**APRIORI ALGORITHM**

1. **Giới thiệu**

Thuật toán Apriori là một thuật toán để khai thác các tập hạng mục phổ biến và các luật (quy tắc) kết hợp trong một tập dữ liệu. Thuật toán hoạt động bằng cách xác định các hạng mục hoặc tập hạng mục phổ biến nhất trong tập dữ liệu, sau đó sử dụng những mục phổ biến đó để tạo ra các quy tắc liên kết mô tả mối quan hệ giữa chúng. Nó dựa trên ý tưởng rằng nếu một tập hạng mục là phổ biến, thì tất cả các tập con của nó cũng là phổ biến.

1. **Cách thức hoạt động**

Thuật toán Apriori thực hiện thông qua 2 bước chính: *tạo tập hạng mục phổ biến* và *tạo ra các quy tắc liên kết*.

**Bước 1: Tạo tập phổ biến**

Thuật toán Apriori tạo ra tập hạng mục phổ biến bằng cách lặp lại quá trình quét tập dữ liệu để đếm số lần xuất hiện của các tập hạng mục ứng cử và lọc bỏ các tập mục không đáp ứng ngưỡng hỗ trợ tối thiểu (minimum support threshold). Ngưỡng hỗ trợ tối thiểu là một tham số do người dùng xác định, chỉ định số lần mà một tập mục phải xuất hiện tối thiểu trong tập dữ liệu để được coi là "phổ biến".

Thuật toán Apriori sử dụng phương pháp "bottom-up" (đi ngược từ dưới lên) để tạo ra các tập mục phổ biến có kích thước ngày càng lớn. Bằng cách tạo ra các tập mục phổ biến có kích thước 1 (tức là các hạng mục riêng lẻ), sau đó sử dụng những tập hạng mục phổ biến đó để tạo ra các tập hạng mục ứng cử có kích thước 2 bằng cách kết hợp các cặp tập mục phổ biến. Sau đó, nó đếm số lần xuất hiện của mỗi tập hạng mục ứng cử trong tập dữ liệu, loại bỏ các tập mục không đáp ứng ngưỡng hỗ trợ tối thiểu và tạo ra các tập mục phổ biến có kích thước 2. Quá trình này được lặp lại khiến cho các tập mục có kích thước ngày càng lớn và dừng lại khi không còn tập phổ biến nào được tìm thấy.

Bước 2: tạo ra các quy tắc liên kết.

Sau khi tạo ra tập hạng mục phổ biến, Apriori tạo ra các quy tắc liên kết mô tả mối quan hệ giữa các mục trong các tập hạng mục phổ biến. Một quy tắc liên kết có dạng “A => B” (với A, B là các tập hạng mục), nghĩa là khi A xuất hiện thì B cũng xuất hiện. Quy tắc liên kết được đánh giá bằng độ tin cậy, tức tỉ lệ của các trường hợp A và B cùng xuất hiện trong tập dữ liệu, nó mô tả một mối quan hệ giữa các mục trong tập dữ liệu và được sử dụng để dự đoán các sự kiện mới dựa trên các sự kiện đã biết.

Thuật toán Apriori chỉ giữ lại những quy tắc liên kết có độ tin cậy cao hơn ngưỡng do người dùng xác định. Ngưỡng độ tin cậy tối thiểu chỉ định tỉ lệ tối thiểu của các trường hợp trong tập dữ liệu mà A và B đồng thời xuất hiện để quy tắc liên kết "A => B" được coi là tin cậy.

1. Ví dụ

Một ví dụ về thuật toán Apriori là *Recommendation System*. Ví dụ sau đây sẽ mô tả về hệ thống đề xuất phim

Giả sử đây là dữ liệu của những người dùng thích một số bộ phim:

|  |  |
| --- | --- |
| **Người dùng** | **Phim** |
| A | Titanic, Avatar, Star Wars |
| B | Titanic, Avatar |
| C | Star Wars, Harry Potter |
| D | Titanic, Harry Potter |

Giả sử ngưỡng hỗ trợ tối thiểu là 50% và ngưỡng tin cậy tối thiểu là 60%. Thuật toán Apriori sẽ hoạt động như sau:

* Bước 1: Tìm các hạng mục thường xuyên có một phần tử (tập hạng mục có kích thước 1) bằng cách đếm số lần xuất hiện của mỗi phim trong tập dữ liệu và so sánh với ngưỡng hỗ trợ.

|  |  |  |
| --- | --- | --- |
| **Phim** | **Số lần xuất hiện** | **Ngưỡng Hỗ trợ (%)** |
| Titanic | 3/4 = 0.75 | 75% |
| Avatar | 2/4 = 0.5 | 50% |
| Star Wars | 2/4 = 0.5 | 50% |
| Harry Potter | 2/4 = 0.5 | 50% |

Các hạng mục thường xuyên riêng lẻ {Titanic}, {Avatar}, {Star Wars} và {Harry Potter}.

Bước 2: Tạo các tập hạng mục có kích thước 2 (hai phần tử) bằng cách kết hợp các tập hạng mục có kích thước một.

{Titanic}, {Avatar} -> {Titanic, Avatar}

{Titanic}, {Star Wars} -> {Titanic, Star Wars}

{Titanic}, {Harry Potter} -> {Titanic, Harry Potter}

{Avatar}, {Star Wars} -> {Avatar, Star Wars}

{Avatar}, {Harry Potter} -> {Avatar, Harry Potter}

{Star Wars}, {Harry Potter} -> {Star Wars, Harry Potter}

Bước 3: Lọc bỏ các tập hạng mục có kích thước 2 bằng cách loại bỏ những tập nào có ít nhất một tập con không phổ biến. Trong trường hợp này, không tập mục nào bị loại bỏ.

Bước 4: Tìm các hạng mục thường xuyên có hai phần tử bằng cách đếm số lần xuất hiện của từng tập trong tập dữ liệu và so sánh với ngưỡng hỗ trợ.

|  |  |  |
| --- | --- | --- |
| **Tập hai phần tử** | **Số lần xuất hiện** | **Ngưỡng Hỗ trợ (%)** |
| Titanic, Avatar | 2/4 = 0.5 | 50% |
| Titanic, Star Wars | 1/4 = 0.25 |  |

1. So sánh

* Thuật toán Apriori và thuật toán FP-Growth là hai thuật toán khai phá dữ liệu phổ biến để tìm các tập mục phổ biến và các quy tắc liên kết.
* Thuật toán FP-growth là một phương pháp khác để khai thác luật kết hợp trong dữ liệu. Tương tự như thuật toán Apriori, thuật toán FP-growth cũng dựa trên việc phân tích tập dữ liệu để tìm các tập hợp mục phổ biến và các quy tắc liên kết giữa chúng.
* Tuy nhiên, chúng có một số khác biệt về cách thức hoạt động và hiệu suất.
* [Thuật toán Apriori là một thuật toán dựa trên kỹ thuật nối (Join-Based), trong khi thuật toán FP-Growth là một thuật toán dựa trên kỹ thuật cây (Tree-Based)](https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/).
* [Thuật toán Apriori sử dụng kỹ thuật sinh ứng viên (Candidate Generation), trong đó các tập mục phổ biến được mở rộng thêm một mục tại một thời điểm](https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/). [Thuật toán FP-Growth sử dụng kỹ thuật sinh cây FP (FP-Tree) để lưu trữ các tập mục phổ biến](https://www.mygreatlearning.com/blog/understanding-fp-growth-algorithm/).
* [Thuật toán Apriori cần quét cơ sở dữ liệu nhiều lần trong từng bước của nó, do đó gây ra chi phí tính toán cao khi số lượng các mục lớn**2**](https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/). [Thuật toán FP-Growth chỉ cần quét cơ sở dữ liệu hai lần để xây dựng cây FP và sinh ra các cây FP có điều kiện (Conditional FP-Tree) cho từng mục trong dữ liệu](https://www.mygreatlearning.com/blog/understanding-fp-growth-algorithm/).
* [Thuật toán Apriori có thể chạy nhanh hơn thuật toán FP-Growth trong trường hợp không có hoặc ít có các tập mục thỏa mãn ngưỡng hỗ trợ tối thiểu, hoặc khi độ dài của các tập mục phổ biến chỉ là 2](https://stackoverflow.com/questions/56651242/why-does-apriori-run-faster-than-fp-growth-in-this-implementation). [Thuật toán FP-Growth có thể chạy nhanh hơn thuật toán Apriori khi ngưỡng hỗ trợ tối thiểu thấp và độ dài của các tập mục phổ biến lớn**5**](https://www.researchgate.net/publication/259042315_Performance_Comparison_of_Apriori_and_FP-Growth_Algorithms_in_Generating_Association_Rules).

Do đó, việc lựa chọn giữa hai thuật toán này phụ thuộc vào bản chất của dữ liệu và những giả định về các tập mục phổ biến.

1. Chủ đề nâng cao

Một trong các chủ đề nâng cao được nhắc đến khi nói về Apriori là *Multiple minimum support thresholds* (Nhiều ngưỡng hỗ trợ tối thiểu).

Trong thuật toán Apriori, một ngưỡng hỗ trợ tối thiểu được sử dụng để xác định tập phổ biến. Tuy nhiên, trong một số trường hợp, sử dụng nhiều ngưỡng hỗ trợ tối thiểu để xác định các tập phổ biến ở các cấp độ khác nhau có thể hữu ích. Ví dụ, trong phân tích giỏ hàng, một nhà bán lẻ có thể muốn xác định các tập phổ biến cho các loại sản phẩm khác nhau, chẳng hạn như thực phẩm, quần áo và điện tử. Bằng cách sử dụng nhiều ngưỡng hỗ trợ tối thiểu, nhà bán lẻ có thể xác định các tập phổ biến cho mỗi danh mục một cách riêng biệt.

1. Tổng kết

* Trong bài báo cáo này, chúng ta đã tìm hiểu về thuật toán Apriori và cách áp dụng nó để khai thác quy luật kết hợp trong các tập dữ liệu lớn.
* Thuật toán Apriori là một thuật toán khai phá dữ liệu nổi tiếng và được áp dụng trong nhiều lĩnh vực khác nhau, chẳng hạn như học máy, khai phá web, phân tích dữ liệu thương mại điện tử, phân tích dữ liệu y tế và nhiều lĩnh vực khác.
* Tóm lại, thuật toán Apriori là một cách hiệu quả để khai thác các tập phổ biến và tạo các luật kết hợp từ các tập dữ liệu lớn. Tuy nhiên, nó có thể tốn kém về mặt tính toán đối với các bộ dữ liệu rất lớn hoặc khi ngưỡng hỗ trợ tối thiểu được đặt quá thấp.

Reference:

[Apriori Algorithm - GeeksforGeeks](https://www.geeksforgeeks.org/apriori-algorithm/)

[Apriori algorithm - Wikipedia](https://en.wikipedia.org/wiki/Apriori_algorithm)

[Apriori Algorithm in Data Mining: Implementation With Examples (softwaretestinghelp.com)](https://www.softwaretestinghelp.com/apriori-algorithm/)

[What is Apriori Algorithm With Example? Simplest Explanation! (mltut.com)](https://www.mltut.com/what-is-apriori-algorithm-with-example/)

[Apriori vs FP-Growth in Market Basket Analysis - A Comparative Guide - (analyticsindiamag.com)](https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/)

[Movie Recommendation With ML - Apriori Explained | Kaggle](https://www.kaggle.com/code/ankits29/movie-recommendation-with-ml-apriori-explained)

[mark-lvl/recommendation-engine: Using Apriori algorithm for affinity analysis on movie dataset and extracting association rules as movie recommendation (github.com)](https://github.com/mark-lvl/recommendation-engine)