

BÀI TẬP VỀ NHÀ

KỸ THUẬT XẾP HẠNG

1. Pointwise Approach

- **Khái niệm:** Phương pháp này coi bài toán xếp hạng như một bài toán hồi quy hoặc phân loại. Mỗi tài liệu (item) được đánh giá độc lập để dự đoán điểm số liên quan (relevance score) hoặc xác suất liên quan của nó với truy vấn (query).
- **Cách hoạt động:**
 - Đầu vào: Một tài liệu và truy vấn.
 - Đầu ra: Một điểm số (score) hoặc nhãn (label) cho tài liệu đó.
 - Mô hình học cách dự đoán điểm số này dựa trên các đặc trưng của tài liệu và truy vấn.
 - Ví dụ: Dự đoán điểm liên quan từ 0 đến 5 hoặc xác suất tài liệu phù hợp.
- **Ưu điểm:**
 - Đơn giản, dễ triển khai.
 - Có thể sử dụng các thuật toán hồi quy/phân loại có sẵn (như SVM, Random Forest).
- **Nhược điểm:**
 - Không xem xét mối quan hệ tương đối giữa các tài liệu.
 - Chỉ tập trung vào điểm số của từng tài liệu, không tối ưu hóa thứ tự xếp hạng tổng thể.
- **Ứng dụng:** Thường được sử dụng trong các hệ thống đơn giản hoặc khi không cần tối ưu hóa thứ tự chính xác.

2. Pairwise Approach

- **Khái niệm:** Phương pháp này tập trung vào việc so sánh cặp tài liệu để xác định thứ tự tương đối (nào tốt hơn). Bài toán được chuyển thành phân loại nhị phân, dự đoán tài liệu nào trong cặp nên xếp cao hơn.
- **Cách hoạt động:**
 - Đầu vào: Một cặp tài liệu và truy vấn.
 - Đầu ra: Dự đoán xem tài liệu nào trong cặp liên quan hơn (ví dụ: tài liệu A > tài liệu B).
 - Mô hình học cách tối ưu hóa thứ tự tương đối của các cặp tài liệu.
 - Ví dụ thuật toán: RankSVM, RankNet, LambdaMART.
- **Ưu điểm:**
 - Tốt hơn Pointwise vì xem xét mối quan hệ tương đối giữa các tài liệu.

- Phù hợp với các bài toán xếp hạng thực tế, nơi thứ tự quan trọng hơn điểm số tuyệt đối.
- **Nhược điểm:**
 - Không tối ưu hóa trực tiếp toàn bộ danh sách xếp hạng.
 - Có thể gặp vấn đề khi số lượng cặp tài liệu lớn, dẫn đến chi phí tính toán cao.
- **Ứng dụng:** Thường được dùng trong các hệ thống tìm kiếm hoặc đề xuất, ví dụ như đề xuất tin tức (theo nguồn Amazon Science).

3. Listwise Approach

- **Khái niệm:** Phương pháp này xem xét toàn bộ danh sách tài liệu cùng lúc và tối ưu hóa trực tiếp thứ tự xếp hạng của danh sách đó. Nó tập trung vào việc tối ưu hóa một hàm mất mát (loss function) dựa trên toàn bộ danh sách.
- **Cách hoạt động:**
 - Đầu vào: Một danh sách tài liệu và truy vấn.
 - Đầu ra: Một thứ tự xếp hạng tối ưu cho toàn bộ danh sách.
 - Mô hình sử dụng các hàm mất mát như NDCG hoặc MAP (Mean Average Precision) để đánh giá và tối ưu hóa thứ tự.
 - Ví dụ thuật toán: LambdaRank, ListNet, SoftRank.
- **Ưu điểm:**
 - Tối ưu hóa trực tiếp chất lượng của toàn bộ danh sách xếp hạng.
 - Phù hợp với các bài toán yêu cầu thứ tự chính xác, như tìm kiếm hoặc đề xuất phức tạp.
- **Nhược điểm:**
 - Phức tạp hơn, khó triển khai và yêu cầu tính toán lớn hơn.
 - Cần dữ liệu huấn luyện chất lượng cao với thứ tự xếp hạng rõ ràng.
- **Ứng dụng:** Được sử dụng trong các hệ thống tìm kiếm nâng cao, như Google Search hoặc các hệ thống đề xuất phức tạp.

4. So sánh Pointwise, Pairwise, Listwise

Tiêu chí	Pointwise	Pairwise	Listwise
Đầu vào	Từng tài liệu đơn lẻ	Cặp tài liệu	Toàn bộ danh sách tài liệu
Mục tiêu	Dự đoán điểm số liên quan	Dự đoán thứ tự tương đối của cặp	Tối ưu hóa thứ tự toàn danh sách
Độ phức tạp	Thấp	Trung bình	Cao

Hiệu quả xếp hạng	Kém hơn (không xét thứ tự)	Tốt hơn (xét cặp)	Tốt nhất (xét toàn bộ danh sách)
Ví dụ thuật toán	Logistic Regression, Random Forest	RankSVM, RankNet, LambdaMART	ListNet, LambdaRank, SoftRank

Tài liệu tham khảo:

1. OpinionX. (2023, August 29). *Pairwise comparison: Definition, importance, and real-world applications*. OpinionX Blog. <https://www.opinionx.co/blog/pairwise-comparison>
2. Bhangdia, N. (2023, August 16). *Pointwise vs. pairwise vs. listwise learning to rank*. Medium. <https://medium.com/@nikhilbd/pointwise-vs-pairwise-vs-listwise-learning-to-rank-80a8fe8faddf>
3. Sumit. (2023, December 7). *Prompting LLMs for ranking tasks*. ReachSumit Blog. <https://blog.reachsumit.com/posts/2023/12/prompting-llm-for-ranking/>