

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

Lê Quốc Cường
Lê Đào Duy Trọng

XÂY DỰNG HỆ THỐNG ĐỀ XUẤT
SẢN PHẨM DỰA TRÊN PHƯƠNG PHÁP
HỌC TĂNG CƯỜNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 06/2023

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

Lê Quốc Cường - 19120057
Lê Đào Duy Trọng - 19120145

XÂY DỰNG HỆ THỐNG ĐỀ XUẤT
SẢN PHẨM DỰA TRÊN PHƯƠNG PHÁP
HỌC TĂNG CƯỜNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN

ThS. Trần Trung Kiên

TS. Nguyễn Ngọc Thảo

Tp. Hồ Chí Minh, tháng 06/2023

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.

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

Chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 23 tháng 6 năm 2023

Nhóm sinh viên thực hiện

Lê Quốc Cường

Lê Đào Duy Trọng



fit@hcmus

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

ĐỀ CƯƠNG KHOÁ LUẬN TỐT NGHIỆP

**XÂY DỰNG HỆ THỐNG ĐỀ XUẤT SẢN
 PHẨM DỰA TRÊN PHƯƠNG PHÁP
 HẠC TĂNG CƯỜNG**

(Building Recommender System Using Reinforcement Learning)

1 THÔNG TIN CHUNG

Người hướng dẫn:

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

[Nhóm] Sinh viên thực hiện:

1. Lê Quốc Cường (MSSV: 19120057)
2. Lê Đào Duy Trọng (MSSV: 19120145)

Loại đề tài: Nghiên cứu

Thời gian thực hiện: Từ 01/2023 đến 06/2023

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Trong thời đại công nghệ thông tin ngày nay, các doanh nghiệp và cửa hàng trực tuyến đang tìm cách sử dụng các hệ thống đề xuất sản phẩm để cung cấp những sản phẩm phù hợp với sở thích của từng khách hàng. Điều này giúp tăng trải nghiệm của người dùng, kích thích nhu cầu mua sắm và tăng doanh thu cho các doanh nghiệp và cửa hàng trực tuyến.

Một cách cụ thể, bài toán xây dựng hệ thống đề xuất sản phẩm được phát biểu như sau:

- Cho đầu vào là thông tin về ngữ cảnh bao gồm thông tin của người dùng (thời gian, địa điểm, lịch sử tra cứu...) và thông tin của các sản phẩm hiện có.
- Yêu cầu: xây dựng được một hệ thống mà có thể đề xuất các sản phẩm phù hợp với người dùng dựa trên thông tin ngữ cảnh được cung cấp.

Khó khăn lớn của bài toán này là thông tin của người dùng có thể không đủ (ví dụ người dùng mới hoặc người dùng cũ nhưng sở thích thay đổi theo thời gian) để hệ thống có thể đủ hiểu người dùng và đề xuất các sản phẩm phù hợp.

Một hướng tiếp cận gần đây có thể giúp giải quyết khó khăn ở trên là sử dụng học tăng cường, và đây cũng là hướng tiếp cận mà đề tài tập trung tìm hiểu.

2.2 Mục tiêu đề tài

- Nắm được ý tưởng của các hướng tiếp cận đã được đề xuất để giải quyết bài toán xây dựng hệ thống đề xuất sản phẩm, từ đó chọn ra một phương pháp tốt (ứng với một bài báo uy tín) để tập trung tìm hiểu sâu.
- Nắm rõ lý thuyết của các phương pháp đã chọn.

- Cài đặt lại phương pháp đã chọn để có thể đạt được các kết quả như trong bài báo gốc, thực hiện thêm các thí nghiệm ngoài bài báo trên cùng dữ liệu của bài báo để thấy rõ hơn về ưu và nhược điểm của phương pháp.
- Mở rộng phương pháp với các dữ liệu khác ngoài bài báo (nếu có đủ thời gian).
- Rèn luyện những kỹ năng mềm cần thiết khác: Kỹ năng làm việc nhóm, quản lý công việc, thuyết trình...

2.3 Phạm vi của đề tài

Đề tài tìm hiểu và cài đặt lại phương pháp được đề xuất trong một bài báo uy tín. Đề tài sử dụng dữ liệu mà bài báo sử dụng. Ngoài ra, đề tài có thể có thêm các thí nghiệm ngoài bài báo (trên cùng dữ liệu của bài báo) để thấy rõ hơn về ưu và nhược điểm của phương pháp. Nếu có đủ thời gian thì đề tài có thể mở rộng phương pháp với các dữ liệu ngoài bài báo.

2.4 Cách tiếp cận dự kiến

Để xây dựng nên các hệ thống đề xuất sản phẩm, chúng ta có thể sử dụng các phương pháp truyền thống như: Content-based Filtering [1], Collaborative Filtering [2],... Tuy nhiên, các phương pháp này đề xuất sản phẩm dựa trên thông tin hiện có của người dùng, chỉ thực hiện khai thác (exploit) nên sẽ gặp khó khăn khi thông tin về người dùng không đủ.

Trong những năm gần đây, phương pháp học tăng cường (Reinforcement Learning) đã được áp dụng để giải quyết bài toán đề xuất sản phẩm. Phương pháp này không luôn luôn thực hiện khai thác (exploit) như các phương pháp truyền thống mà sẽ kết hợp với thực hiện khai phá (explore). Khi thấy cần phải hiểu hơn về người dùng thì phương pháp này sẽ đề xuất các sản phẩm để hướng tới mục tiêu là hiểu hơn về người dùng. Sau khi đã hiểu hơn về người dùng, phương pháp này sẽ cập nhật chiến lược đề xuất sản phẩm cho người dùng sao cho chiến lược này là tốt nhất với các thông tin hiện có về người dùng (thực hiện khai thác). Bằng

cách kết hợp khai phá và khai thác như vậy, phương pháp này có thể giải quyết được vấn đề thông tin người dùng không đủ như người dùng mới hoặc người dùng cũ có sở thích thay đổi. Các tác giả trong bài báo [4] đã đưa ra một cách đơn giản và hiệu quả để thực hiện phương pháp này.

Ngoài ra, phương pháp học sâu cũng có thể được kết hợp với học tăng cường (Deep Reinforcement Learning) để xây dựng hệ thống đề xuất như bài báo [3]. Phương pháp này vượt trội hơn với phương pháp học tăng cường thông thường ở khả năng xử lý dữ liệu phức tạp, tính tự động hoá cao và hiệu suất tốt hơn. Tuy nhiên, nhược điểm của phương pháp này là đòi hỏi nhiều dữ liệu, tài nguyên tính toán cao hơn so với phương pháp học tăng cường thông thường.

Mỗi phương pháp đều có ưu và nhược điểm riêng. Tuy nhiên, vì giới hạn về thời gian, kiến thức và nguồn tài nguyên nên chúng em chọn phương pháp học tăng cường được sử dụng trong bài báo [4] để giải quyết bài toán xây dựng hệ thống đề xuất sản phẩm.

2.5 Kết quả dự kiến của đề tài

- Cài đặt lại được từ đầu phương pháp được đề xuất trong bài báo [4].
- Có được các kết quả thí nghiệm cho thấy mã nguồn tự cài đặt cho các kết quả tương tự với bài báo gốc.
- Có được các kết quả thí nghiệm ngoài bài báo (trên cùng dữ liệu của bài báo) để giúp thấy rõ hơn về ưu và nhược điểm của phương pháp.
- Nếu còn thời gian thì có thể cài đặt để mở rộng phương pháp với các dữ liệu ngoài bài báo và có được các kết quả thí nghiệm tương ứng.

2.6 Kế hoạch thực hiện

Công việc	Thời gian thực hiện
Tìm hiểu và lựa chọn nội dung đề tài khóa luận	01/01/2023 - 21/02/2023
Lựa chọn paper theo chủ đề đã chọn	22/01/2023 - 13/02/2023
Đọc, hiểu nội dung chính của paper	14/02/2023 - 21/02/2023
Viết đề cương khóa luận	22/02/2023 - 07/03/2023
Tìm hiểu các kiến thức cần thiết trong paper	08/03/2023 - 08/04/2023
Tìm hiểu, chạy thử code theo phương pháp Contextual bandit	09/04/2023 - 25/04/2023
Cài đặt lại phương pháp Contextual bandit	26/04/2023 - 10/05/2023
Thử nghiệm đánh giá phương pháp	11/05/2023 - 18/05/2023
Hoàn thành code, viết cuốn	19/05/2023 - 26/05/2023
Chỉnh sửa cuốn, làm slide báo cáo	27/05/2023 - 15/06/2023

Tài liệu

- [1] Pasquale Lops, Marco de Gemmis, Giovanni Semeraro *"Content-based Recommender Systems: State of the Art and Trends"*. 2011.
- [2] Yehuda Koren, Robert Bell, Chris Volinsky *"Matrix Factorization Techniques for Recommender Systems"*. 2009.
- [3] YXiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Jiliang Tang, Hui Liu *"Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems"*. 2019.
- [4] Lihong Li, Wei Chu, John Langford, Robert E. Schapire *"A Contextual-Bandit Approach to Personalized News Article Recommendation"*. 2012.

XÁC NHẬN
CỦA NGƯỜI HƯỚNG DẪN
(Ký và ghi rõ họ tên)



Trần Trung Kiên

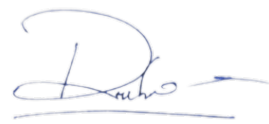


Nguyễn Ngọc Thảo

TP. Hồ Chí Minh, ngày 03 tháng 04 năm 2023
NHÓM SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)



Lê Quốc Cường



Lê Đào Duy Trọng

Mục lục

1	Giới thiệu	1
1.1	Phát biểu bài toán đề xuất sản phẩm	1
1.2	Thách thức của bài toán	3
1.3	Phương pháp giải quyết bài toán mà chúng em tìm hiểu .	4
1.4	Bố cục của khoá luận	7
2	Kiến thức nền tảng	8
2.1	Phương pháp Bandit	8
2.2	Các giải thuật trong Bandit	14
2.2.1	Giải thuật ϵ -Greedy	14
2.2.2	Giải thuật UCB	19
3	Giải thuật LinUCB để giải quyết bài toán đề xuất sản phẩm	25
3.1	Áp dụng ngữ cảnh và mô hình tuyến tính trong LinUCB .	25
3.2	LinUCB với các mô hình tuyến tính riêng biệt	27
3.3	LinUCB với các mô hình tuyến tính có sự chia sẻ	32
4	Kết quả thí nghiệm	38
4.1	Thiết lập các thí nghiệm	38
4.1.1	Tập dữ liệu	38
4.1.2	Phương pháp rút trích đặc trưng	41
4.1.3	Phương pháp đánh giá các giải thuật	42

4.2	Thí nghiệm 1: So sánh kết quả cài đặt của khóa luận với bài báo gốc	45
4.3	Thí nghiệm 2: Đánh giá hiệu suất giải thuật LinUCB với các giá trị siêu tham số khác nhau.	46
4.4	Thí nghiệm 3: So sánh LinUCB Disjoint với LinUCB Hybrid	47
4.5	Thí nghiệm 4: So sánh LinUCB với Lin ϵ -Greedy	49
5	Tổng kết và hướng phát triển	51
5.1	Tổng kết	51
5.2	Hướng phát triển	52
	Tài liệu tham khảo	53

Danh sách hình

1.1	Minh hoạ hệ thống đề xuất của Yahoo! đề xuất bài tin tức cho người đọc.	2
2.1	Minh họa máy đánh bạc (slot machine)	9
2.2	Quy trình thực hiện của giải thuật ϵ -Greedy	16
4.1	Hình ảnh phần nổi bật của “Today Module” trên trang chủ của “Yahoo!”. Theo mặc định bài tin tức ở vị trí F1 sẽ được hiển thị nổi bật ở vị trí “STORY”.	39
4.2	Đánh giá hiệu suất giải thuật LinUCB với các giá trị siêu tham số khác nhau trên tập “tuning data”.	47
4.3	So sánh hiệu suất của LinUCB Hybrid và LinUCB Disjoint trên tập “evaluation data”	48
4.4	So sánh hiệu suất của Lin ϵ -Greedy Hybrid và LinUCB Hybrid trên tập “evaluation data”	49
4.5	So sánh hiệu suất của Lin ϵ -Greedy Hybrid và LinUCB Hybrid trên tập “evaluation data”	50

Danh sách bảng

4.1	Kết quả của cài đặt của khóa luận và của bài báo gốc với độ đo là “Relative CTR” trên tập “evaluation data”	45
-----	---	----

Tóm tắt

Trong thời đại công nghệ thông tin ngày nay, các doanh nghiệp và cửa hàng trực tuyến đang tìm cách sử dụng các hệ thống đề xuất sản phẩm để cung cấp những sản phẩm phù hợp với sở thích của từng khách hàng. Điều này giúp tăng trải nghiệm của người dùng, kích thích nhu cầu mua sắm và tăng doanh thu cho các doanh nghiệp và cửa hàng trực tuyến. Vì vậy trong khoá luận này, nhóm chúng em sẽ trình bày về phương pháp “Học tăng cường” (Reinforcement Learning) mà nhóm chúng em tìm hiểu để giải quyết bài toán xây dựng hệ thống đề xuất sản phẩm. Phương pháp này cho thấy sự cải thiện đáng kể về độ chính xác trong việc đề xuất sản phẩm phù hợp với người dùng so với các phương pháp truyền thống khác.

Chương 1

Giới thiệu

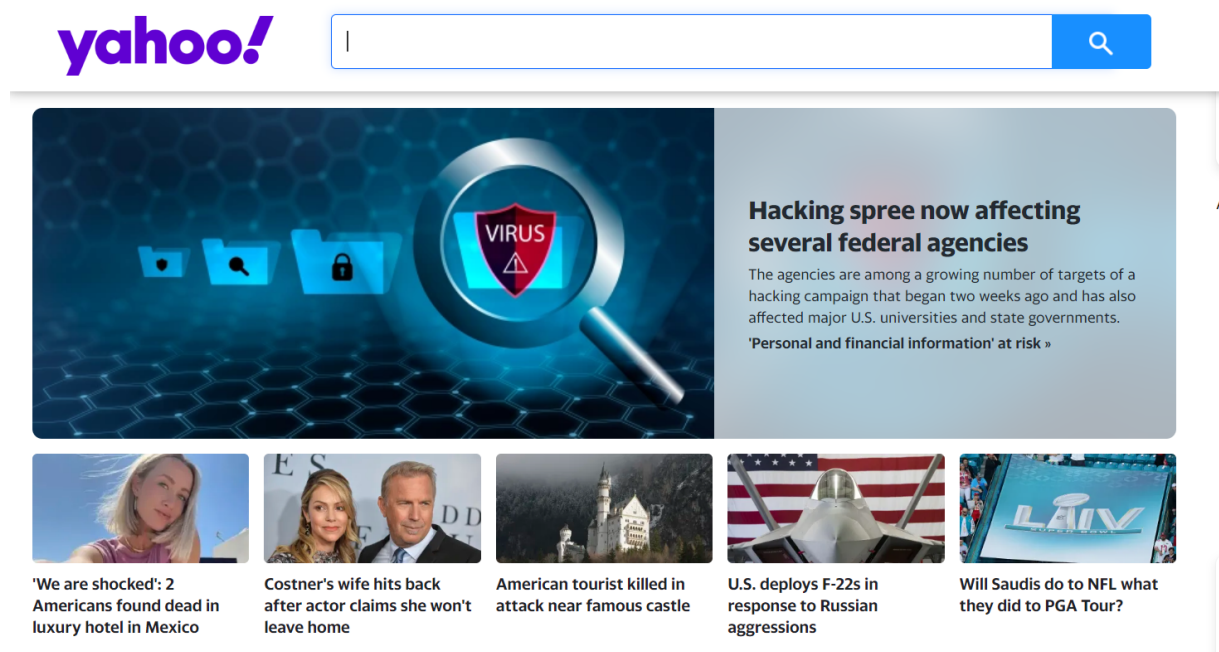
Trong chương này, đầu tiên nhóm chúng em sẽ phát biểu về bài toán đề xuất sản phẩm cũng như ý nghĩa của nó đối với cuộc sống hiện nay. Tiếp theo, chúng em sẽ nêu ra những thách thức của bài toán. Sau đó, chúng em sẽ trình bày phương pháp “Học tăng cường” (Reinforcement Learning) mà chúng em tìm hiểu để giải quyết bài toán đề xuất sản phẩm. Cuối cùng, chúng em sẽ trình bày về bố cục các phần còn lại của khóa luận.

1.1 Phát biểu bài toán đề xuất sản phẩm

Trong thời đại ngày nay, sự đa dạng và số lượng các sản phẩm dành cho con người ngày càng tăng. Điều này đã tạo ra một thách thức cho cả người tiêu dùng lẫn các doanh nghiệp. Đối với người dùng, đó là làm thế nào để có thể tìm kiếm và lựa chọn những sản phẩm phù hợp trong một danh mục sản phẩm vô cùng đa dạng. Với hàng ngàn sản phẩm có sẵn trên thị trường, việc tìm kiếm một sản phẩm phù hợp với cá nhân trở nên phức tạp và mất thời gian. Trong khi đó, đối với các doanh nghiệp, cửa hàng trực tuyến, vấn đề đặt ra là làm thế nào để có thể thu hút người dùng trải nghiệm, mua sắm các sản phẩm của họ.

Để giải quyết những vấn đề này, hệ thống đề xuất sản phẩm đã ra đời nhằm giúp các doanh nghiệp và cửa hàng trực tuyến cung cấp những sản

phẩm phù hợp với sở thích của từng khách hàng, từ đó kích thích nhu cầu mua sắm và tăng doanh thu; đồng thời, hệ thống đề xuất sản phẩm còn giúp cho trải nghiệm của khách hàng được cá nhân hoá, tiết kiệm thời gian lựa chọn sản phẩm. Trong thực tế, đã có nhiều hệ thống đề xuất sản phẩm, có thể kể đến như là hệ thống đề xuất video của Youtube, hệ thống đề xuất phim của Netflix, hệ thống đề xuất bài tin tức của Yahoo! (Hình 1.1) cùng nhiều hệ thống đề xuất khác.



Hình 1.1: Minh hoạ hệ thống đề xuất của Yahoo! đề xuất bài tin tức cho người đọc.

Bài toán đề xuất sản phẩm được phát biểu như sau:

- Đầu vào là dữ liệu về ngữ cảnh bao gồm thông tin của người dùng (thời gian, địa điểm, lịch sử tra cứu...) và thông tin của sản phẩm hiện có. Dữ liệu này được đưa vào mô hình của các hệ thống đề xuất dưới dạng các ma trận đặc trưng của người dùng và ma trận đặc trưng của sản phẩm.
- Yêu cầu: xây dựng được một hệ thống mà có thể đề xuất các sản

phẩm phù hợp với người dùng dựa trên thông tin về ngữ cảnh được cung cấp.

1.2 Thách thức của bài toán

Một trong những khó khăn lớn của bài toán đề xuất sản phẩm là thiếu thông tin của người dùng hoặc thông tin người dùng có thể không đủ để hệ thống có thể hiểu sâu về người dùng và đề xuất những sản phẩm phù hợp. Điều này đặc biệt phức tạp đối với những người dùng mới, vì thông tin của họ chưa được thu thập trong hệ thống, hoặc đối với những người dùng cũ nhưng sở thích của họ có thể thay đổi theo thời gian.

Một ví dụ thực tế để minh họa khó khăn của bài toán đề xuất sản phẩm có thể là trong lĩnh vực thương mại điện tử. Giả sử chúng ta có một người dùng mới truy cập vào một trang web bán hàng trực tuyến để tìm kiếm một sản phẩm điện. Với thông tin cung cấp ban đầu của người dùng mới, hệ thống chỉ có rất ít hoặc không có thông tin về sở thích, lựa chọn trước đây, hoặc hành vi mua hàng của người dùng này. Điều này tạo ra một thách thức lớn đối với hệ thống đề xuất sản phẩm, vì không có đủ dữ liệu để hiểu rõ người dùng và đưa ra những đề xuất phù hợp. Ngoài ra, hệ thống cũng phải đối mặt với những khó khăn khi sở thích của người dùng thay đổi theo thời gian. Ví dụ, người dùng có thể đã từng quan tâm đến các sản phẩm điện tử như điện thoại thông minh, nhưng sau một thời gian, sở thích của họ có thể chuyển sang các sản phẩm khác như máy tính bảng hoặc đồ điện tử gia đình. Điều này đòi hỏi hệ thống phải có khả năng cập nhật và thích ứng để đưa ra những đề xuất mới phù hợp với sở thích hiện tại của người dùng.

Vấn đề khám phá và khai thác là một phần quan trọng trong giải quyết vấn đề này. Khi đối diện với người dùng mới hoặc người dùng cũ có những thay đổi trong sở thích, hệ thống đề xuất sản phẩm cần có khả năng khám phá để tìm hiểu thêm về người dùng. Điều này có thể được thực hiện bằng cách tiến hành các thử nghiệm hoặc thu thập thông tin phản hồi từ người

dùng, từ đó nắm bắt được những yếu tố mới và cập nhật sở thích cá nhân của họ.

Nhưng việc chỉ dựa vào khám phá không đủ để đảm bảo sự cá nhân hoá và đề xuất chính xác cho từng người dùng. Đó là lý do tại sao vấn đề khai thác cũng được đặt ra. Bằng cách khai thác thông tin đã có về người dùng, như lịch sử mua hàng, hoạt động trước đây hoặc thông tin cá nhân khác, hệ thống có thể tận dụng và áp dụng các phương pháp máy học và các thuật toán đề xuất sản phẩm để đưa ra những đề xuất phù hợp và cá nhân hóa cho từng người dùng.

Tuy nhiên, việc kết hợp khám phá và khai thác cũng không dễ dàng, đòi hỏi sự cân nhắc cẩn thận và sự linh hoạt trong việc áp dụng các thuật toán và phương pháp thích hợp. Đồng thời, sự cân bằng giữa việc tìm hiểu người dùng và tận dụng thông tin đã có hay nói cách khác là sự cân bằng giữa việc khám phá và khai thác sẽ đóng vai trò quan trọng trong việc đảm bảo tính cá nhân hoá và chính xác của đề xuất sản phẩm.

1.3 Phương pháp giải quyết bài toán mà chúng em tìm hiểu

Để xây dựng các hệ thống đề xuất sản phẩm, chúng ta có thể sử dụng các phương pháp truyền thống như Content-based Filtering [3] (lọc dựa trên nội dung) và Collaborative Filtering [4] (lọc dựa trên sự cộng tác). Tuy nhiên, các phương pháp truyền thống này gặp phải một số khó khăn nhất định.

Phương pháp Content-based Filtering tập trung vào việc đề xuất các sản phẩm dựa trên thông tin hiện có của người dùng. Nó sẽ phân tích và đánh giá các thuộc tính và đặc điểm của sản phẩm, sau đó so sánh với sở thích của người dùng để đưa ra các đề xuất phù hợp. Tuy nhiên, vấn đề xảy ra khi thông tin về người dùng không đủ hoặc không chính xác. Nếu

không có đủ thông tin về người dùng, phương pháp này sẽ gặp khó khăn trong việc đưa ra các đề xuất chính xác và phù hợp với nhu cầu cá nhân.

Ngược lại, Collaborative Filtering dựa trên thông tin từ người dùng khác để đề xuất sản phẩm. Phương pháp này tìm kiếm các người dùng có sở thích tương tự và dựa trên hành vi hoặc đánh giá của họ để đưa ra các đề xuất cho người dùng hiện tại. Tuy nhiên, trong trường hợp người dùng mới hoặc người dùng có sở thích thay đổi, phương pháp Collaborative Filtering sẽ gặp khó khăn vì không có đủ thông tin từ người dùng để tạo ra các đề xuất chính xác.

Để giải quyết vấn đề về thiếu thông tin người dùng hoặc thông tin của người dùng không đủ để hệ thống có thể hiểu và đề xuất sản phẩm, trong những năm gần đây, phương pháp học tăng cường (Reinforcement Learning) đã trở thành một lựa chọn phổ biến để giải quyết bài toán đề xuất sản phẩm. Khác với các phương pháp truyền thống, phương pháp học tăng cường không luôn luôn tập trung chỉ vào khai thác (exploit), mà thay vào đó, nó kết hợp cả khai thác và khám phá (explore). Bằng cách sử dụng cả hai khía cạnh này, phương pháp học tăng cường đạt được sự cân bằng giữa việc hiểu rõ hơn về người dùng và sử dụng thông tin hiện có để đưa ra những đề xuất sản phẩm phù hợp.

Khi áp dụng phương pháp học tăng cường trong việc đề xuất sản phẩm, quá trình bắt đầu với việc hiểu hơn về người dùng. Phương pháp này sẽ đưa ra các đề xuất để tập trung vào việc thu thập thông tin về người dùng, như tương tác, phản hồi, hoặc các thử nghiệm. Bằng cách thực hiện khám phá, phương pháp học tăng cường cố gắng tìm hiểu sâu hơn về sở thích và nhu cầu của người dùng.

Sau khi thu thập đủ thông tin và hiểu rõ hơn về người dùng, phương pháp học tăng cường sẽ cập nhật chiến lược đề xuất sản phẩm để tối ưu hóa đề xuất cho người dùng. Quá trình này thường được thực hiện bằng cách sử dụng các thuật toán và kỹ thuật học tăng cường để tìm ra chiến lược tối ưu nhất dựa trên thông tin hiện có về người dùng. Điều này đảm bảo rằng phương pháp học tăng cường cũng thực hiện khai thác thông tin

một cách hiệu quả, từ việc tìm hiểu người dùng đến việc đề xuất sản phẩm phù hợp.

Bằng việc kết hợp khám phá và khai thác thông tin như vậy, phương pháp học tăng cường giúp giải quyết vấn đề khi thông tin về người dùng không đủ. Đối với người dùng mới, nơi thông tin cung cấp có hạn, hoặc người dùng cũ có sở thích thay đổi theo thời gian, phương pháp này có khả năng cung cấp đề xuất sản phẩm cá nhân hóa và chính xác.

Ngoài ra, phương pháp học sâu cũng có thể được kết hợp với học tăng cường (Deep Reinforcement Learning) [5] để xây dựng hệ thống đề xuất. Phương pháp này vượt trội hơn với phương pháp học tăng cường thông thường ở khả năng xử lý dữ liệu phức tạp, tính tự động hoá cao và hiệu suất tốt hơn. Tuy nhiên, nhược điểm của phương pháp này là đòi hỏi nhiều dữ liệu, tài nguyên tính toán cao hơn so với phương pháp học tăng cường thông thường.

Mỗi phương pháp đều có ưu và nhược điểm riêng. Tuy nhiên, trong giới hạn về thời gian, kiến thức và nguồn tài nguyên, chúng em đã quyết định chọn phương pháp học tăng cường, được ứng dụng thành công trong bài báo “A Contextual-Bandit Approach to Personalized News Article Recommendation” [1]. Cụ thể hơn là giải thuật LinUCB đã được trình bày trong bài báo.

Cách tiếp cận của giải thuật LinUCB là tối ưu hóa việc lựa chọn đề xuất sản phẩm dựa trên việc đánh giá và tính toán giá trị của từng hành động lựa chọn bài tin tức. Thông qua việc sử dụng mô hình tuyến tính giải thuật có thể ước lượng và dự đoán mức độ phù hợp của mỗi bài tin tức với người dùng dựa trên thông tin ngữ cảnh. Điều này cho phép hệ thống cá nhân hoá đề xuất sản phẩm theo từng trường hợp cụ thể và đáp ứng nhu cầu đa dạng của người dùng.

1.4 Bố cục của khoá luận

Đối với các phần còn lại của khoá luận chúng em sẽ trình bày như sau:

- Chương 2: Trình bày những kiến thức nền tảng về phương pháp “Bandit” và các giải thuật trong “Bandit”.
- Chương 3: Trình bày giải thuật LinUCB và hai phiên bản của giải thuật:
 - Phiên bản đơn giản: LinUCB với các mô hình tuyến tính riêng biệt (LinUCB Disjoint).
 - Phiên bản phức tạp: LinUCB với các mô hình tuyến tính có sự chia sẻ (LinUCB Hybrid).
- Chương 4: Trình bày các thí nghiệm để đánh giá hiệu suất của giải thuật LinUCB.
- Chương 5: Tổng kết các hướng phát triển thêm của đề tài.

Chương 2

Kiến thức nền tảng

Trong chương này, nhóm chúng em sẽ trình bày những kiến thức nền tảng để giúp hiểu chi tiết và sâu hơn về giải thuật chính trong khóa luận. Đầu tiên, chúng em sẽ trình bày về “Bandit” - phương pháp được áp dụng cho hệ thống đề xuất sản phẩm trong khóa luận. Sau đó, chúng em sẽ trình bày hai giải thuật trong “Bandit” là “ ϵ -Greedy” và “UCB” - hai giải thuật được áp dụng để giải quyết bài toán đề xuất sản phẩm. Đặc biệt là về phần giải thuật “UCB” sẽ cung cấp nền tảng để hiểu rõ hơn về những cải tiến của giải thuật chính được trình bày ở chương kế tiếp.

2.1 Phương pháp Bandit

Bandit là một phương pháp nằm trong phương pháp học tăng cường. Phương pháp này được áp dụng trong môi trường không chắc chắn. Trong môi trường này các thông tin không được biết trước, nghĩa là các kết quả và các sự kiện có thể xảy ra ngẫu nhiên và không thể dự đoán chính xác trước đó. Để hiểu rõ hơn, chúng em sẽ lấy ví dụ về môi trường không chắc chắn:

Giả sử chúng ta có một hệ thống gợi ý tin tức cá nhân, nơi người dùng có thể tạo tài khoản và chọn các chủ đề quan tâm. Hệ thống này sẽ tự động gợi ý tin tức cho người dùng dựa trên sở thích và lịch

sử dụng tin của họ. Tuy nhiên, trong môi trường không chắc chắn, sở thích của người dùng có thể thay đổi theo thời gian hoặc họ có thể quan tâm đến các chủ đề mới hoặc cũng có thể hệ thống chưa có những thông tin liên quan đến người dùng. Ngoài ra, môi trường tin tức thường thay đổi liên tục với sự xuất hiện của các tin tức mới và xu hướng thị trường.

Phương pháp bandit được lấy ý tưởng từ các máy đánh bạc (slot machine). Mỗi máy đánh bạc có một cần gạt và khi chúng ta kéo cần gạt này, máy đánh bạc sẽ trả về cho người chơi một phần thưởng nào đó. Trong một nhóm các máy đánh bạc, tại mỗi thời điểm, người chơi sẽ chọn một máy đánh bạc để kéo cần gạt và nhận về phần thưởng, sau nhiều lần kéo, người chơi phải tìm ra được máy đánh bạc tối ưu nhất tại mỗi thời điểm, để đạt được nhiều phần thưởng nhất có thể.



Hình 2.1: Minh họa máy đánh bạc (slot machine)

Hình 2.1 minh họa một nhóm các máy đánh bạc, khi kéo cần gạt, máy đánh bạc sẽ trả về hình ảnh chữ số, trái cây, ... và sẽ có những quy tắc để quy đổi những hình ảnh này thành phần thưởng. Mỗi máy đánh bạc sẽ có

một phân phối xác suất khác nhau cho kết quả nó trả về.

Trong phương pháp bandit, khi chỉ chơi trên một máy đánh bạc duy nhất, chúng ta gọi nó là “one-armed bandit”. Tuy nhiên, nếu chơi trên một nhóm máy đánh bạc, chúng ta gọi đó là “multi-armed bandit” hoặc “K-armed bandit” nếu có K máy đánh bạc trong nhóm. Tương tự việc lựa chọn máy đánh bạc để kéo cần gạt, phương pháp bandit thử nghiệm các hành động khác nhau và dựa vào kết quả thu được để cập nhật ước lượng về giá trị của từng hành động. Theo thời gian, phương pháp bandit sẽ tự động lựa chọn các hành động mang lại lợi ích cao nhất dựa trên việc thử nghiệm và khám phá quá trình tương tác với môi trường.

Trong phương pháp bandit, sự cân bằng giữa khám phá (explore) và khai thác (exploit) là rất quan trọng. Khám phá đòi hỏi thử nghiệm các hành động chưa được biết để thu thập thông tin và kiến thức về môi trường. Tương tự, như người chơi thử kéo cần gạt từng máy để tìm hiểu phần thưởng của từng máy đánh bạc. Trái lại, khai thác tập trung vào việc chọn các hành động tiềm năng đã biết để đạt được lợi ích tối đa. Nó giống như việc người chơi đã biết phần thưởng của mỗi máy đánh bạc và chỉ cần chọn máy có phần thưởng cao nhất để tối đa hóa lợi ích cá nhân từ việc chơi máy đánh bạc.

Trong trò chơi máy đánh bạc, mỗi hành động đều mất một khoản chi phí cố định. Do đó, chúng ta cần phải có chiến lược khám phá và khai thác phù hợp để đạt được tổng điểm thưởng lớn nhất với mức chi phí thấp nhất. Việc này đòi hỏi sự cân trọng trong việc đánh giá rủi ro và phần thưởng của từng hành động. Điều quan trọng là phải tìm ra một sự cân bằng hợp lý giữa việc khám phá và khai thác, để không mắc phải tình trạng “khám phá quá mức” dẫn đến mất lợi thế của thông tin đã biết và cũng không rơi vào “khai thác quá mức” dẫn đến bỏ lỡ cơ hội khám phá những hành động tiềm năng mới. Bằng cách thực hiện các chiến lược hợp

lí, phương pháp bandit sẽ giúp chúng ta đạt được sự cân bằng tối ưu và tối đa hóa lợi ích trong trò chơi máy đánh bạc.

Trong khóa luận, phương pháp bandit được áp dụng để giải quyết bài toán đề xuất tin tức. Bằng cách áp dụng phương pháp bandit, hệ thống có khả năng tự động thử nghiệm và đánh giá hiệu quả của các tin tức khác nhau, từ đó tìm ra cách đề xuất tin tức mang lại sự trải nghiệm tốt nhất cho người dùng.

Một số khái niệm quan trọng trong phương pháp bandit:

- **Hành động (action):** là sự lựa chọn mà người dùng hoặc hệ thống thực hiện để tương tác với môi trường trong một tình huống cụ thể. Trong phương pháp bandit, mỗi hành động đại diện cho một lựa chọn tiềm năng có thể được thực hiện. Ví dụ, trong bài toán đề xuất tin tức, hành động chính là việc lựa chọn một bài tin tức để đề xuất cho người dùng. Chúng ta ký hiệu hành động tại thời điểm t là A_t .
- **Phần thưởng (reward):** là giá trị hay lợi ích nhận được sau khi thực hiện một hành động. Trong đề xuất tin tức, phần thưởng sẽ là 1 hoặc 0 (tương ứng với người dùng click hay không click vào bài tin tức). Mỗi hành động sẽ có một phần thưởng riêng, được lấy ngẫu nhiên từ một phân phối xác suất dành riêng cho hành động đó. Những phần thưởng này là cơ sở cho việc học và tìm kiếm những hành động tốt nhất. Mục tiêu là tối ưu hóa tổng phần thưởng tích lũy thông qua việc liên tục đánh giá và cập nhật giá trị của từng hành động. Phần thưởng nhận được tại thời điểm t , sau khi thực hiện hành động A_t được ký hiệu là r_t .
- **Giá trị điểm thưởng trung bình hay còn được gọi là giá trị ước lượng của hành động:** là giá trị trung bình của phần thưởng thu được từ một hành động được thực hiện trong quá trình thử nghiệm hoặc được ước lượng từ mô hình tuyến tính. Giá trị điểm thưởng trung bình thường được sử dụng để đánh giá và so sánh các hành động trong

phương pháp bandit. Ký hiệu của giá trị điểm thưởng trung bình của hành động a tại thời điểm t với n lần hành động a đã được thực hiện là $Q_t(a)$

$$Q_t(a) = \frac{1}{n} \sum_{i=1}^n r_i \quad (2.1)$$

- Độ hối tiếc (regret): Là một đại lượng đo lường hiệu suất của phương pháp bandit, đại diện cho sự mất mát khi không chọn hành động tốt nhất. Regret được tính bằng hiệu của tổng phần thưởng tối đa có thể đạt được nếu luôn chọn hành động tốt nhất và tổng phần thưởng thực tế thu được. Độ hối tiếc của giải thuật A sau T lần thực hiện các hành động được biểu diễn bằng công thức:

$$R_A(T) = E \left[\sum_{i=1}^T r_{t, a_t^*} \right] - E \left[\sum_{i=1}^T r_{t, a_t} \right] \quad (2.2)$$

Trong đó a_t^* và a_t lần lượt là hành động có điểm thưởng cao nhất và hành động được lựa chọn tại thời điểm t , E là giá trị kỳ vọng.

Để hiểu rõ hơn về phương pháp bandit, chúng em xin trình bày quy trình thực hiện của phương pháp bandit:

1. Khởi tạo: khởi tạo các tham số và ước lượng ban đầu cho các hành động. Các giá trị ban đầu này có thể được đặt ngẫu nhiên hoặc được xác định dựa trên kiến thức sẵn có về môi trường.
2. Lựa chọn hành động: trong mỗi bước thời gian, phương pháp bandit chọn một hành động để thực hiện. Quyết định này có thể dựa trên nguyên tắc khám phá và khai thác của giải thuật.
3. Thực hiện hành động: sau khi chọn hành động, phương pháp bandit thực hiện hành động đó trong môi trường thực tế.

4. Nhận phần thưởng: sau khi thực hiện hành động, phương pháp bandit nhận được một phần thưởng (reward) từ môi trường.
5. Cập nhật: sau khi nhận được phần thưởng, phương pháp bandit cập nhật các tham số và ước lượng về giá trị của hành động đã thực hiện.
6. Lặp lại quá trình: các bước từ 2 đến 5 được lặp lại cho đến khi đạt đến điều kiện dừng, chẳng hạn như đạt đủ số lần thử nghiệm hoặc đạt được hiệu suất mong muốn.

Trong phần trình bày trên, chúng em đã giới thiệu quy trình cơ bản của phương pháp bandit. Tuy nhiên, để đạt được hiệu suất tối đa, chúng ta cần tối ưu hóa các siêu tham số của phương pháp bandit. Điều này có thể được thực hiện bằng cách so sánh hiệu suất của cùng một giải thuật, nhưng với các giá trị siêu tham số khác nhau. Bên cạnh đó, chúng ta cần so sánh thêm về hiệu suất với các giải thuật khác, bằng cách thực hiện các thử nghiệm và đánh giá kết quả. Từ đó, chúng ta có thể xác định giải thuật và siêu tham số phù hợp nhất đáp ứng được yêu cầu của bài toán đang được giải quyết.

Phương pháp bandit có nhiều ứng dụng thực tế trong nhiều lĩnh vực khác nhau. Sau đây sẽ là một vài ví dụ điển hình của phương pháp bandit.

- Đề xuất nội dung và quảng cáo trực tuyến: Trong lĩnh vực này, phương pháp bandit được sử dụng để lựa chọn quảng cáo hay nội dung (phim, bài hát, podcast, ...) phù hợp cho từng người dùng. Từ đó, tăng tỉ lệ click cũng như hiệu suất quảng cáo và đề xuất nội dung.
- Chơi game: Trong lĩnh vực chơi game, phương pháp bandit có thể được sử dụng để tối ưu hóa việc chọn hành động trong trò chơi và tìm ra chiến lược tối ưu để đạt được kết quả tốt nhất.

- Quản lý tài nguyên: Trong quản lý tài nguyên, phương pháp bandit có thể được áp dụng để tối ưu hóa việc chọn đối tác hoặc nhà cung cấp dịch vụ dựa trên hiệu suất và lợi ích mà từng đối tác mang lại.

Ngoài các ví dụ trên, trong thực tế phương pháp bandit còn được áp dụng cho rất nhiều lĩnh vực như y tế, dịch thuật, kiểm tra và đánh giá, quản lý tài chính, ...

2.2 Các giải thuật trong Bandit

2.2.1 Giải thuật ϵ -Greedy

Khi áp dụng phương pháp bandit trong bài toán đề xuất tin tức, chúng ta nhận được điểm thưởng khi đề xuất một bài tin tức cho người dùng. Điểm thưởng này giúp chúng ta ước tính mức độ phù hợp của bài tin tức với người dùng. Để tối ưu hóa việc đề xuất tin tức, chúng ta có thể áp dụng phương pháp lựa chọn tham lam, còn được gọi là giải thuật Greedy. Phương pháp này dựa trên việc lựa chọn bài tin tức có ước tính giá trị cao nhất để đề xuất cho người dùng.

Việc lựa chọn hành động có giá trị cao nhất tại thời điểm t , có thể được biểu diễn bằng công thức:

$$A_t = \arg \max_a Q_t(a)$$

Trong đó: $Q_t(a)$ là giá trị điểm thưởng trung bình của hành động a tại thời điểm t và $\arg \max$ chỉ định cho việc lựa chọn hành động a có $Q_t(a)$ là lớn nhất.

Giải thuật Greedy đơn giản, dễ hiểu và dễ triển khai. Nó không yêu cầu tính toán phức tạp và không cần lưu trữ lịch sử, giúp tiết kiệm tài nguyên tính toán và thời gian. Khi môi trường ổn định và việc khám phá

không quan trọng, Greedy có thể nhanh chóng hội tụ đến giải pháp tối ưu. Điều này làm cho nó rất hữu ích trong nhiều bài toán như quảng cáo trực tuyến, đề xuất nội dung, ... Giải thuật Greedy mang lại hiệu quả cao và có thể áp dụng rộng rãi để giải quyết các vấn đề thực tế.

Lựa chọn hành động tối ưu mang lại nhiều lợi ích trong quá trình thử nghiệm. Tuy nhiên, giải thuật Greedy có một điểm yếu khá lớn là thiếu khả năng khám phá. Việc chỉ tập trung vào việc lựa chọn hành động tốt nhất dựa trên thông tin hiện có trong hệ thống có thể dẫn đến bỏ qua những lựa chọn có tiềm năng phần thưởng cao hơn trong tương lai. Nghiêm trọng hơn, nếu một hành động có giá trị ban đầu cao, Greedy có thể liên tục chọn nó mà bỏ qua các hành động khác có phần thưởng lớn hơn.

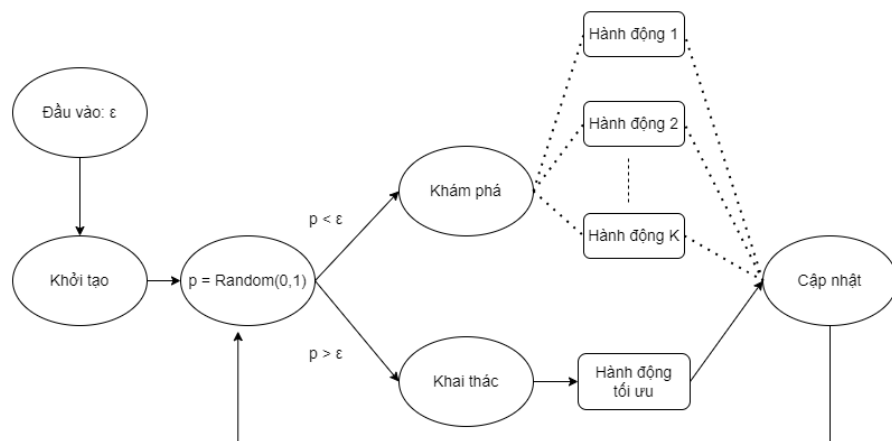
Hơn nữa, trong môi trường động có sự thay đổi theo thời gian, Greedy chỉ tập trung vào việc khai thác mà không khám phá, dẫn đến khả năng điều chỉnh chậm và không tìm ra các lựa chọn có phần thưởng cao hơn. Hiệu suất của giải thuật Greedy cũng phụ thuộc nhiều vào giá trị khởi tạo ban đầu cho từng hành động. Nếu giá trị khởi tạo ban đầu không tốt hoặc không tối ưu, Greedy có thể không đạt hiệu suất tốt nhất.

Như đã được trình bày ở trên, giải thuật Greedy mang theo một rủi ro đáng kể khi chỉ tập trung vào việc lựa chọn một hành động tối ưu và liên tục lựa chọn hành động này do nó có giá trị phần thưởng cao nhất. Hậu quả của điều này là các hành động tối ưu khác có thể bị bỏ qua. Để khắc phục vấn đề này, một cách đơn giản là bổ sung yếu tố khám phá vào giải thuật Greedy. Và ϵ -Greedy chính là giải thuật có được điểm cải tiến này trong Greedy.

Về cơ bản, giải thuật ϵ -Greedy là một phương pháp lựa chọn hành động theo cách tham lam. Tức là, hành động được lựa chọn là hành động có giá trị phần thưởng ước tính cao nhất. Tuy nhiên, ở mỗi bước thời gian

t , thay vì lựa chọn tham lam, một hành động có thể được chọn một cách ngẫu nhiên từ tập hợp các hành động. Xác suất để lựa chọn ngẫu nhiên một hành động được xác định bởi giá trị của tham số ϵ .

Bằng cách này, giải thuật ϵ -Greedy kết hợp cả hai yếu tố quan trọng là khai thác và khám phá. Trong quá trình thực hiện, mọi hành động sẽ được thực hiện để thu thập thông tin về giá trị phần thưởng thực sự của chúng, từ đó cung cấp các ước tính ngày càng chính xác hơn. Mục tiêu của giải thuật ϵ -Greedy là tối ưu hóa phần thưởng tích lũy trong quá trình thử nghiệm, đồng thời vẫn duy trì khả năng khám phá các hành động mới. Theo thời gian, giải thuật ϵ -Greedy tăng dần mức độ chắc chắn và tin cậy của thông tin thu thập được, từ đó cung cấp quyết định tối ưu hơn.



Hình 2.2: Quy trình thực hiện của giải thuật ϵ -Greedy

Hình 2.2 mô tả quy trình thực hiện của giải thuật ϵ -Greedy:

- Đầu vào: $\epsilon \in [0, 1]$, xác suất cho việc lựa chọn một hành động ngẫu nhiên.
1. Khởi tạo: Đầu tiên, mỗi hành động sẽ được khởi tạo giá trị điểm thưởng trung bình (thường là 0) và số lần được lựa chọn là 0.

2. Chọn ngẫu nhiên theo phân phối đều giá trị xác suất $p \in [0,1]$, giá trị p quyết định việc lựa chọn tham lam hay lựa chọn ngẫu nhiên.
3. Lựa chọn hành động:
 - Với $p < \epsilon$, giải thuật sẽ thực hiện việc khám phá bằng cách lựa chọn ngẫu nhiên một hành động trong tập hợp các hành động.
 - Với $p \geq \epsilon$, giải thuật sẽ thực hiện việc khai thác bằng cách lựa chọn hành động có giá trị trung bình điểm thưởng cao nhất.
4. Cập nhật: Sau khi nhận được điểm thưởng từ hành động đã thực hiện, giải thuật sẽ cập nhật lại giá trị điểm thưởng trung bình và số lần được lựa chọn cho hành động đã thực hiện.
5. Quay lại bước 2, tiếp tục vòng lặp mới
 - Giải thuật ϵ -Greedy sẽ kết thúc khi đạt đến điều kiện dừng, chẳng hạn như đạt đủ số lần thử nghiệm hoặc đạt ngưỡng hiệu suất đã đặt ra

Trong giải thuật ϵ -greedy, siêu tham số ϵ đóng vai trò quan trọng và có ảnh hưởng đáng kể đến hiệu suất của giải thuật. ϵ xác định tỷ lệ giữa khai thác và khám phá trong quá trình chọn hành động. Giá trị của ϵ ảnh hưởng đến hiệu suất của giải thuật có thể được diễn giải như sau:

- Khi ϵ bằng 0, giải thuật trở thành Greedy, trong đó hành động được lựa chọn dựa trên chiến lược tham lam. Điều này đồng nghĩa với việc không có sự khám phá và không thể xác định được hành động tối ưu trong quá trình thử nghiệm. Greedy tập trung vào việc khai thác những hành động có giá trị cao nhất dựa trên thông tin hiện có. Tuy nhiên, điều này có thể dẫn đến việc bỏ qua các hành động tối ưu khác vì không có sự khám phá.

- Khi ϵ tăng lên, tỷ lệ lựa chọn ngẫu nhiên các hành động cũng tăng. Điều này đồng nghĩa với việc khám phá cũng được thực hiện nhiều hơn. Giải thuật ϵ -greedy giúp xác định các hành động tối ưu và lựa chọn chúng thường xuyên hơn. Đồng thời, các hành động không tối ưu sẽ ít được lựa chọn.
- Khi ϵ tiến gần đến giá trị 1, hầu như việc khám phá luôn xảy ra. Tuy nhiên, cần lưu ý rằng việc khám phá quá mức có thể dẫn đến lãng phí thời gian và tài nguyên trong việc khám phá các hành động không quan trọng. Đồng thời, giảm thiểu khai thác có thể ảnh hưởng đáng kể đến hiệu suất của giải thuật.

Vì vậy, để đạt được hiệu suất tốt nhất, giá trị của siêu tham số ϵ cần được điều chỉnh phù hợp dựa trên mục tiêu của bài toán và đặc điểm của hệ thống. Điều này đảm bảo rằng cân bằng được giữa việc khai thác và khám phá để đạt được hiệu suất tốt nhất.

Trong phần phương pháp bandit, độ hối tiếc (regret) là hiệu của tổng phần thưởng tối đa nhận được và tổng phần thưởng nhận được trong thực tế. Bằng cách giảm thiểu giá trị regret, chúng ta có thể tối đa hóa tổng phần thưởng tích lũy. Trong giải thuật ϵ -Greedy, sau một vài bước thời gian đầu tiên, tổng phần thưởng tích lũy nhận được sẽ tăng tuyến tính; tương đương với việc thông tin về các hành động tối ưu trong toàn bộ tập hành động đã được khám phá. Tuy nhiên, do việc khám phá ngẫu nhiên vẫn diễn ra, dẫn đến sự gia tăng tuyến tính của độ hối tiếc theo thời gian. Tỷ lệ tăng của độ hối tiếc xảy ra trong giai đoạn đầu của quá trình thử nghiệm cao hơn một chút là do chưa xác định được hành động tốt nhất. Vì độ hối tiếc có sự gia tăng tuyến tính theo thời gian, nên giải thuật ϵ -Greedy vẫn còn một số hạn chế. Do đó, cần cân nhắc và áp dụng các phương pháp khác nhau, hoặc sử dụng các biến thể khác của giải thuật, để đạt được hiệu suất tối ưu trong các bài toán thực tế.

2.2.2 Giải thuật UCB

Giải thuật UCB (Upper Confidence Bound) là một phương pháp hiệu quả trong việc giải quyết bài toán đề xuất sản phẩm, cụ thể trong khóa luận lần này chính là đề xuất tin tức cho người dùng. Giải thuật UCB kết hợp giữa sự khám phá và khai thác, giúp chúng ta tìm ra những tin tức hấp dẫn và thú vị nhất, từ đó mang lại trải nghiệm chất lượng và thú vị hơn cho người dùng.

Giải thuật UCB là một phương pháp đáng chú ý trong bài toán “multi-armed bandit” và lĩnh vực đề xuất tin tức. Với mục tiêu tối ưu hóa quá trình đề xuất tin tức, UCB sử dụng một chiến lược thông minh để cân bằng giữa việc khám phá và khai thác. Trong giải thuật UCB, những tin tức có tiềm năng hấp dẫn được đánh giá cao dựa trên dữ liệu thu thập được từ quá trình trước đó. Điều này đảm bảo rằng chúng ta tập trung vào những tin tức có khả năng mang lại kết quả tốt. Tuy nhiên, UCB cũng không bỏ qua khả năng khám phá những tin tức mới và chưa biết trước. Việc này đảm bảo rằng chúng ta không bị giới hạn bởi những bài tin tức đã biết và có thể khám phá những bài tin tức tiềm năng khác. Kết quả của việc áp dụng UCB là người dùng sẽ nhận được những đề xuất tin tức phong phú và đa dạng với nội dung thú vị và có giá trị cao. Điều này tạo ra một trải nghiệm tốt hơn và đáp ứng được sở thích đa dạng của người dùng.

Giải thuật UCB cung cấp một phương pháp hiệu quả để cân bằng giữa khai thác thông tin hiện có và khám phá thông tin mới. Thay vì chỉ đơn thuần chọn hành động ngẫu nhiên để khám phá, UCB sử dụng một cơ chế đặc biệt để đảm bảo sự cân bằng này. UCB đánh giá giá trị điểm thưởng trung bình và khoảng tin cậy của mỗi hành động dựa trên số lần đã chọn hành động đó và thông tin thu thập được từ các lượt lựa chọn trước đó. Việc này giúp UCB đưa ra quyết định thông minh khi cần khám phá và khai thác. Để khám phá, UCB sẽ ưu tiên lựa chọn những hành động có độ tin cậy thấp (khoảng tin cậy cao) hơn, để tìm hiểu thêm thông tin về

chúng. Đồng thời, để khai thác, UCB sẽ chọn những hành động có giá trị điểm thưởng trung bình cao nhất dựa trên ước lượng tốt nhất hiện tại.

Trong quá trình đề xuất tin tức, cách hoạt động của giải thuật UCB là sử dụng công thức tính toán giá trị “ucb” cho mỗi bài tin tức, từ đó chọn ra bài tin tức có giá trị “ucb” lớn nhất để đề xuất cho người dùng. Quá trình này được lặp lại sau mỗi lượt chọn. Thông tin về số lần chọn và giá trị điểm thưởng trung bình của mỗi bài tin tức được cập nhật để cung cấp thông tin mới nhất cho quyết định đề xuất tiếp theo.

Trong giải thuật UCB, A_t là bài tin tức được lựa chọn ở bước thời gian t để đề xuất cho người dùng được chọn bởi công thức sau:

$$ucb = Q_t(a) + \alpha \sqrt{\frac{\log(t)}{N_t(a)}} \quad (2.3)$$

$$A_t = \arg \max_a [ucb] \quad (2.4)$$

Trong đó:

- $Q_t(a)$ là giá trị điểm thưởng trung bình của hành động ‘a’ tại bước thời gian ‘t’.
- $N_t(a)$ số lần hành động ‘a’ được chọn trước thời điểm ‘t’.
- α là giá trị tin cậy, kiểm soát mức độ khám phá.
- $\arg \max$ chỉ định cho việc lựa chọn hành động ‘a’ có giá trị “ucb” lớn nhất.

Trong giải thuật UCB, giá trị “ucb” được tính toán bởi 2 thành phần riêng biệt:

1. Khai thác:

- $Q_t(a)$, đại diện cho phần khai thác của giải thuật UCB. Về cơ bản nếu chỉ lấy nửa phương trình này thì hành động được lựa chọn sẽ là hành động có giá trị điểm thưởng trung bình cao nhất.

2. Khám phá:

- Nửa sau của phương trình bổ sung tính năng khám phá, với mức độ khám phá được kiểm soát bởi siêu tham số α . Trên thực tế, phần này của phương trình cung cấp thước đo độ tin cậy cho việc ước tính giá trị của hành động.
- Nếu một hành động không được lựa chọn thường xuyên hoặc không được lựa chọn, thì $N_t(a)$ sẽ nhỏ. Do đó, giá trị khoảng tin cậy chắc chắn sẽ lớn, làm cho hành động này có nhiều khả năng được chọn hơn. Mỗi khi một hành động được thực hiện, chúng ta sẽ tự tin hơn về ước tính của hành động đó. Trong trường hợp này, $N_t(a)$ tăng lên, và do đó, giá trị khoảng tin cậy chắc chắn giảm, khiến hành động này ít có khả năng được chọn bởi việc khám phá, dù vậy nó vẫn có thể được chọn bởi việc khai thác.
- Khi một hành động không được chọn, giá trị khoảng tin cậy sẽ tăng chậm, do hàm $\log(t)$ trong tử số. Trong khi đó, mỗi khi hành động đó được chọn, giá trị khoảng tin cậy sẽ giảm nhanh chóng do sự gia tăng của $N_t(a)$ là tuyến tính. Vì vậy, việc khám phá sẽ ưu tiên đối với các hành động không được chọn thường xuyên, do giá trị ước tính của hành động đó có độ tin cậy thấp (giá trị khoảng tin cậy cao).
- Khi thời gian trôi qua, việc khám phá sẽ giảm dần (vì giá trị ' N ' sẽ tăng đến vô cùng, tương đương với việc khám phá sẽ hầu như là không, lúc này việc lựa chọn các hành động chỉ dựa vào phần khai thác).

Trong bài toán đề xuất tin tức cho người dùng, quá trình giải thuật UCB thực hiện được trình bày theo giải thuật 1:

Algorithm 1 Giải thuật UCB

```

1: Input:  $\alpha \in [0, +\infty)$ 
2: for  $t = 1$  to  $T$  do
3:   for all  $a \in \mathcal{A}_t$  do
4:     if  $a$  là bài tin tức mới then
5:        $\mathbf{Q}_a \leftarrow 0$  (Khởi tạo giá trị trung bình điểm thưởng cho bài
6:         tin tức)
7:        $\mathbf{N}_a \leftarrow 0$  (Khởi tạo số lần bài tin tức được lựa chọn để đề
8:         xuất cho người dùng)
9:     end if
10:    Tính giá trị UCB:  $ucb = Q_t(a) + \alpha \sqrt{\frac{\log(t)}{N_t(a)}}$ 
11:  end for
12:  Chọn bài tin tức:  $a_t = \arg \max_{a \in A} ucb_{a,t}$ 
13:  Nhận điểm thưởng:  $r_t$ 
14:  Cập nhật tham số:
15:     $N_{a_t} \leftarrow N_{a_t} + 1$ 
16:     $Q_{a_t} \leftarrow Q_{a_t} + \frac{r_t - Q_{a_t}}{N_{a_t}}$ 
17: end for

```

Quy trình đề xuất tin tức bằng giải thuật UCB bao gồm các bước sau:

- Đầu vào: Siêu tham số $\alpha \in [0, +\infty)$ giúp cân bằng giữa việc khám phá và khai thác bằng cách kiểm soát mức độ của việc khám phá.
1. Khởi tạo: Đầu tiên, mỗi bài tin tức sẽ được khởi tạo giá trị điểm thưởng trung bình (thường là 0) và số lần được lựa chọn là 0
 2. Tính toán giá trị ucb cho mỗi bài tin tức theo công thức 2.3.
 3. Lựa chọn bài tin tức có giá trị “ucb” lớn nhất để đề xuất cho người dùng.

4. Nhận điểm thưởng sau khi đề xuất tin tức.
 5. Cập nhật: Cập nhật lại giá trị điểm thưởng trung bình và số lần được lựa chọn cho bài tin tức đã được đề xuất cho người dùng.
 6. Quay lại bước 2, tiếp tục vòng lặp mới.
- Điều kiện dừng của giải thuật UCB khi đạt đủ số lần thử nghiệm hoặc đạt ngưỡng hiệu suất đã đặt ra.

Trong giải thuật UCB, tồn tại một siêu tham số quan trọng, được gọi là α . Giá trị của α đóng vai trò quan trọng trong việc điều chỉnh mức độ khám phá và khai thác của giải thuật. α ảnh hưởng trực tiếp đến cách giải thuật tính toán giá trị “ucb” cho từng tin tức. Siêu tham số α trong UCB được sử dụng để xác định mức độ tin cậy (confidence level) mà chúng ta mong muốn đạt được trong việc ước lượng giá trị “ucb”. Nó xác định phạm vi của khoảng tin cậy cho giá trị điểm thưởng trung bình của mỗi bài tin tức. Khi α tăng lên, phạm vi của khoảng tin cậy mở rộng, cho phép chúng ta khám phá nhiều hơn những bài tin tức có ít thông tin hơn.

Bằng cách điều chỉnh giá trị của α một cách phù hợp, chúng ta có thể điều tiết sự cân bằng giữa việc khám phá thông tin mới và khai thác thông tin hiện có. Qua đó, giải thuật UCB cho phép chúng ta tận dụng tối đa thông tin có sẵn và đồng thời mở rộng phạm vi để khám phá những khía cạnh mới và không rõ ràng trong quá trình đề xuất tin tức.

Tuy nhiên, việc chọn giá trị α thích hợp là một thách thức. Nếu giá trị α quá cao, sẽ có mức độ khám phá cao nhưng cũng có nguy cơ chọn những tin tức kém chất lượng. Ngược lại, nếu giá trị α quá thấp, giải thuật sẽ dễ rơi vào khai thác và bỏ qua những tin tức tiềm năng. Thông thường, giá trị α được chọn để đáp ứng yêu cầu về đặc điểm của hệ thống và mục tiêu đề xuất tin tức. Có thể thử nghiệm và điều chỉnh giá trị α trong quá trình huấn luyện và thử nghiệm để tìm ra giá trị tối ưu cho hệ thống cụ thể.

Giải thuật UCB có ưu điểm là nhanh chóng xác định hành động tối ưu và chỉ thử các hành động khác khi chúng có khoảng tin cậy cao. Điều này

dẫn đến việc giải thuật UCB có độ hối tiếc (regret) thấp hơn nhiều so với giải thuật ϵ -Greedy. Đa số giá trị regret cao xảy ra trong các vòng lặp đầu, khi mỗi bài tin tức được lựa chọn lần đầu để có ước tính về giá trị điểm thưởng trung bình ban đầu.

Mức độ gia tăng của regret trong giải thuật UCB tuân theo hàm $\log(T)$, trong đó T là tổng số bước thời gian. Điều này có nghĩa là với sự tăng dần của T , tốc độ gia tăng của regret cũng tăng theo tỷ lệ logarithmic. Điều này cho thấy giải thuật UCB có khả năng giảm thiểu regret hiệu quả và tạo ra kết quả tốt hơn theo thời gian. Từ việc giảm thiểu regret, giải thuật UCB mang lại hiệu quả trong việc tối ưu hóa quá trình đề xuất tin tức, đảm bảo rằng người dùng sẽ nhận được những đề xuất tin tức có giá trị cao và đáng chú ý, mang đến trải nghiệm tốt hơn và giúp nâng cao chất lượng của hệ thống đề xuất.

Chương 3

Giải thuật LinUCB để giải quyết bài toán đề xuất sản phẩm

Ở chương 3, chúng em sẽ trình bày về giải thuật *LinUCB* được đề xuất trong bài báo “A Contextual-Bandit Approach to Personalized News Article Recommendation” [1] để giải quyết bài toán đề xuất sản phẩm. Đầu tiên, chúng em sẽ trình bày ý tưởng sử dụng ngữ cảnh và mô hình tuyến tính của giải thuật *LinUCB* để cá nhân hoá việc đề xuất sản phẩm cho từng người dùng; đây là điểm cải tiến so với giải thuật *UCB* đã được trình bày ở chương 2. Sau đó, chúng em sẽ trình bày cụ thể về hai phiên bản của giải thuật *LinUCB*: phiên bản đơn giản là *LinUCB* với các mô hình tuyến tính riêng biệt (*LinUCB Disjoint*) và phiên bản phức tạp hơn là *LinUCB* với các mô hình tuyến tính có sự chia sẻ (*LinUCB Hybrid*).

3.1 Áp dụng ngữ cảnh và mô hình tuyến tính trong LinUCB

Như đã trình bày ở chương 1, ngữ cảnh đóng vai trò quan trọng trong việc nâng cao khả năng đề xuất. Nếu không có ngữ cảnh thì hệ thống không có khả năng cá nhân hoá đề xuất cho từng người dùng. Áp dụng ngữ cảnh và mô hình tuyến tính trong giải thuật *LinUCB* cho phép khai

thác tối đa thông tin có sẵn về người dùng và sản phẩm để đề xuất những sản phẩm phù hợp nhất.

Qua việc áp dụng ngữ cảnh và mô hình tuyến tính trong LinUCB, hệ thống sẽ đạt được sự cá nhân hoá cao hơn và đem lại trải nghiệm đáng tin cậy cho người dùng. Điều này đại diện cho một bước tiến đáng chú ý so với giải thuật UCB đã được trình bày trong chương trước.

Đầu tiên, hãy xem xét sự khác biệt giữa hai giải thuật LinUCB và UCB. Giải thuật UCB đã được sử dụng rộng rãi trong việc đề xuất sản phẩm dựa trên tương tác người dùng, nhưng nó có một hạn chế quan trọng là không thể cá nhân hoá đề xuất cho từng người dùng cụ thể. UCB dựa vào một công thức tổng quát (công thức 2.3) để tính toán giá trị dự đoán, mà không xem xét đặc điểm riêng của từng người dùng hay ngữ cảnh. Điều này dẫn đến việc đề xuất sản phẩm không phù hợp và giới hạn khả năng tối ưu hóa trải nghiệm người dùng.

Với giải thuật LinUCB, chúng ta có thể khắc phục nhược điểm này bằng cách sử dụng ngữ cảnh và mô hình tuyến tính. Giải thuật LinUCB cải thiện quá trình đề xuất sản phẩm bằng hai cách quan trọng. Thứ nhất, LinUCB giúp cá nhân hoá đề xuất sản phẩm cho từng người dùng cụ thể. Thay vì sử dụng một công thức tổng quát, LinUCB xem xét đặc điểm riêng của từng người dùng và đưa ra những đề xuất phù hợp với sở thích và tương tác trước đó của họ. Điều này giúp cá nhân hoá trải nghiệm và tăng cường sự hài lòng của người dùng. Thứ hai, LinUCB cải thiện quá trình đề xuất sản phẩm bằng cách tận dụng thông tin ngữ cảnh. Ngữ cảnh có thể là các biến như thời gian, vị trí, hoặc trạng thái người dùng. LinUCB sẽ sử dụng thông tin ngữ cảnh này để xác định các yếu tố quan trọng ảnh hưởng đến sự lựa chọn của người dùng và điều chỉnh giá trị dự đoán cho từng sản phẩm. Điều này đặc biệt hữu ích khi thông tin về người dùng thay đổi theo thời gian.

Ví dụ cụ thể với hệ thống đề xuất bài tin tức, giải thuật LinUCB sử dụng ngữ cảnh và mô hình tuyến tính như sau:

1. Ngữ cảnh

- Với mỗi bước thời gian t , giải thuật LinUCB có thể sẽ được áp dụng trên các ngữ cảnh khác nhau. Ngữ cảnh là thông tin về người dùng và bài tin tức.
- Thông tin về ngữ cảnh có thể được sử dụng để điều chỉnh các tham số của giải thuật nhằm cân bằng giữa việc khai thác và khám phá.

2. Mô hình tuyến tính

- Mô hình tuyến tính được áp dụng để tính toán giá trị của từng bài tin tức dựa trên đặc trưng về ngữ cảnh. Cụ thể hơn, mô hình tuyến tính sẽ tính toán các giá trị điểm thưởng trung bình và khoảng tin cậy của từng bài tin tức dựa trên thông tin về người dùng và bài tin tức đó.
- Mô hình tuyến tính được sử dụng để tính toán giá trị ước lượng cho từng bài tin tức dựa trên ngữ cảnh. Từ đó giúp giải thuật chọn được bài tin tức tốt nhất cho người dùng.

Tóm lại, việc áp dụng ngữ cảnh và mô hình tuyến tính trong giải thuật LinUCB là một bước tiến quan trọng so với giải thuật UCB truyền thống. LinUCB cho phép cá nhân hoá đề xuất sản phẩm cho từng người dùng cụ thể và tận dụng thông tin ngữ cảnh để tối ưu hóa quá trình đề xuất. Sự kết hợp này mang lại lợi ích đáng kể cho cả người dùng và hệ thống, nâng cao trải nghiệm và hiệu suất của việc đề xuất sản phẩm.

3.2 LinUCB với các mô hình tuyến tính riêng biệt

Dựa trên cơ sở của việc áp dụng ngữ cảnh và mô hình tuyến tính vào trong LinUCB, ở phần này chúng em sẽ tập trung vào việc nghiên cứu phiên bản đơn giản của giải thuật LinUCB, được gọi là LinUCB với các

mô hình tuyến tính riêng biệt (LinUCB Disjoint).

Trong giải thuật LinUCB với các mô hình tuyến tính riêng biệt, mỗi bài tin tức có mô hình tuyến tính riêng độc lập với nhau. Trong đó giá trị kì vọng của điểm thưởng $r_{t,a}$ cho hành động đề xuất bài tin tức a là tuyến tính với các biến ngữ cảnh $x_{t,a}$ của nó tại thời điểm t . Ta có thể thực hiện phép nhân ma trận để dự đoán giá trị điểm thưởng cho mỗi hành động.

$$E[r_{t,a}|x_{t,a}] = x_{t,a}^T \theta_a^* \quad (3.1)$$

Các hệ số θ_a^* chưa biết nhưng nó có thể được xác định bằng cách áp dụng mô hình hồi quy ridge để ước lượng giá trị của nó qua mỗi bước thời gian t .

$$\hat{\theta}_a = (D_a^T D_a + I_d)^{-1} D_a^T c_a \quad (3.2)$$

Để đơn giản hoá phương trình, chúng ta có thể viết lại phương trình trên bằng hai biến A_a và b_a :

$$\hat{\theta}_a \leftarrow A_a^{-1} b_a \quad (3.3)$$

Trong đó:

$$A_a \stackrel{\text{def}}{=} D_a^T D_a + I_d \quad (3.4)$$

$$b_a = D_a^T c_a \quad (3.5)$$

Hồi quy ridge cho phép chúng ta xem phân phối của các hệ số $p(\theta_a)$ dưới dạng ước lượng Bayesian với giá trị trung bình Gaussian $\hat{\theta}_a$ và hiệp phương sai A_a^{-1} . Với $x_{t,a}$ là đặc trưng của người dùng, giá trị điểm thưởng kì vọng có thể tính như sau:

- Giá trị kỳ vọng

$$x_{t,a}^T \hat{\theta}_a \quad (3.6)$$

- Độ lệch chuẩn

$$\sqrt{x_{t,a}^T A_a^{-1} x_{t,a}} \quad (3.7)$$

Quá trình đề xuất bài tin tức được thực hiện bằng cách giải thuật LinUCB lựa chọn bài tin tức có giá trị UCB cao nhất tại thời điểm t .

$$a_t \stackrel{\text{def}}{=} \arg \max_{a \in A_t} \left(x_{t,a}^T \hat{\theta}_a + \alpha \sqrt{x_{t,a}^T A_a^{-1} x_{t,a}} \right) \quad (3.8)$$

Với siêu tham số α càng cao thì khoảng tin cậy càng rộng. Do đó, nó dẫn đến việc hệ thống tập trung vào khám phá nhiều hơn khai thác. Khi người dùng tương tác với hệ thống và cung cấp thông tin phản hồi, mô hình được cập nhật để cải thiện dự đoán trong tương lai. Đảm bảo rằng mô hình thực hiện một sự cân bằng giữa việc khám phá bài tin tức mới và tận dụng thông tin đã biết trong hệ thống. Điều này giúp tránh việc mô hình chỉ tập trung vào việc đề xuất những bài tin tức quen thuộc mà bỏ qua những bài tin tức mới có thể phù hợp hơn với người dùng.

Giải thuật LinUCB với mô hình tuyến tính riêng biệt có thể được mô tả bằng mã giả sau:

Algorithm 2 LinUCB với các mô hình tuyến tính riêng biệt

```
1: Input:  $\alpha \in R^+$ 
2: for  $t = 1$  to  $T$  do
3:   Quan sát các đặc trưng của bài tin tức  $a \in \mathcal{A}_t$ :  $\mathbf{x}_{t,a} \in R^d$ 
4:   for all  $a \in \mathcal{A}_t$  do
5:     if  $a$  là bài tin tức mới then
6:        $\mathbf{A}_a \leftarrow \mathbf{I}_d$  (ma trận đơn vị có kích thước là  $d$ )
7:        $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$  (vectơ không có kích thước là  $d$ )
8:     end if
9:     Tính trọng số của bài tin tức:  $\hat{\theta}_{a,t} = \mathbf{A}_a^{-1} \mathbf{b}_a$ 
10:    Tính khoảng tin cậy của bài tin tức:  $C_{a,t} = \alpha \sqrt{x_{t,a}^\top \mathbf{A}_a^{-1} x_{t,a}}$ 
11:    Tính giá trị “ucb” của bài tin tức:  $ucb = \hat{\theta}_{a,t}^\top x_t + C_{a,t}$ 
12:  end for
13:  Chọn bài tin tức:  $a_t = \arg \max_{a \in \mathcal{A}_t} ucb$ 
14:  Nhận điểm thưởng  $r_t$ 
15:  Cập nhật mô hình của bài tin tức:
16:     $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + x_{t,a_t} x_{t,a_t}^\top$ 
17:     $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t x_{t,a_t}$ 
18: end for
```

Giải thuật LinUCB với các mô hình tuyến tính riêng biệt được phân thành các pha sau đối với mỗi bước thời gian t với vectơ ngữ cảnh tương ứng x_t :

1. Nếu bài báo là mới, khởi tạo ma trận \mathbf{A}_a (kích thước $d \times d$, là ma trận đơn vị) và vectơ \mathbf{b}_a (kích thước d là vectơ không). d đại diện cho số chiều của các biến đầu vào.
2. Tính toán θ_a sử dụng công thức hồi quy ridge.
3. Sử dụng x_t , tính toán UCB của mỗi hành động dựa trên tổng của giá trị kỳ vọng trung bình và khoảng tin cậy ($\alpha * \text{độ lệch chuẩn}$).
4. Chọn hành động có giá trị “ucb” cao nhất trong tất cả hành động. Nếu có các hành động có cùng giá trị “ucb”, chọn một cách ngẫu nhiên giữa chúng.

5. Cập nhật giá trị điểm thưởng nhận được tại bước thời gian t

Giải thuật LinUCB với các mô hình tuyến tính riêng biệt có những điểm đáng chú ý sau. Đầu tiên, độ phức tạp tính toán của thuật toán này tuyến tính theo số lượng các hành động và tối đa là bậc ba theo số lượng các đặc trưng. Điều này có nghĩa là thời gian tính toán tăng theo cấp số nhân với số lượng các hành động và bậc ba với số lượng các đặc trưng. Để giảm độ phức tạp khi tính toán, giải thuật LinUCB Disjoint cập nhật ma trận A_{a_t} trong mỗi bước của giải thuật (mất $O(d^2)$ thời gian) và tính toán cũng như lưu trữ $Q_a \stackrel{\text{def}}{=} A_a^{-1}$ (cho tất cả các hành động a) một cách định kỳ thay vì thời gian thực giúp giảm độ phức tạp tính toán. Bằng cách này, chúng ta có thể tiết kiệm tài nguyên tính toán và bộ nhớ trong quá trình thực thi của giải thuật, đồng thời duy trì hiệu suất và khả năng mở rộng của nó.

Một ưu điểm quan trọng của LinUCB Disjoint là khả năng hoạt động tốt trên tập hành động động mà vẫn duy trì hiệu quả miễn là kích thước của tập hành động (A_t) không quá lớn. Điều này là hợp lý với nhiều ứng dụng trong thực tế. Ví dụ như trong việc đề xuất bài báo tin tức, các biên tập viên thường thêm hoặc xóa bài tin tức từ một nhóm bài tin tức nhưng kích thước của nhóm này thường không thay đổi đáng kể. LinUCB Disjoint cho phép cập nhật chiến lược đề xuất một cách linh hoạt và nhanh chóng trong các tình huống như vậy, đồng thời duy trì tính hiệu quả của thuật toán.

Bên cạnh việc có nhiều ưu điểm, giải thuật LinUCB với các mô hình tuyến tính riêng biệt cũng tồn tại những hạn chế nhất định. Đầu tiên, chính việc giải thuật này giả định các đặc trưng của các hành động độc lập với nhau đã dẫn đến sự không phù hợp trong một số tình huống thực tế khi mà các đặc trưng có sự tương tác phụ thuộc lẫn nhau. Ví dụ, giả sử chúng ta xem xét hai đặc trưng của sản phẩm là “giá” và “đánh giá khách hàng”. Thường thì giá của một sản phẩm sẽ ảnh hưởng đến đánh giá của khách hàng. Nếu một sản phẩm có giá cao, khả năng cao khách hàng sẽ đánh giá sản phẩm đó thấp hơn. Ngược lại, một sản phẩm có giá rẻ có

thể nhận được đánh giá cao hơn từ khách hàng. Trong trường hợp này, LinUCB Disjoint giả định rằng các đặc trưng của sản phẩm độc lập với nhau và không xem xét mối tương quan phụ thuộc giữa chúng. Điều này có thể dẫn đến việc hệ thống không thể tận dụng thông tin quan trọng về tương quan giữa giá và đánh giá của sản phẩm để đưa ra các đề xuất tốt nhất cho người dùng. Một hạn chế khác của giải thuật LinUCB Disjoint đó là việc cập nhật và lưu trữ ma trận Q_a có thể tạo ra một lượng lớn dữ liệu phụ trợ không cần thiết, đòi hỏi bộ nhớ và tài nguyên tính toán cao. Điều này có thể ảnh hưởng đến khả năng thực hiện và hiệu suất của giải thuật trên các hệ thống có tài nguyên hạn chế.

3.3 LinUCB với các mô hình tuyến tính có sự chia sẻ

Ở phần trên, chúng em đã trình bày về giải thuật LinUCB với mô hình tuyến tính riêng biệt cũng như cách để triển khai chi tiết của giải thuật. Với giải thuật trên thì giá trị điểm thưởng của mỗi hành động là một hàm tuyến tính của các biến đầu vào. Đồng thời mô hình của mỗi hành động là khác biệt và không có các đặc trưng chung.

Tuy nhiên, trong một số trường hợp, ta có thể nhận thấy mô hình của các hành động không tách biệt hoàn toàn với nhau. Ví dụ, trong một hệ thống đề xuất tin tức, một số bài tin tức được đề xuất cho người dùng có thể tương tự nhau, do đó ở những bài tin tức này đang có những đặc trưng chung nhất định. Thêm vào đó, hãy xem xét ba bài tin tức, trong đó hai bài đầu tiên liên quan đến thể thao và bài thứ ba là khoa học viễn tưởng, có khả năng cao là người dùng thích bài tin tức thể thao đầu tiên sẽ thích bài tin tức thể thao thứ hai hơn so với bài tin tức về khoa học viễn tưởng. Vì vậy, việc chia sẻ đặc trưng giữa các bài tin tức có thể mang lại hiệu quả đề xuất cao hơn.

Điều này đã đưa đến một khái niệm về một phiên bản phức tạp hơn của giải thuật LinUCB. Đó chính là giải thuật LinUCB với các mô hình tuyến tính có sự chia sẻ. Trong giải thuật này, các mô hình tuyến tính có sự kết hợp giữa hai tham số: tham số được sử dụng riêng cho mỗi hành động khác nhau và tham số được sử dụng chung cho tất cả các hành động. Phiên bản này của giải thuật LinUCB được thể hiện qua công thức sau:

$$E[r_{t,a}|x_{t,a}] = z_{t,a}^T \beta^* + x_{t,a}^T \theta_a^* \quad (3.9)$$

Trong đó:

- $x_{t,a}$ là biến ngữ cảnh giống như ở giải thuật LinUCB với các mô hình tuyến tính riêng biệt.
- $z_{t,a}$ là biến kết hợp giữa đặc trưng của bài tin tức và đặc trưng của người dùng. Nếu không có biến này thì giải thuật sẽ quay về giải thuật LinUCB với các mô hình tuyến tính riêng biệt.
- β^* và θ^* là các vectơ trọng số. Mô hình trên được gọi là có sự chia sẻ bởi vectơ β^* được chia sẻ bởi tất cả các hành động trong khi θ^* chỉ được dùng với một hành động nhất định.

Bằng cách sử dụng cả hai thông tin là đặc trưng người dùng x_t và đặc trưng của từng bài tin tức a , chúng ta có thể tạo ra $z_{t,a}$ dưới dạng tích Tensor của hai phần thông tin đó. Vì mỗi bài tin tức có các đặc trưng cụ thể riêng, điều này dẫn đến các vector $z_{t,a}$ khác nhau cho mỗi bài tin tức tương ứng với một x_t cho trước.

Trong mô hình tuyến tính, giống như $x_{t,a}$ có các hệ số tương ứng θ_a , $z_{t,a}$ cũng có các hệ số tương ứng β . Khác biệt chính là trong khi θ_a chỉ áp dụng cho một hành động cụ thể, β áp dụng cho tất cả các hành động. Giá trị điểm thưởng của hành động được chọn kết hợp với $x_{t,a}$ dùng để cập nhật hệ số θ_a và đồng thời giá trị điểm thưởng này kết hợp với $z_{t,a}$ để cập nhật hệ số β .

Để thấy cách các đặc trưng chung giữa các hành động hỗ trợ cho nhau, ta hãy cũng xem xét về ví dụ trước đó với 3 bài tin tức (2 bài về thể thao và 1 bài khoa học viễn tưởng). Ở mỗi bước thời gian t , chúng ta có một tương tác của một người dùng và do đó chỉ có một đơn vị điểm thưởng cho hành động được chọn. Giả sử ở bước thời gian tiếp theo, bài tin tức thể thao đầu tiên được chọn và chúng ta tính toán $z_{t,a}$ bằng cách sử dụng $x_{t,a}$ và các đặc trưng của bài tin tức tương ứng. Chúng ta tính toán θ_a cho bài tin tức thể thao thứ nhất, đồng thời tính toán β . Vì chúng ta giả định rằng hai bài tin tức về thể thao là tương tự nhau về các đặc trưng chung, điều này dẫn đến một giả định hợp lý là các giá trị $z_{t,a}$ của chúng sẽ tương tự nhau. Với việc sử dụng giá trị của $z_{t,a}$ để tính toán giá trị cho β , chúng ta có thể cập nhật thông tin về điểm thưởng cho các bài tin tức thể thao tương tự, mặc dù chỉ quan sát được điểm thưởng cho một bài viết về thể thao.

Giải thuật LinUCB với mô hình tuyến tính có sự chia sẻ có thể được mô tả bằng mã giả sau:

Algorithm 3 LinUCB với các mô hình tuyến tính có sự chia sẻ

Input: $\alpha \in R^+$

$A_0 \leftarrow I_d$ (ma trận đơn vị có kích thước là d)

$b_0 \leftarrow \mathbf{0}_{d \times 1}$ (vectơ không có kích thước là d)

for $t = 1$ to T **do**

Quan sát các đặc trưng của bài tin tức $\alpha \in \mathcal{A}_t$: $(\mathbf{z}_{t,a}, \mathbf{x}_{t,a}) \in R^{k+d}$

$\hat{\beta} = A_0^{-1} b_0$

for all $a \in \mathcal{A}_t$ **do**

if a là bài tin tức mới **then**

$A_a \leftarrow I_d$ (ma trận đơn vị có kích thước là d)

$B_a \leftarrow \mathbf{0}_{d \times k}$ (ma trận đơn vị kích thước là $d \times k$)

$b_a \leftarrow \mathbf{0}_{d \times 1}$ (vectơ không kích thước là d)

end if

Tính toán trọng số của bài tin tức: $\hat{\theta}_{a,t} = A_a^{-1}(b_a - B_a \hat{\beta})$

Tính: $s_{t,a} = z_{t,a}^\top A_0^{-1} z_{t,a} - 2z_{t,a}^\top A_0^{-1} B_a^\top A_a^{-1} x_{t,a} + x_{t,a}^\top A_a^{-1} x_{t,a} +$
 $x_{t,a}^\top A_a^{-1} B_a A_0^{-1} B_a^\top A_a^{-1} x_{t,a}$

Tính khoảng tin cậy: $C_{a,t} = \alpha \sqrt{s_{t,a}}$

Tính giá trị “ucb” cho bài tin tức: $ucb = z_{t,a}^\top \hat{\beta} + x_{t,a}^\top \hat{\theta}_{a,t} + C_{a,t}$

end for

Chọn bài tin tức: $a_t = \arg \max_{a \in A_t} ucb$

Nhận điểm thưởng: r_t

Cập nhật mô hình cho bài tin tức:

$A_0 \leftarrow A_0 + B_{a_t}^\top A_{a_t}^{-1} B_{a_t}$

$b_0 \leftarrow b_0 + B_{a_t}^\top A_{a_t}^{-1} b_{a_t}$

$A_{a_t} \leftarrow A_{a_t} + x_{t,a_t} x_{t,a_t}^\top$

$B_{a_t} \leftarrow B_{a_t} + x_{t,a_t} z_{t,a_t}^\top$

$b_{a_t} \leftarrow b_{a_t} + r_t x_{t,a_t}$

$A_0 \leftarrow A_0 + z_{t,a_t} z_{t,a_t}^\top - B_{a_t}^\top A_{a_t}^{-1} B_{a_t}$

$b_0 \leftarrow b_0 + r_t z_{t,a_t} - B_{a_t}^\top A_{a_t}^{-1} b_{a_t}$

end for

Giải thuật được phân thành các bước sau đối với mỗi bước thời gian t với vectơ ngữ cảnh tương ứng x_t :

1. Khởi tạo A_0 (ma trận đơn vị $k \times k$) và b_0 (vectơ 0 kích thước k). Ở đây, k đại diện cho số lượng đặc trưng khi kết hợp thông tin của

người dùng với thông tin của bài tin tức.

2. Nếu bài tin tức là mới, khởi tạo A_a (ma trận đơn vị $d \times d$), b_a (vector 0 kích thước $d \times 1$) và B_a (ma trận 0 kích thước $d \times k$). Ở đây, d đại diện cho số lượng đặc trưng của bài tin tức.
3. Tại mỗi bước thời gian, tính toán $\hat{\theta}$ bằng cách sử dụng ma trận nghịch đảo A_0 và vector b_0 .
4. Tính toán θ_a bằng cách sử dụng công thức hồi quy ridge.
5. Sử dụng x_t , xác định $z_{t,a}$ trước và tính toán giá trị “ucb” của mỗi hành động dựa trên tổng của giá trị kỳ vọng trung bình và khoảng tin cậy (alpha nhân với độ lệch chuẩn).
6. Từ tất cả các hành động, chọn hành động có giá trị “ucb” cao nhất. Nếu có nhiều hành động có cùng giá trị “ucb”, chọn ngẫu nhiên giữa chúng.
7. Nhận điểm thưởng dựa trên hành động được chọn tại bước thời gian t .
8. Cập nhật các block được chia sẻ A_0 và b_0 dựa trên $A_{a,t}$, $B_{a,t}$ và $b_{a,t}$ của hành động được chọn.
9. Cập nhật giá trị của các tham số $A_{a,t}$, $B_{a,t}$ và $b_{a,t}$ của hành động được chọn.

Như đã đề cập trước đó, $z_{t,a}$ là tích vô Tensor của cả x_t và đặc trưng của bài tin tức. Đối với mục đích triển khai giải thuật, k đại diện cho chiều kết hợp của x_t và các đặc trưng của bài tin tức.

Với mỗi $x_{t,a}$ tại một thời điểm t , hành động được lựa chọn bởi giải thuật LinUCB với các mô hình tuyến tính có sự chia sẻ được biểu diễn bởi công thức sau:

$$a_t = \arg \max_{a \in \mathcal{A}_t} p_{t,a}$$

$$p_{t,a} \leftarrow \boxed{\mathbf{z}_{t,a}^\top \hat{\boldsymbol{\beta}} + \mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a} + \boxed{\alpha \sqrt{s_{t,a}}}$$

Điểm thưởng của hành động được tính dựa trên tổng của hai thành phần:

- Ước lượng thưởng trung bình của mỗi hành động sử dụng cả các đặc trưng được chia sẻ và không được chia sẻ (khung màu xanh). Trong đó, β^* đại diện cho các đặc trưng được chia sẻ. Khác với θ_a , β được huấn luyện dựa trên tất cả các bước thời gian sao cho các hành động có đặc trưng tương tự sẽ có nhận điểm thưởng tương tự.
- Khoảng tin cậy tương ứng (khung màu đỏ), trong đó α là siêu tham số.

Tương tự như LinUCB Disjoint, giá trị α càng cao thì khoảng tin cậy càng rộng. Điều này dẫn đến việc mô hình tập trung vào việc khám phá nhiều hơn thay vì khai thác.

Ở đây, ta có thể thấy rằng giải thuật LinUCB Hybrid tính toán hiệu quả vì các khối trong giải thuật (A_0 , b_0 , A_a , B_a và b_a) đều có kích thước cố định và có thể được cập nhật theo từng bước. Hơn nữa, các đại lượng liên quan đến các hành động không tồn tại trong A_t không còn tham gia vào quá trình tính toán. Cuối cùng, chúng ta cũng có thể tính toán và lưu trữ các ma trận nghịch đảo (A_0^{-1} và A_a^{-1}) định kỳ thay vì cuối mỗi lần thử t để giảm độ phức tạp tính toán cho mỗi lần thử xuống còn $O(d^2 + k^2)$.

Chương 4

Kết quả thí nghiệm

Trong chương này, chúng em sẽ trình bày các thí nghiệm để đánh giá hiệu suất của giải thuật LinUCB ở chương 3. Đầu tiên, chúng em sẽ nói về cách thiết lập các thí nghiệm, bao gồm: mô tả tập dữ liệu, trình bày phương pháp rút trích đặc trưng và phương pháp đánh giá giải thuật LinUCB cũng như các giải thuật học tăng cường khác bằng dữ liệu “offline”. Ở thí nghiệm đầu tiên, chúng em sẽ so sánh kết quả cài đặt giải thuật LinUCB và các giải thuật khác của khóa luận với bài báo gốc. Ở thí nghiệm thứ 2, chúng em sẽ đánh giá hiệu suất giải thuật LinUCB với các giá trị siêu tham số khác nhau. Ở thí nghiệm thứ 3, chúng em sẽ so sánh giải thuật LinUCB sử dụng các mô hình tuyến tính riêng biệt với LinUCB sử dụng các mô hình tuyến tính có sự chia sẻ. Với thí nghiệm cuối cùng, chúng em sẽ so sánh giải thuật LinUCB với Lin ϵ -Greedy (giải thuật ϵ -Greedy sử dụng mô hình tuyến tính).

4.1 Thiết lập các thí nghiệm

4.1.1 Tập dữ liệu

Để tiến hành các thí nghiệm trong khóa luận, chúng em sử dụng bộ dữ liệu “R6A - Yahoo! Front Page Today Module User Click Log Dataset”

là một bộ dữ liệu sự kiện thu thập thông tin về hành vi người dùng trên trang chủ của Yahoo, cụ thể hơn là hành vi của người dùng ở phần “Today Module”.



Hình 4.1: Hình ảnh phần nổi bật của “Today Module” trên trang chủ của “Yahoo!”. Theo mặc định bài tin tức ở vị trí F1 sẽ được hiển thị nổi bật ở vị trí “STORY”.

Như minh họa ở hình 4.1, sẽ có 4 bài tin tức ở vị trí cuối trang, được đánh số từ ‘F1’ đến ‘F4’. Mỗi bài tin tức sẽ được hiển thị bằng một bức ảnh và tiêu đề. Một trong 4 bài tin tức này sẽ được hiển thị nổi bật ở phần “STORY” với thông tin liên quan đến bài tin tức được hiển thị nhiều hơn. Điều này cũng giống như việc đề xuất bài tin tức cho người dùng. Tương tác giữa người dùng với bài tin tức (nhấp chuột hoặc không nhấp chuột) sẽ được ghi lại như một sự kiện.

Bộ dữ liệu này được thu thập trên một nhóm người dùng ngẫu nhiên vào tháng 5 năm 2009. Hệ thống sẽ đề xuất ngẫu nhiên một bài tin tức cho người dùng theo phân phối đều. Việc đề xuất ngẫu nhiên này giúp cho phương pháp đánh giá các giải thuật của tác giả bài báo [1] có được sự tin cậy, sự khách quan cũng như không bị thiên lệch và đạt được hiệu quả. Tập dữ liệu này chứa 45.811.883 lượt truy cập của người dùng vào “Today

Module”. Mỗi sự kiện được ghi lại khi người dùng tương tác với bài tin tức gồm 3 thành phần:

- Thông tin bài tin tức được lựa chọn ngẫu nhiên để đề xuất cho người dùng, cụ thể là ‘id’ của bài tin tức.
- Thông tin của người dùng cũng như là thông tin của các bài tin tức ở vị trí cuối trang. Thông tin này là 1 vector 6 chiều tương ứng với việc gom nhóm người dùng/bài tin tức thành 5 nhóm, thể hiện xác suất của người dùng/bài tin tức thuộc từng nhóm và một đặc trưng cố định có giá trị là 1. Việc gom nhóm người dùng dựa trên các tiêu chí như giới tính, độ tuổi, địa điểm, hành vi tương tác của người dùng,... đối với bài tin tức dựa trên các tiêu chí như nội dung, chủ đề, nguồn,...
- Thông tin về việc người dùng có nhấp chọn vào bài tin tức được đề xuất ở phần “STORY” hay không. Thông tin này cũng chính là giá trị điểm thưởng (1 hoặc 0; 1: khi người dùng nhấp chọn vào bài tin tức, 0: khi người dùng không nhấp chọn)

Ở các chương trước, chúng em đã trình bày về các giải thuật như là ϵ -Greedy, UCB và LinUCB. Đặc điểm chung của các giải thuật này đều có đầu vào là siêu tham số và giá trị siêu tham số này ảnh hưởng rất lớn đến hiệu suất của giải thuật. Chính vì vậy nhóm tác giả đã chia bộ dữ liệu thành 2 phần. Phần thứ được gọi là “tuning data” chứa dữ liệu sự kiện của ngày 01/05/2009, với khoảng 4.7 triệu sự kiện; “tuning data” được sử dụng để chọn siêu tham số mang lại hiệu suất tốt nhất cho giải thuật. Phần còn lại được gọi là “evaluation data” chứa dữ liệu sự kiện từ ngày 03/05/2009 đến 09/05/2009, với khoảng 36 triệu sự kiện; “evaluation data” được sử dụng để đánh giá hiệu suất của các giải thuật, trong đó các giải thuật này sử dụng siêu tham số đã chọn được ở phần “tuning data”.

4.1.2 Phương pháp rút trích đặc trưng

Trong phần này, chúng em sẽ trình bày cách mà nhóm tác giả của bài báo [1] đã rút trích các đặc trưng của người dùng và bài tin tức để phục vụ cho việc tiến hành các thí nghiệm.

Đối với đặc trưng của người dùng. Đầu tiên, để giảm thiểu độ nhiễu trong dữ liệu, nhóm tác giả sẽ giữ lại các đặc trưng có “độ hỗ trợ” ít nhất là 0.1. “Độ hỗ trợ” của một đặc trưng chính là tỷ lệ phần trăm người dùng có đặc trưng đó. Sau khi đã giảm nhiễu, vector đặc trưng của người dùng sẽ có hơn 1000 thuộc tính, bao gồm: thông tin về giới tính (2 thuộc tính), độ tuổi (10 thuộc tính), đặc trưng địa lí (khoảng 200 thuộc tính, mỗi thuộc tính ứng với một địa điểm trên thế giới) và sẽ có khoảng 1000 thuộc tính mô tả lịch sử truy cập, tương tác của người dùng trên trang chủ của Yahoo!. Các thuộc tính này đều có giá trị nhị phân.

Tương tự, mỗi bài tin tức được biểu diễn bởi một vectơ đặc trưng có khoảng 100 thuộc tính. Những đặc trưng này bao gồm: nguồn của bài tin tức và chủ đề được gán nhãn bởi biên tập viên, với mỗi thông tin như vậy có đến hàng chục thuộc tính.

Với mục đích tăng tốc độ tính toán và giảm thiểu việc sử dụng nguồn tài nguyên. Nhóm tác giả đã giảm chiều các vector và gom nhóm đặc trưng của các bài tin tức cũng như người dùng. Cụ thể là:

- Đầu tiên, nhóm tác giả sẽ sử dụng hồi quy Logistic để tính toán xác suất nhấp chuột dựa trên đặc trưng của người dùng và bài tin tức. Sao cho $\phi_u^T \mathbf{W} \phi_a$ xấp xỉ với xác suất người dùng u sẽ nhấp chọn bài tin tức a , trong đó ϕ_u và ϕ_a tương ứng với vector đặc trưng của người dùng và bài tin tức, \mathbf{W} là ma trận trọng số được tối ưu hóa bằng hồi quy Logistic.

- Sau đó, vector đặc trưng của người dùng được chiếu lên không gian mới bằng cách tính toán $\psi_u = \phi_u^T \mathbf{W}$. Ở đây, phần tử thứ i trong ψ_u cho người dùng u là mức độ người dùng u thích bài viết thứ i . Tiếp đến, nhóm tác giả sẽ sử dụng K-means để gom nhóm các đặc trưng của người dùng trong không gian ψ_u thành 5 cụm. Vector đặc trưng cuối cùng của người dùng sẽ có 6 phần tử. Giá trị của năm phần tử đầu tiên thể hiện khả năng thuộc từng nhóm của người dùng trong năm nhóm khác nhau, tổng giá trị của năm phần tử này là 1, phần tử thứ 6 là hằng số có giá trị là 1.

Tương tự cho người dùng, đối với mỗi bài tin tức a , nhóm tác giả cũng đã thực hiện việc giảm chiều và gom cụm để thu được vector đặc trưng 6 chiều. Tích Tensor 2 vector đặc trưng của người dùng và bài tin tức tạo ra vector đặc trưng có 36 chiều, được ký hiệu là $z_{t,a}$ và được sử dụng trong giải thuật LinUCB với các mô hình tuyến tính có sự chia sẻ.

Nhóm tác giả đã lựa chọn không gian của vector đặc trưng cho người dùng và bài tin tức là tương đối nhỏ vì trong thực tế, khi triển khai trên hệ thống thực, lượng thông tin về người dùng và bài tin tức là rất lớn, nên với không gian nhỏ của các vector sẽ giúp tối ưu được việc tính toán và lưu trữ, tăng tốc độ phản hồi, từ đó đem lại sự trải nghiệm tốt nhất cho người dùng.

4.1.3 Phương pháp đánh giá các giải thuật

Đánh giá hiệu suất của các giải thuật LinUCB cũng như các giải thuật khác được áp dụng trong hệ thống đề xuất sản phẩm là vô cùng khó khăn, tuy nhiên vẫn có một số phương pháp như là: đánh giá dựa trên dữ liệu “online” và đánh giá dựa trên dữ liệu “offline”. Đánh giá trên dữ liệu “online” nghĩa là chúng ta triển khai giải thuật đề xuất sản phẩm trên hệ thống thực, giải thuật sẽ trực tiếp đề xuất bài tin tức cho người dùng dựa trên những thông tin về người dùng và bài tin tức, sau đó sẽ nhận phản

hồi trực tiếp từ người dùng thông qua việc tương tác với bài tức đã được đề xuất. Tuy nhiên việc đánh giá dựa trên dữ liệu “online” gặp thách thức vô cùng lớn trong vấn đề cơ sở hạ tầng và nguồn tài nguyên vì việc phải triển khai trên hệ thống thực. Chính vì vậy, trong khóa luận phương pháp đánh giá dựa trên dữ liệu “offline” sẽ phù hợp hơn trong việc đánh giá hiệu suất giải thuật LinUCB cũng như các giải thuật khác.

Một giải pháp trong đánh giá dựa trên dữ liệu “offline” đó chính là xây dựng hệ thống giả lập dựa trên những dữ liệu đã được ghi lại từ hệ thống thực, sau đó sẽ đánh giá các giải thuật thông qua hệ thống giả lập này. Tuy nhiên, hệ thống giả lập không đủ độ tin cậy cho việc đánh giá vì lý do bị thiên lệch khi sử dụng bộ dữ liệu trong hệ thống thực, bộ dữ liệu này thường chứa các thông tin về bài báo mà khả năng là người dùng có hứng thú cao, nên không đảm bảo được tính khách quan trong hệ thống giả lập này.

Để khắc phục vấn đề bị thiên lệch được trình bày ở trên, nhóm tác giả của bài báo [1] đã đề xuất một phương pháp đánh giá đơn giản dựa trên tập dữ liệu “R6A - Yahoo! Front Page Today Module User Click Log Dataset” mà chúng em đã trình bày ở phần 4.1.1. Nhóm tác giả cũng đã trình bày phương pháp đánh giá này trong bài báo [2].

Algorithm 4 Phương pháp đánh giá

```
1: Input:  $T > 0$ ; Giải thuật  $\pi$ ; Chuỗi dữ liệu sự kiện
2:  $h_0 \leftarrow \theta$  Khởi tạo 1 danh sách lịch sử rỗng
3:  $R_0 \leftarrow 0$  Khởi tạo giá trị tổng phần thưởng là 0
4: for  $t = 1$  to  $T$  do
5:   repeat
6:     Lấy dữ liệu sự kiện kế tiếp là:  $(x_1, \dots, x_K, a, r_a)$ 
7:   until  $\pi(h_{t-1}, (x_1, \dots, x_K)) = a$ 
8:    $h_t \leftarrow \text{CONCATENATE}(h_{t-1}, (x_1, \dots, x_K, a, r_a))$ 
9:    $R_t \leftarrow R_{t-1} + r_a$ 
10: end for
11: Output:  $R_T / T$ 
```

Phương pháp đánh giá (4) có đầu vào là:

- T : Số lượng các vòng lặp.
- π : Giải thuật đề xuất bài báo mà chúng ta muốn đánh giá.
- Chuỗi dữ liệu sự kiện là những sự kiện đã được thu thập theo phương pháp đã được trình bày ở phần 4.1.1.

Đầu ra của thuật toán (4) là R_T / T còn được gọi là CTR (click through rate) - tỉ lệ nhấp chuột của người dùng vào bài tin tức được đề xuất, nhờ vào CTR chúng ta có thể đánh giá hiệu suất của giải thuật LinUCB và các giải thuật khác. Giá trị CTR càng lớn chứng tỏ hiệu suất của giải thuật càng cao. Ngoài ra, khái niệm “Relative CTR” cũng được dùng để đánh giá hiệu suất của giải thuật, giá trị của “Relative CTR” được tính bằng cách lấy giá trị CTR của giải thuật π chia cho giá trị CTR của giải thuật “random”. Với giải thuật “random” sẽ luôn luôn thực hiện việc đề xuất ngẫu nhiên một bài tin tức cho người dùng tại mọi thời điểm.

Với phương pháp đánh giá các giải thuật, đầu tiên sẽ khởi tạo h_0 để lưu lại các sự kiện được xem là sự kiện “phù hợp”. Một sự kiện “phù hợp” là sự kiện thuộc “Chuỗi dữ liệu sự kiện” và tại 1 thời điểm t , sự kiện này

chứa thông tin bài tin tức trùng với thông tin bài tin tức mà giải thuật đã đề xuất cho người dùng. Thông tin bên trong các sự kiện này bao gồm giá trị điểm thưởng, thông tin của người dùng, thông tin của bài tin tức, ... sẽ được dùng để cập nhật các tham số và giúp cải thiện hiệu suất của giải thuật. Bên cạnh đó, phương pháp đánh giá này cũng sẽ khởi tạo R_0 để tính tổng điểm thưởng đạt được khi đề xuất bài tin tức cho người dùng. Phương pháp này thực hiện T vòng lặp, trong mỗi vòng lặp, sẽ liên tục xem xét thông tin của các sự kiện chưa được xét đến, cho đến khi có một sự kiện được xem sự kiện là “phù hợp”, giải thuật sẽ cập nhật lịch sử và giá trị tổng điểm thưởng, sau đó sẽ qua vòng lặp tiếp theo.

4.2 Thí nghiệm 1: So sánh kết quả cài đặt của khóa luận với bài báo gốc

Trong thí nghiệm đầu tiên, chúng em sẽ đánh giá giải thuật LinUCB ở chương 3 và hai giải thuật UCB, ε -Greedy ở chương 2 dựa trên tập “evaluation data” được giới thiệu ở phần 4.1.1. Sau đó chúng em sẽ sử dụng kết quả đánh giá để so sánh với kết quả trong bài báo gốc [1].

Giải thuật	Kết quả của khóa luận	Kết quả của bài báo
LinUCB Hybrid	1.487	1.663
LinUCB Disjoint	1.59	1.647
UCB	1.528	1.569
ε -Greedy	1.499	1.326

Bảng 4.1: Kết quả của cài đặt của khóa luận và của bài báo gốc với độ đo là “Relative CTR” trên tập “evaluation data”

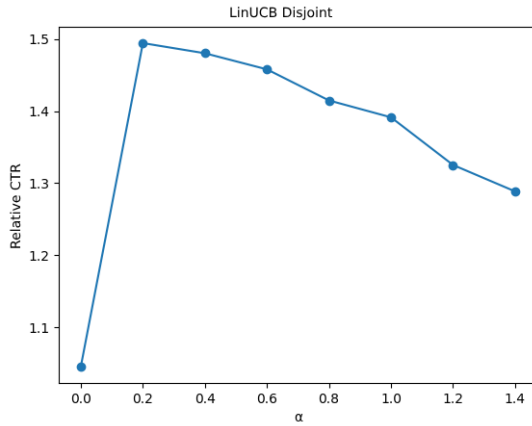
Dựa vào bảng 4.1, ta thấy được ngoài giải thuật ε -Greedy, kết quả của các giải thuật khác có phần thấp hơn so với kết quả của bài báo gốc. Và trong bài báo gốc có một số chi tiết về việc thiết lập các thí nghiệm không

được nhóm tác giả công bố, nên đó cũng chính là một phần nguyên nhân dẫn đến khác biệt về kết quả. Đối với giải thuật ε -Greedy, ngoài vấn đề về thiết lập thí nghiệm, giải thuật này còn có yếu tố ngẫu nhiên và đó cũng là một trong những nguyên nhân dẫn đến kết quả trong khóa luận của giải thuật này cao hơn kết quả trong bài báo gốc.

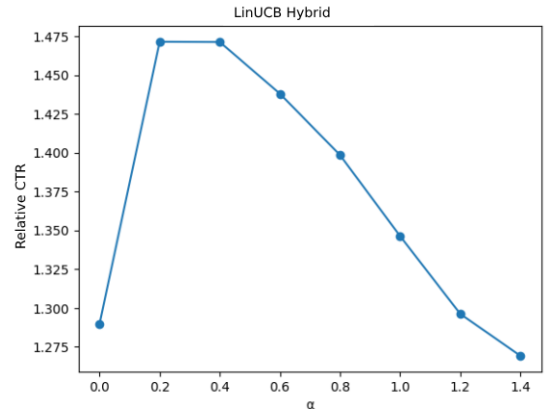
Ngoài kết quả so sánh giữa khóa luận và bài báo gốc, chúng ta còn có thể thấy được đối với các thuật toán có ngữ cảnh như LinUCB Disjoint có kết quả tốt hơn các giải thuật phi ngữ cảnh như là UCB và ε -Greedy.

4.3 Thí nghiệm 2: Đánh giá hiệu suất giải thuật LinUCB với các giá trị siêu tham số khác nhau.

Trong thí nghiệm này, chúng em sẽ trình bày hiệu suất của giải thuật LinUCB với các giá trị siêu tham số khác nhau, độ đo cho hiệu suất của giải thuật là “Relative CTR”. Từ đó ta có thể thấy rõ hơn mức độ quan trọng của siêu tham số trong giải thuật.



(a) LinUCB Disjoint



(b) LinUCB Hybrid

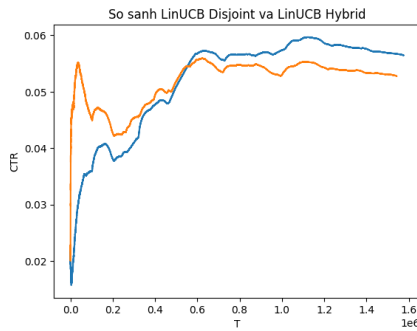
Hình 4.2: Đánh giá hiệu suất giải thuật LinUCB với các giá trị siêu tham số khác nhau trên tập “tuning data”.

Như kết quả trong hình 4.2, ta có thể thấy rằng đường cong được tạo bởi các giá trị Relative CTR có hình chữ ‘U’ ngược. Khi giá trị của siêu tham số α quá nhỏ, việc khám phá ít được xảy ra, giải thuật không xác định được các bài tin tức tốt nhất, dẫn đến tỉ lệ nhấp chọn bài tin tức thấp. Ngược lại khi α quá lớn, giải thuật dường như chỉ thực hiện việc khám phá, dẫn đến việc có thể tốn nhiều thời gian và tài nguyên cho những bài tin tức không quan trọng, đồng thời cũng lãng phí những cơ hội lựa chọn những bài tin tức tốt hơn cho, điều này cũng dẫn đến việc tỉ lệ người dùng nhấp chọn thấp. Khi giá trị α ở mức trung bình, như kết quả của thí nghiệm là $\alpha = 0.2$ cho cả 2 giải thuật là LinUCB Hybrid và LinUCB Disjoint, sẽ giúp giải thuật đạt được sự tối ưu trong việc cân bằng giữa khai thác bài tin tức tốt nhất hiện có và khám phá những bài tin tức tiềm năng khác, từ đó đem lại kết quả tối ưu nhất.

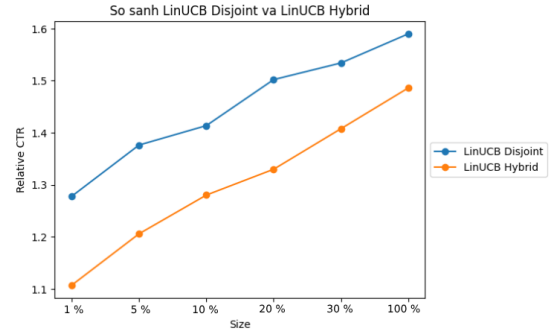
4.4 Thí nghiệm 3: So sánh LinUCB Disjoint với LinUCB Hybrid

Trong thí nghiệm 3, chúng em sẽ so sánh hiệu suất của giải thuật LinUCB với các mô hình tuyến tính riêng biệt (LinUCB Disjoint) và LinUCB

với các mô hình tuyến tính có sự chia sẻ (LinUCB Hybrid) dựa trên tập “evaluation data”. Kết quả trong thí nghiệm này sẽ giúp chúng ta đánh giá được hiệu suất của giải thuật khi sử dụng tham số dùng chung cho các mô hình tuyến tính.



(a) Giá trị CTR qua từng bước thời gian T trên toàn bộ tập dữ liệu



(b) Giá trị Relative CTR với các tập dữ liệu có kích thước khác nhau.

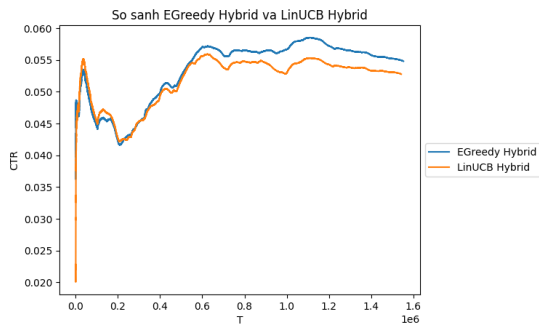
Hình 4.3: So sánh hiệu suất của LinUCB Hybrid và LinUCB Disjoint trên tập “evaluation data”

Khi quan sát kết quả ở hình 4.3a, ta thấy được rằng với những bước thời gian đầu tiên, LinUCB Hybrid có hiệu suất tốt hơn so với LinUCB Disjoint, từ đó ta có những cái nhìn đầu tiên về độ hiệu quả của “tham số dùng chung” cho các mô hình tuyến tính.

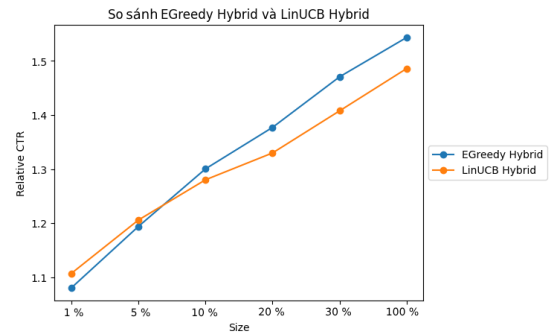
Từ hình 4.3b ta thấy được hiệu suất của giải thuật LinUCB Disjoint có phần cao hơn so với LinUCB Hybrid trên toàn bộ các kích thước của tập dữ liệu, qua đó dựa vào kết quả trên chúng ta có thể thấy rằng “tham số dùng chung” cho các mô hình tuyến tính chưa đem lại được sự hiệu quả cao; như chúng em đã trình bày ở thí nghiệm 1, một số chi tiết về thiết lập các thí nghiệm trong bài báo gốc không được công bố, nên đó cũng là một phần nguyên nhân dẫn đến hiệu suất của LinUCB Hybrid thấp hơn LinUCB Disjoint.

4.5 Thí nghiệm 4: So sánh LinUCB với Lin ϵ -Greedy

Trong thí nghiệm 4, chúng em sẽ so sánh hiệu suất của 2 giải thuật LinUCB với Lin ϵ -Greedy. Đầu tiên chúng em sẽ so sánh LinUCB Hybrid với Lin ϵ -Greedy Hybrid (ϵ -Greedy với các mô hình tuyến tính có sự chia sẻ) và sau đó chúng em sẽ so sánh LinUCB Disjoint với Lin ϵ -Greedy Disjoint (ϵ -Greedy với các mô hình tuyến tính riêng biệt). Với mục đích đánh giá hiệu suất của các giải thuật khi cả hai đều sử dụng các mô hình tuyến tính nhưng khác nhau ở phương pháp cân bằng giữa việc khám phá và khai thác.



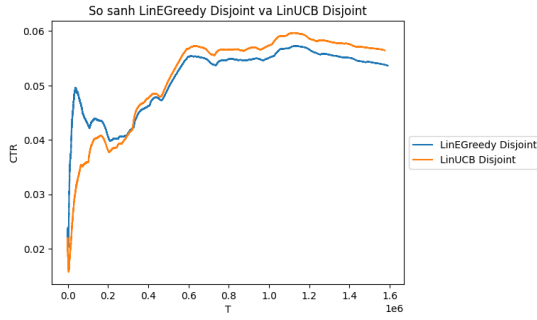
(a) Giá trị CTR qua từng bước thời gian T trên toàn bộ tập dữ liệu



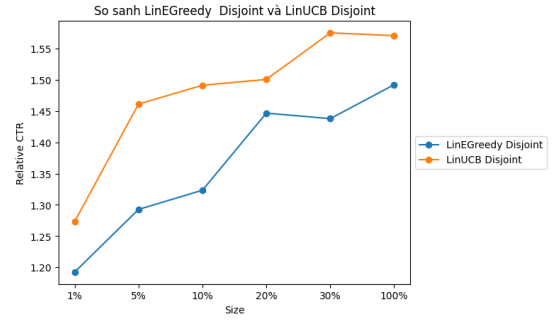
(b) Giá trị Relative CTR với các tập dữ liệu có kích thước khác nhau

Hình 4.4: So sánh hiệu suất của Lin ϵ -Greedy Hybrid và LinUCB Hybrid trên tập “evaluation data”

Quan sát hình 4.4a và hình 4.4b, ta có thể thấy hiệu suất của LinUCB Hybrid so với Lin ϵ -Greedy Hybrid có sự tương đồng khá cao. Kết quả này có thể được giải thích một phần bởi vì cả hai giải thuật đều sử dụng các mô hình tuyến tính có sự chia sẻ.



(a) Giá trị CTR qua từng bước thời gian T trên toàn bộ tập dữ liệu



(b) Giá trị Relative CTR với các tập dữ liệu có kích thước khác nhau

Hình 4.5: So sánh hiệu suất của Lin ϵ -Greedy Hybrid và LinUCB Hybrid trên tập “evaluation data”

Theo như kết quả trong hình 4.5a, ta có thể thấy được sự tương đồng khá lớn giữa hiệu suất của giải thuật LinUCB Disjoint và Lin ϵ -Greedy Disjoint. Bên cạnh đó, khi ta quan sát kỹ hơn, trong những bước đầu tiên, hiệu suất của Lin ϵ -Greedy Disjoint có phần cao hơn so với LinUCB Disjoint và trong những bước tiếp theo, ta có thể quan sát được hiệu suất có sự đảo chiều, hiệu suất của LinUCB Disjoint cao hơn so với Lin ϵ -Greedy Disjoint. Quan sát hình 4.5b, ta có thể thấy được trên mọi kích thước của tập dữ liệu, LinUCB Disjoint đạt được hiệu suất cao hơn so với Lin ϵ -Greedy Disjoint.

Từ những đánh giá về hiệu suất của giải thuật LinUCB và Lin ϵ -Greedy với các mô hình khác nhau ở trên. Ta có thể thấy rằng, trong trường hợp giải thuật có nhiều thông tin về bài tin tức, thì hiệu suất của LinUCB và Lin ϵ -Greedy khá tương đồng; và ngược lại trong trường hợp giải thuật có ít thông tin về bài tin tức, LinUCB hiệu quả hơn Lin ϵ -Greedy trong việc đề xuất tin tức cho người dùng. Qua đó ta có thể đánh giá được phần nào về độ hiệu quả trong phương pháp cân bằng giữa khai thác và khám phá của LinUCB so với Lin ϵ -Greedy.

Chương 5

Tổng kết và hướng phát triển

5.1 Tổng kết

Trong khoá luận này, chúng em đã tìm hiểu về phương pháp học tăng cường cụ thể là giải thuật LinUCB được trình bày trong bài báo [1] để giải quyết bài toán đề xuất sản phẩm. Phương pháp này có nhiều ưu điểm như:

- **Hiệu quả cao:** Qua các thí nghiệm mà chúng em đã trình bày ở chương 4, bằng việc kết hợp mô hình tuyến tính và thông tin về ngữ cảnh, có thể thấy giải thuật LinUCB mang lại hiệu quả trong việc đề xuất sản phẩm cao hơn so với các giải thuật khác như UCB, ϵ -Greedy, ... LinUCB giúp cho việc đề xuất sản phẩm được cá nhân hoá với từng người dùng. Từ đó nâng cao hiệu quả đề xuất sản phẩm.
- **Khả năng cân bằng giữa khai thác và khám phá:** Giải thuật LinUCB sử dụng một giải pháp thông minh để cân bằng giữa việc khai thác những hành động đã được biết đến và việc khám phá những hành động mới. Giải thuật sử dụng khoảng tin cậy để đánh giá rủi ro của mỗi hành động và chọn hành động có tiềm năng tốt nhất. Điều này giúp tối ưu sự cân bằng giữa việc tận dụng thông tin hiện có và khám phá những hành động mới.

- **Dễ dàng cài đặt và thực hiện:** Giải thuật LinUCB có thể được cài đặt một cách dễ dàng bằng cách sử dụng “Hồi quy Ridge” để tính toán các giá trị ước lượng và khoảng tin cậy.

Ngoài những ưu điểm trên, giải thuật LinUCB cũng tồn tại những hạn chế nhất định:

- **Giới hạn của mô hình tuyến tính:** LinUCB sử dụng ngữ cảnh trong mô hình tuyến tính để ước lượng giá trị cho từng hành động. Tuy nhiên, nếu ngữ cảnh không có hoặc không chính xác dẫn tới việc ước lượng sai giá trị cho các hành động, sẽ ảnh hưởng rất lớn đến hiệu suất của giải thuật. Chính vì vậy, việc đảm bảo chất lượng thông tin ngữ cảnh trong LinUCB rất là quan trọng.
- **Yêu cầu nhiều tài nguyên tính toán:** Việc cập nhật và lưu trữ các ma trận đòi hỏi hệ thống phải có bộ nhớ và khả năng tính toán cao. Điều này có thể ảnh hưởng đến khả năng thực hiện và hiệu suất của giải thuật trên các hệ thống thực có tài nguyên hạn chế.

5.2 Hướng phát triển

Đối với hướng phát triển trong tương lai, giải thuật LinUCB có thể áp dụng đối với nhiều sản phẩm như âm nhạc, phim, Ngoài ra, một hướng phát triển khác là kết hợp phương pháp học tăng cường với học sâu và học tăng cường chuyên sâu (deep reinforcement learning và deep contextual bandits). Học sâu có khả năng học các biểu diễn phức tạp và trừu tượng từ dữ liệu, trong khi học tăng cường chuyên sâu kết hợp cả khả năng học sâu và khả năng đưa ra quyết định tối ưu trong môi trường tương tác. Bằng cách kết hợp các phương pháp này, có thể tạo ra các giải thuật mạnh mẽ có khả năng mô hình hóa và học từ dữ liệu phức tạp và đưa ra quyết định chính xác trong các bài toán thực tế.

Tài liệu tham khảo

Tiếng Anh

- [1] Li, Lihong et al. “A Contextual-Bandit Approach to Personalized News Article Recommendation”. In: *The 19th International World Wide Web Conference* (2012).
- [2] Li, Lihong et al. “Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms”. In: *The Fourth ACM International Conference on Web Search and Data Mining* (2012).
- [3] Lops, Pasquale, Gemmis, Marco De, and Semeraro, Giovanni. “Content-based Recommender Systems: State of the Art and Trends”. In: *Recommender Systems Handbook*. 2011, pp. 73–105.
- [4] Zhang, Ruisheng et al. “Collaborative Filtering for Recommender Systems”. In: *The Second International Conference on Advanced Cloud and Big Data* (2014).
- [5] Zhao, YXiangyu et al. “Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems”. In: *AAAI Conference on Artificial Intelligence* (2019).