập dữ liệu quan sát được.

Bên cạnh những ưu điểm trên, phương pháp MF-IPS cũng có một số hạn chế nhất định:

- Hạn chế lớn nhất của phương pháp này là việc ước lượng được các xu hướng chính xác một cách hoàn hảo gần như là điều không thể, và kết quả của việc ước lượng này sẽ ảnh hưởng đến quá trình học của MF-IPS.

- Phương pháp MF-IPS có thể khiến mô hình của ta sẽ có phương sai cao, khiến cho mô hình không ổn định.

5.2 Hướng phát triển

Do bước ước lượng xu hướng và bước Matrix Factorization tách biệt với nhau, nên ta có thể dễ dàng ứng dụng bước ước lượng xu hướng cho các bài toán học máy khác, nếu như dữ liệu ta thu thập được trong bài

toán đó gặp vấn đề MNAR, việc ước lượng xu hướng có thể dễ dàng thực hiện bằng nhiều phương pháp ước lượng xác suất có điều kiện khác nhau, ta có thể tham khảo thêm ở bài báo [1]. Ngoài ra, với việc kết nối giữa bài toán xây dựng hệ thống gợi ý và bài toán suy diễn nhân quả, ta cũng có thể sử dụng nhiều phương pháp hiệu quả hơn của bài toán suy diễn nhân quả để xây dựng hệ thống gợi ý với hiệu quả tốt hơn, phản ánh được chính xác hơn sở thích của người dùng.

Tài liệu tham khảo

Tiếng Anh

[1] DF, McCaffrey, G, Ridgeway, and AR., Morral. "Propensity score estimation with boosted regression for evaluating causal effects in observational studies". In: *Psychological methods* 9.4 (2004), 403—425. DOI: https://10.1037/1082-989X.9.4.403.

[2] Imbens, W., Guido, and Rubin, Donald B. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781139025751.

[3] Marlin et al. "Collaborative Prediction and Ranking with Non-Random Missing Data". In: RecSys '09 (2009), 5—12. DOI: 10.1145/1639714.1639717. URL: https://doi.org/10.1145/1639714.1639717.

[4] Marlin, Benjamin et al. "Collaborative Filtering and the Missing at Random Assumption". In: *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. 2007, pp. 267–275.

[5] Mislevy, Robert J. In: *Journal of Educational Statistics* 16.2 (1991), pp. 150–155. ISSN: 03629791. URL: http://www.jstor.org/stable/1165119 (visited on 06/21/2022).

[6] Nan, Hu, Pavlou, Paul A., and Zhang, Jie Jennifer. "On self-selection biases in online product reviews". In: *MIS Q.* 40.2 (2017), pp. 449–471. DOI: https://doi.org/10.25300/MISQ/2017/41.2.06.

[7] Owen, Art B. *Monte Carlo theory, methods and examples*. 2013.

[8] Saito and Yuta. "Asymmetric tri-training for debiasing missing-not-at-random explicit feedback". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020.

[9] Schnabel, Tobias et al. "Recommendations as Treatments: Debiasing Learning and Evaluation". In: *ICML* 48 (2016), 1670—1679.

[10] Swaminathan et al. "The Self-Normalized Estimator for Counterfactual Learning". In: *Advances in Neural Information Processing Systems*. Ed. by Cortes, C. et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/39027dfad5138c9ca0c474d71db915c3-Paper.pdf.

[11] Wendel, J. G. "Groups and Conditional Monte Carlo". In: *The Annals of Mathematical Statistics* 28.4 (1957), pp. 1048–1052. ISSN: 00034851. URL: http://www.jstor.org/stable/2237072 (visited on 06/22/2022).