

Báo cáo tuần 4

**Tìm hiểu Learning to Rank,
Phân tích dữ liệu EB-NeRD & Thử nghiệm
RecGPT**



Người phụ trách: Anh Ngô Văn Vĩ

Người thực hiện: Nguyễn Đình Hiếu

Mục lục

Phần 1: Tìm hiểu Learning to Rank	3
1. Giới thiệu	3
2. Các phương pháp.....	4
2.1 Pointwise	4
2.2 Pairwise	4
2.3 Listwise	5
3. Thách thức thực tiễn khi triển khai LTR	5
Phần 2: Phân tích dữ liệu EB-NeRD	6
1. Giới thiệu	6
2. Dataset Format.....	6
2.1 Behaviors.....	6
2.2 History	7
2.3 Articles	7
2.4 Artifacts	7
Phần 3: RecGPT	8
1. Nội dung Paper	8
1.1 Đặt vấn đề	8
1.2 Cách hoạt động	8
1.3 Dữ liệu thử nghiệm.....	9
1.4 Kết quả và ứng dụng	9
2. Thủ nghiệm	10
Nguồn tham khảo:	11

Phần 1: Tìm hiểu Learning to Rank

1. Giới thiệu

Muốn tìm đúng món đồ khi mua sắm có thể rất mệt mỏi. Có thể sẽ mất hàng giờ liền để tìm kiếm những món đồ chỉ hơi liên quan, rồi cuối cùng bỏ cuộc trong vì thất vọng. Đây có thể là lý do tại sao 79% người dùng nếu không tìm thấy thứ họ muốn sẽ rời đi và tìm kiếm trên một trang web khác.

Ta cần phải có cách nào đó tốt hơn để phục vụ khách hàng như một hệ thống tìm kiếm hiệu quả hơn trên trang web chặng hạn.

“Learning to rank” là một phương pháp học máy giúp đưa ra các kết quả tìm kiếm phù hợp — và quan trọng hơn là, được **xếp hạng** theo mức độ liên quan. Và là thành phần quan trọng trong một hệ thống gợi ý.

Learning to Rank (LTR) là một nhóm các kỹ thuật thuật toán áp dụng học máy có giám sát để giải quyết các bài toán xếp hạng cải thiện mức độ liên quan của kết quả tìm kiếm trên trang web. Nói cách khác, nó là thứ sắp xếp thứ tự các kết quả khi người dùng tìm kiếm.

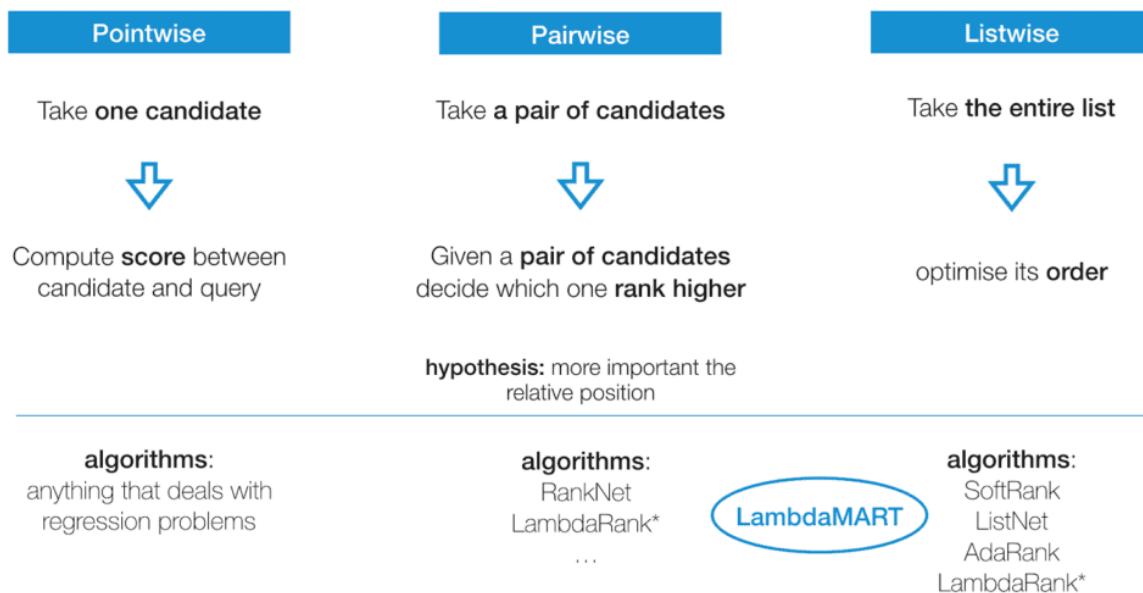
Để xây dựng **LTR**, cần có dữ liệu gồm hành vi người dùng, hồ sơ người dùng, và cần một công cụ tìm kiếm mạnh mẽ như SOLR.

Dữ liệu huấn luyện cho một mô hình **LTR** bao gồm danh sách các kết quả cho một truy vấn, kèm theo mức độ liên quan của từng kết quả đối với truy vấn đó. Ta cần xem xét các kết quả và quyết định có đưa từng kết quả vào tập dữ liệu hay không.

Tập dữ liệu đã được chọn lọc này trở thành “gold standard” của mô hình để đưa ra dự đoán. Chúng ta gọi nó là “ground truth” và sử dụng nó để đo lường độ chính xác của các dự đoán mà mô hình đưa ra.

2. Các phương pháp

Ba phương pháp chính trong **Learning to Rank (LTR)** là: **Pointwise**, **Pairwise** và **Listwise**.



2.1 Pointwise

Phương pháp **pointwise** xem xét **từng item riêng lẻ**, sử dụng phân loại (classification) hoặc hồi quy (regression) để tìm ra thứ hạng tốt nhất cho từng kết quả.

Chúng ta sẽ cho mỗi item một điểm số dựa trên mức độ phù hợp, sau đó cộng điểm và sắp xếp kết quả theo điểm. Điểm số của mỗi item **được tính độc lập**, không phụ thuộc vào các item khác.

2.2 Pairwise

Phương pháp **pairwise** xem xét **hai item cùng lúc**. Nó cũng sử dụng classification hoặc regression để quyết định item nào nên xếp trên.

Chúng ta so sánh cặp item này với **ground truth** (tập dữ liệu chuẩn đã được sắp hạng thủ công) và điều chỉnh nếu thứ tự không khớp. Mục tiêu là giảm số lần sắp xếp sai cặp, tức là giảm số trường hợp thứ tự của cặp khác với thứ tự chuẩn (gọi là **inversions**).

2.3 Listwise

Phương pháp **listwise** xử lý **toàn bộ danh sách item cùng lúc**. Máy học sẽ học từ các danh sách được sắp xếp đúng (ground truth list) để tìm ra cách sắp xếp tối ưu cho danh sách của mình.

Listwise sử dụng các mô hình xác suất để tối thiểu hóa lỗi sắp xếp toàn cục.

Phương pháp này có thể phức tạp hơn nhiều so với pointwise hay pairwise, nhưng mang lại hiệu quả cao trong những bài toán đòi hỏi độ chính xác cao.

3. Thách thức thực tiễn khi triển khai LTR

Cần phải quyết định phương pháp tiếp cận trước khi bắt đầu xây dựng mô hình.

Với cùng một bộ dữ liệu, nên huấn luyện một mô hình duy nhất cho toàn bộ dữ liệu, hay nhiều mô hình riêng biệt cho từng tập dữ liệu? Các mô hình LTR phổ biến hiện nay thực sự hoạt động tốt đến mức nào cho bài toán cần giải quyết?

Đặc biệt, mô hình được huấn luyện cần có khả năng **tổng quát hóa** với:

- Các **truy vấn chưa từng xuất hiện** trong tập huấn luyện.
- Các **item chưa từng thấy** nhưng cần được xếp hạng cho các truy vấn có trong tập huấn luyện.

Phần 2: Phân tích dữ liệu EB-NeRD

1. Giới thiệu

Bộ dữ liệu Ekstra Bladet News Recommendation Dataset (EB-NeRD) hỗ trợ nghiên cứu về hệ thống gợi ý tin tức, thu thập từ hành vi người dùng của trang báo Đan Mạch Ekstra Bladet trong 6 tuần (từ 27/4 đến 8/6/2023).

Bộ dữ liệu bao gồm:

- Hơn 1 triệu người dùng, hơn 37 triệu impression logs, hơn 251 triệu lượt tương tác, hơn 125 000 bài báo.
- Log hành vi người dùng hoạt động.
- Tin tức kèm theo các đặc trưng văn bản như tiêu đề, tóm tắt, nội dung, danh mục, v.v.
- Các đặc trưng sinh ra từ mô hình nội bộ như: chủ đề (topics), thực thể tên (NER), và embedding bài báo.

Thời gian thu thập dữ liệu được chọn để tránh các sự kiện đặc biệt nhằm đảm bảo hành vi người dùng ở mức bình thường.

2. Dataset Format

Tập dữ liệu gồm 4 file là Behaviors, History, Articles, Artifacts.

2.1 Behaviors

Chứa log hành vi người dùng (impression logs). Mỗi dòng là một impression (một sự kiện hệ thống để xuất danh sách bài viết cho người dùng).

Dữ liệu gồm cả log kỹ thuật (ví dụ: thời gian đọc, cuộn chuột) và thông tin người dùng (giới tính, độ tuổi,...).

Cột quan trọng trong File này theo em là Inview Articles IDs (Danh sách các bài viết được hiển thị trong lượt impression này) dùng làm danh sách ứng viên để xếp hạng và Clicked Article IDs (Các bài được click vào trong danh sách trên) dùng làm ground truth.

2.2 History

Log lịch sử click của từng người dùng. Chứa lịch sử các bài báo mà người dùng đã click vào, cùng thời gian đọc, phần trăm cuộn chuột v.v.

Đây là tập dữ liệu dài hạn, ghi nhận thói quen và sở thích của người dùng để làm input cho mô hình gợi ý.

#	Column	Context	Example	dtype
1	User ID	The anonymized user ID.	44038	u32
2	Article IDs	The articles clicked by the user.	[9618533, ... 9646154]	list[i32]
3	Timestamps	The timestamps of when the articles were clicked. The format is "YYYY/MM/DD HH:MM:SS".	[2023-02-02 16:37:42, ... 2023-02-22 18:28:38]	list[datetime[us]]
4	Read times	The read times of the clicked articles.	[425.0, ... 12.0]	list[f32]
5	Scroll Percentages	The scroll percentages of the clicked articles.	[null, ... 100.0]	list[f32]

2.3 Articles

Là metadata của từng bài báo, đóng vai trò quan trọng trong việc biểu diễn bài viết để làm input cho mô hình gợi ý.

Mỗi dòng là một bài báo duy nhất. Chứa thông tin nội dung, chuyên mục, loại bài, và cả các thông tin thống kê như lượt xem, thời gian đọc, sentiment,... Là nguồn để sinh ra article embedding hoặc feature input trong hệ thống recommendation.

2.4 Artifacts

Dữ liệu được xử lý sẵn, file này cung cấp các vector embedding của bài báo dựa trên nội dung (tiêu đề, thân, ảnh,...). Được trích xuất sẵn bằng 3 mô hình (BERT, RoBERTa, contrastive).

Phù hợp để làm recommendation content-based hoặc hybrid. Giúp tiết kiệm công đoạn tiền xử lý, đặc biệt là NLP text embedding.

Phần 3: RecGPT

1. Nội dung Paper

1.1 Đặt vấn đề

Các hệ gợi ý truyền thống dựa trên embedding theo ID item không thể khai quát hóa với dữ liệu thuộc miền mới hoặc giải quyết tốt bài toán cold-start, vì mỗi item hoặc user mới đều thiếu lịch sử tương tác để học embedding riêng.

→ Bài báo đặt câu hỏi liệu có thể phát triển một Foundation Model cho Sequential Recommendation có cơ chế zero-shot để chuyển giao kiến thức giữa các miền (sản phẩm) khác nhau, tương tự như các LLM trong NLP?

- ❶ Foundation Model: Là một mô hình học sâu được pre-train trên tập dữ liệu rất lớn và đa dạng.
- ❷ Sequential Recommendation: Là dạng bài toán gợi ý dựa trên thứ tự các tương tác của người dùng theo thời gian, với mục tiêu dự đoán item tiếp theo trong chuỗi đó.
- ❸ Zero-shot: Là khả năng của một mô hình giải quyết bài toán trên một miền dữ liệu mới mà trong quá trình huấn luyện nó chưa từng thấy dữ liệu của miền đó mà không cần thêm bước hiệu chỉnh nào (fine-tuning).

1.2 Cách hoạt động

- Pre-train trên một tập dữ liệu không lò để học biểu diễn chung của các item và tương tác.
- Với mỗi item (kể cả mới), lấy phần mô tả đưa vào MPNet để có vector liên tục.
- Dùng FSQ chia vector đó thành K sub-vector rồi lượng tử hóa thành token rời rạc dựa trên codebook chung.
- Bên cạnh token rời rạc, RecGPT vẫn giữ vector liên tục gốc như một luồng thông tin phụ trợ, bù đắp phần thông tin bị mất khi lượng tử hóa.
- Xâu token của các items trong lịch sử user được đưa vào một Transformer chuyên dụng. Transformer này đặc thù để học quy tắc chuyển từ chuỗi item trước sang item kế tiếp.

- Khi sinh dự đoán, RecGPT dùng beam search song song để đưa ra cả K token của item tiếp theo, và chỉ cho phép các tổ hợp token khớp với catalog hiện có.

Cold-Start

- + Với user mới: mô hình vẫn nhận được chuỗi token semantic từ số ít item đầu, kết hợp kiến thức của pre-training để sinh tiếp item nối sau.
- + Với item mới: nhờ token hóa dựa vào text, không cần fine-tune để học embedding mới.

1.3 Dữ liệu thử nghiệm

Dữ liệu pre-training: Tập hợp 11 nhóm sản phẩm Amazon khác nhau (tổng ~131M tương tác).

Zero-shot evaluation: 3 nhóm sản phẩm e-commerce mới (Baby, Games, Office) và 3 nền tảng khác (Yelp, Washington Maps, Steam).

Cold-start simulation: rút ngắn lịch sử user chỉ còn 1–3 item.

1.4 Kết quả và ứng dụng

- Cross-domain Zero-shot: RecGPT vượt trội so với các baseline được fine-tuned 10% dữ liệu nhóm đích, với Hit@k và NDCG@k cao hơn đáng kể trên cả 6 tập kiểm thử.
- Cold-start: RecGPT đạt hiệu năng vượt trội khi user chỉ có 1–3 lịch sử tương tác, nhờ khả năng tận dụng dữ liệu pre-training lớn.
- Hiệu năng zero-shot và cold-start tăng khi tăng lượng dữ liệu pre-training, với bước nhảy vọt giữa 10% và 25% dữ liệu
- So sánh với các mô hình hiện đại khác: RecGPT chiếm ưu thế so với S3-Rec, UniSRec, VQ-Rec, TIGER, RecFormer, IDGenRec cả về Hit@10 và NDCG@10 trên ba miền thử nghiệm.

- Triển khai thực tế: RecGPT đã được tích hợp vào hệ thống gợi ý sản xuất, phục vụ hàng triệu user mỗi ngày, đáp ứng yêu cầu latency và throughput nghiêm ngặt, chứng minh tính khả thi trong công nghiệp của mô hình.

2. Thủ nghiệm

(Można thử xem source code của họ trên github và tìm tải dataset nhưng tổng lượng data là khoảng 50GB nên em chưa dám tải hết về máy để chạy thử. Em đã hiểu khái quát những gì mà họ đã sử dụng và cơ chế hoạt động của mô hình nhưng code thì em chưa thể nắm bắt rõ được nên chưa thể thử nghiệm được a.

Nguồn tham khảo:

[1] *The ABCs of Learning to Rank*

<https://lucidworks.com/blog/abcs-learning-to-rank/>

[2] *RecSys – Ekstra Bladet Dataset*

<https://recsys.eb.dk/dataset/>

[3] *RecGPT: A Foundation Model for Sequential Recommendation*

<https://arxiv.org/abs/2506.06270>

[4] *RecGPT Source Code*

<https://github.com/HKUDS/RecGPT>