ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Fast Algorithms for Mining Association Rules

Chuyên đề 4

Ngành: **Khoa học dữ liệu**

(Chương trình đào tạo thạc sĩ)

Giảng viên hướng dẫn: PGS.TS Vũ Tiến Dũng

Nguyễn Thị Ngọc Uyên - 23007930
Lý Đức Trung - 23007933

Hà Nội - 2025

LỜI CẢM ƠN

Bài tiểu luận cuối kỳ này được hoàn thành dưới sự hướng dẫn của thầy PGS.TS. Vũ Tiến Dũng. Chúng em xin bày tỏ lòng biết ơn sâu sắc tới thầy vì sự định hướng và gợi ý đề tài trong quá trình học tập môn học này.

Trong quá trình làm bài tiểu luận này, do kiến thức còn hạn chế nên chúng em không tránh khỏi những thiếu sót. Chúng em mong nhận được những ý kiến đóng góp của thầy và các bạn để có thể rút ra kinh nghiệm, hoàn thiện và phát triển đề tài hơn.

Chúng em xin chân thành cảm ơn!

Mục lục

Giới thiệu

Trong bối cảnh các hệ thống thông tin hiện đại đang tạo ra khối lượng dữ liệu khổng lồ mỗi ngày, việc khai thác tri thức từ dữ liệu (data mining) đã trở thành một lĩnh vực nghiên cứu quan trọng và có nhiều ứng dụng thực tiễn. Trong số các kỹ thuật khai phá dữ liệu, khai phá luật kết hợp (association rule mining) là một trong những hướng tiếp cận được quan tâm nhiều nhất bởi khả năng phát hiện các mối quan hệ tiềm ẩn giữa các đối tượng dữ liệu. Những luật kết hợp này không chỉ hỗ trợ mô tả hành vi của người dùng và khách hàng, mà còn mang lại giá trị lớn trong các lĩnh vực như thương mại điện tử, tối ưu hoá bán lẻ, marketing, phân tích rủi ro tài chính và các hệ thống gợi ý thông minh.

Một bài toán trung tâm trong khai phá luật kết hợp là bài toán tìm tập mục thường xuyên (frequent itemset mining). Đây là bước tiền đề quan trọng để từ đó sinh ra các luật kết hợp có ý nghĩa. Tuy nhiên, do không gian tìm kiếm tăng theo cấp số mũ với số lượng mặt hàng và giao dịch, việc khai thác các tập mục thường xuyên đòi hỏi những thuật toán hiệu quả về thời gian và tài nguyên xử lý. Trong lĩnh vực này, ba thuật toán kinh điển và nền tảng được sử dụng rộng rãi nhất là Apriori, AprioriTid và AprioriHybrid. Các thuật toán này không chỉ đặt nền móng cho các phương pháp hiện đại mà còn thể hiện những chiến lược tối ưu hóa quan trọng nhằm giảm số lần quét cơ sở dữ liệu, giảm số lượng ứng viên cần xét và tận dụng tốt hơn tài nguyên bộ nhớ.

Thuật toán **Apriori** giới thiệu một thuộc tính quan trọng cho phép giảm mạnh số lượng tập ứng viên, dựa trên nguyên tắc rằng một tập mục chỉ có thể là phổ biến khi tất cả các tập con của nó cũng phổ biến. Thuật toán **AprioriTid** cải thiện hiệu năng ở các vòng lặp sau bằng cách sử dụng một cấu trúc dữ liệu trung gian, tránh việc phải quét lại toàn bộ cơ sở dữ liệu. Trong khi đó, **AprioriHybrid** kết hợp ưu điểm của cả Apriori và AprioriTid, nhờ đó đạt được hiệu suất tối ưu hơn trong các tình huống thực tế.

Việc nghiên cứu ba thuật toán này không chỉ giúp làm rõ các nguyên lý nền tảng của khai

phá luật kết hợp, mà còn cho thấy cách những chiến lược tối ưu hóa khác nhau ảnh hưởng trực tiếp đến hiệu năng thực thi của thuật toán.

Tiểu luận này được thực hiện nhằm mục tiêu:

- Trình bày tổng quan về khai phá dữ liệu và khai phá luật kết hợp.
- Phân tích chi tiết ba thuật toán Apriori, AprioriTid và AprioriHybrid.
- Minh họa hoạt động của các thuật toán thông qua ví dụ cụ thể.
- Đánh giá và so sánh hiệu năng của các thuật toán trên các tiêu chí lý thuyết.
- Rút ra những nhận xét và định hướng áp dụng phù hợp trong thực tế.

Phạm vi tiểu luận tập trung vào khía cạnh thuật toán và hiệu năng, không đi sâu vào các mở rộng hoặc biến thể hiện đại như FP-Growth, ECLAT hoặc các phương pháp dựa trên cây tiền tố và đồ thị. Tuy nhiên, những thuật toán cổ điển được phân tích trong báo cáo này vẫn giữ vai trò quan trọng trong việc hiểu rõ tư duy thiết kế thuật toán khai phá dữ liệu.

Bố cục của tiểu luận được tổ chức như sau: Chương 1 giới thiệu tổng quan về khai phá dữ liệu và luật kết hợp. Chương 2, Chương 3 và Chương 4 lần lượt phân tích ba thuật toán Apriori, AprioriTid và AprioriHybrid. Chương 5 so sánh hiệu năng các thuật toán và thảo luận các yếu tố ảnh hưởng. Cuối cùng là phần kết luận tổng hợp nội dung và đề xuất hướng phát triển tiếp theo.

Chương 1

Tổng quan về khai phá dữ liệu và luật kết hợp

1.1 Tổng quan về khai phá dữ liệu

1.1.1 Khái niệm khai phá dữ liệu

Khai phá dữ liệu (Data Mining) là quá trình phát hiện các mẫu, quy luật hoặc tri thức tiềm ẩn trong các tập dữ liệu lớn, phức tạp và thường khó phân tích bằng các phương pháp thống kê truyền thống. Đây là một giai đoạn quan trọng trong quy trình khám phá tri thức từ dữ liệu (Knowledge Discovery in Databases – KDD), bao gồm các bước như làm sạch dữ liệu, tích hợp, lựa chọn dữ liệu, khai phá và đánh giá tri thức. Khai phá dữ liệu kết hợp nhiều kỹ thuật từ thống kê, học máy, hệ cơ sở dữ liệu và trí tuệ nhân tạo nhằm rút ra thông tin hữu ích phục vụ cho việc ra quyết định.

1.1.2 Vai trò của khai phá dữ liệu

Trong thời đại dữ liệu lớn (Big Data), các tổ chức và doanh nghiệp thu thập dữ liệu từ nhiều nguồn khác nhau như giao dịch thương mại điện tử, mạng xã hội, hệ thống cảm biến hay dữ liệu tài chính. Việc phân tích dữ liệu này mang lại nhiều giá trị:

- Hỗ trợ ra quyết định chiến lược dựa trên bằng chứng.
- Dự đoán hành vi khách hàng hoặc xu hướng thị trường.
- Tối ưu hóa hoạt động kinh doanh và giảm thiểu rủi ro.

- Phát hiện các bất thường, gian lận hoặc rủi ro tiềm ẩn.

Nhờ những lợi ích này, khai phá dữ liệu trở thành một trong những lĩnh vực cốt lõi của khoa học dữ liệu và công nghệ thông tin.

1.1.3 Các nhiệm vụ chính trong khai phá dữ liệu

Khai phá dữ liệu bao gồm nhiều nhóm nhiệm vụ khác nhau, trong đó phổ biến nhất là:

- **Phân lớp (Classification):** Dự đoán nhãn lớp cho dữ liệu mới dựa trên mô hình học từ dữ liệu quá khứ.
- **Phân cụm (Clustering):** Nhóm các đối tượng tương đồng lại với nhau dựa trên khoảng cách hoặc đặc trưng.
- **Phát hiện bất thường (Anomaly Detection):** Tìm ra các điểm dữ liệu bất thường trong tập hợp lớn.
- **Khai phá luật kết hợp (Association Rule Mining):** Tìm các mẫu quan hệ dạng "nếu–thì" giữa các mục xuất hiện trong dữ liệu.

Trong tiểu luận này, chúng ta tập trung vào nhiệm vụ khai phá luật kết hợp, một phương pháp quan trọng và có nhiều ứng dụng trong thực tế.

1.2 Khai phá luật kết hợp

1.2.1 Mục đích của luật kết hợp

Khai phá luật kết hợp nhằm tìm ra những quan hệ thường xuyên xuất hiện giữa các đối tượng trong dữ liệu. Một luật kết hợp có dạng:

$$X \Rightarrow Y$$

trong đó X và Y là hai tập mục không giao nhau. Luật được hiểu là: khi các mục trong X xuất hiện trong một giao dịch, thì các mục trong Y cũng có khả năng xuất hiện theo.

Mục tiêu chính của khai phá luật kết hợp là phát hiện các mẫu có ý nghĩa, giúp mô tả hành vi người dùng hoặc xu hướng đồng xuất hiện trong dữ liệu.

1.2.2 Ứng dụng trong thực tế

Luật kết hợp được sử dụng rộng rãi trong nhiều lĩnh vực:

- **Bán lẻ và thương mại điện tử:** gợi ý sản phẩm, phân tích giỏ hàng, bố trí hàng hóa.
- **Ngân hàng và tài chính:** phát hiện hành vi gian lận hoặc các mẫu giao dịch bất thường.
- **Marketing:** phân tích chiến dịch quảng cáo, xác định nhóm khách hàng tiềm năng.
- **Y tế:** phát hiện mối liên hệ giữa triệu chứng và bệnh lý.

Nhờ khả năng diễn giải trực quan, luật kết hợp đặc biệt hữu ích trong các hệ thống gợi ý và phân tích hành vi tiêu dùng.

1.2.3 Ý nghĩa trong hệ thống gợi ý

Trong thương mại điện tử, luật kết hợp đóng vai trò quan trọng trong việc xây dựng các mô hình gợi ý kiểu "Frequently Bought Together". Các luật như:

$$\text{(Laptop)} \Rightarrow \text{(Chuột không dây)}$$

giúp hệ thống tự động đề xuất sản phẩm phù hợp, gia tăng doanh thu và cải thiện trải nghiệm người dùng.

1.3 Mô hình dữ liệu giỏ hàng (Basket Data)

1.3.1 Khái niệm và cấu trúc dữ liệu

Dữ liệu giỏ hàng (Basket Data) là mô hình dữ liệu phổ biến trong phân tích giao dịch, trong đó mỗi giao dịch là một tập các mục được mua cùng nhau. Ký hiệu:

$$D = \{T_1, T_2, ..., T_m\}, \quad T_i \subseteq I$$

với I là tập tất cả các mặt hàng.

1.3.2 Đặc trưng của dữ liệu giỏ hàng

Dữ liệu giỏ hàng có các đặc điểm:

- Dữ liệu thường rất lớn và thưa.
- Mỗi giao dịch chứa ít mục so với kích thước toàn bộ tập mục.
- Các mục có xu hướng đồng xuất hiện theo nhóm.

Những đặc điểm này ảnh hưởng trực tiếp đến thiết kế thuật toán tìm tập mục thường xuyên.

1.3.3 Ví dụ minh hoạ dữ liệu giao dịch

Một ví dụ đơn giản về dữ liệu giỏ hàng:

TID	Mục hàng
1	Bread, Milk, Butter
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Cola
4	Bread, Milk, Diaper, Beer

Ví dụ này cho thấy các mục như *Bread*, *Milk*, *Diaper* thường xuất hiện cùng nhau, là cơ sở để sinh ra các luật kết hợp.

1.4 Các khái niệm cơ bản

1.4.1 Độ hỗ trợ (Support)

Độ hỗ trợ phản ánh mức độ phổ biến của luật trong toàn bộ cơ sở dữ liệu giao dịch. Nó được định nghĩa là tỷ lệ phần trăm các giao dịch chứa đồng thời cả X và Y:

$$\text{support}(X \Rightarrow Y) = \frac{\text{count}(X \cup Y)}{|D|},$$

với $|D|$ là tổng số giao dịch.

Độ hỗ trợ càng cao nghĩa là luật xuất hiện thường xuyên trong dữ liệu, do đó có giá trị thực tiễn lớn hơn. Với các tập dữ liệu rất lớn, tiêu chí này giúp loại bỏ những luật chỉ xuất hiện hiếm hoi hoặc mang tính bất thường.

1.4.2 Độ tin cậy (Confidence)

Độ tin cậy là thước đo phản ánh mức độ chính xác của luật kết hợp. Nó cho biết trong số các giao dịch có chứa toàn bộ các mặt hàng trong X, có bao nhiêu phần trăm giao dịch đồng

thời chứa các mặt hàng trong Y. Độ tin cậy của luật $X \Rightarrow Y$ được định nghĩa:

$$\text{confidence}(X \Rightarrow Y) = \frac{\text{count}(X \cup Y)}{\text{count}(X)},$$

trong đó $\text{count}(Z)$ là số lượng giao dịch chứa tập Z.

Độ tin cậy cao thể hiện rằng mối quan hệ giữa X và Y là mạnh và đáng tin cậy. Đây là một trong hai tiêu chí quan trọng nhất để đánh giá chất lượng của một luật kết hợp.

1.4.3 Mục tiêu của khai phá luật kết hợp

Khai phá luật kết hợp hướng đến việc tìm ra tất cả các luật có ý nghĩa, đồng thời đảm bảo luật đủ phổ biến và đủ chính xác. Vì vậy, hai ngưỡng quan trọng được đặt ra trong quá trình này là:

- `minsup`: độ hỗ trợ tối thiểu,
- `minconf`: độ tin cậy tối thiểu.

Một luật chỉ được giữ lại nếu thỏa mãn cả hai ngưỡng trên. Điều này giúp giảm đáng kể số lượng luật không thực sự hữu ích, đồng thời tập trung vào những luật phản ánh hành vi phổ biến của người dùng.

1.5 Các bài toán con trong khai phá luật kết hợp

Bài toán khai phá luật kết hợp thường được chia thành hai phần độc lập:

1. **Tìm tất cả các tập mặt hàng có độ hỗ trợ không nhỏ hơn minsup** Các tập này được gọi là *tập mục lớn* (large itemsets) hay *tập mục thường xuyên*. Việc tìm được các tập này là bước quan trọng nhất, bởi toàn bộ luật kết hợp về sau đều được suy ra từ chúng.
2. **Sinh luật kết hợp từ các tập mục lớn** Sau khi có các tập mục lớn, ta xét các tập con của từng tập mục để tạo thành các luật ứng viên, sau đó tính độ tin cậy để lọc ra các luật đạt minconf.

Trong nghiên cứu và trong bài tiểu luận này, trọng tâm được đặt vào bài toán thứ nhất—tìm tập mục thường xuyên—vì đây là phần tốn nhiều chi phí tính toán và là động lực cho sự ra đời của các thuật toán nổi tiếng như Apriori, AprioriTid và AprioriHybrid.

Chương 2

Thuật toán Apriori

2.1 Giới thiệu

Thuật toán Apriori là một trong những phương pháp quan trọng nhất trong khai phá dữ liệu để tìm các tập mục thường xuyên (frequent itemsets) và sinh luật kết hợp. Mặc dù ra đời từ rất sớm, Apriori vẫn là nền tảng cho nhiều thuật toán hiện đại nhờ vào cơ chế sinh ứng viên theo từng mức và khả năng loại bỏ mạnh mẽ các tổ hợp không cần thiết. Ý tưởng cốt lõi của Apriori dựa trên một tính chất đặc biệt liên quan đến mối quan hệ giữa một tập và các tập con của nó. Tính chất này giúp giảm đáng kể số lượng ứng viên, khiến cho bài toán trở nên khả thi ngay cả với tập dữ liệu có kích thước lớn.

Chương này trình bày chi tiết nguyên lý, quy trình hoạt động, cách sinh ứng viên, cơ chế loại bỏ (prune), đồng thời cung cấp một ví dụ minh họa cụ thể dựa trên nội dung các slide đã cho.

2.2 Nguyên lý cốt lõi

2.2.1 Tính chất cơ bản của Apriori

Tính chất quan trọng nhất của thuật toán Apriori được phát biểu bằng lời như sau:

> Nếu một tập mục kích thước k là một tập mục phổ biến (large itemset), thì mọi tập con có kích thước k-1 của nó cũng phải là tập mục phổ biến.

Tính chất này còn được gọi là nguyên lý "đóng xuống" (downward closure). Điều này có

một hệ quả quan trọng:

- Nếu một tập con bất kỳ có kích thước k-1 không phải là tập phổ biến, thì mọi tập có kích thước k chứa tập con đó chắc chắn không phải là tập phổ biến.

Nhờ đó, thuật toán có thể loại bỏ một lượng rất lớn các ứng viên không cần thiết ngay từ đầu, thay vì phải tốn công đếm độ hỗ trợ trong cơ sở dữ liệu.

2.2.2 Ý nghĩa của nguyên lý

Dựa trên nguyên lý ở trên, Apriori có thể:

- giảm số ứng viên cần xét trong mỗi vòng lặp,
- tránh xét các tập mục có kích thước lớn khi tập con của chúng đã không đạt ngưỡng hỗ trợ tối thiểu,
- giúp việc sinh frequent itemsets theo mức (level-wise) trở nên khả thi.

Nhờ đó, mặc dù cần quét cơ sở dữ liệu nhiều lần, Apriori vẫn đạt hiệu quả tốt nhờ giảm mạnh không gian tìm kiếm.

2.3 Quy trình hoạt động của Apriori

Thuật toán Apriori hoạt động theo phương pháp tìm kiếm theo từng mức. Cụ thể:

1. Quét cơ sở dữ liệu để tìm tất cả các tập 1-itemset có độ hỗ trợ không nhỏ hơn ngưỡng tối thiểu. Đây là tập L1.
2. Ở mỗi vòng lặp tiếp theo, từ tập L(k-1), thuật toán sinh ra tập ứng viên Ck bằng cách kết hợp các phần tử trong L(k-1).
3. Quét cơ sở dữ liệu và đếm độ hỗ trợ cho từng ứng viên trong Ck.
4. Tập Lk gồm tất cả các ứng viên trong Ck có độ hỗ trợ đạt ngưỡng tối thiểu.
5. Thuật toán dừng khi Lk rỗng.

Kết quả cuối cùng là hợp của tất cả Lk được sinh ra trong quá trình thực thi.

2.4 Giả mã thuật toán

Dưới đây là giả mã tương đương với slide, được viết hoàn toàn bằng văn bản thuần:

```
1.  L1 = tập các 1-itemset phổ biến.
2.  Với k = 2; khi L(k-1) không rỗng; tăng k lên:
3.      Sinh tập ứng viên Ck từ L(k-1) bằng hàm apriori-gen.
4.      Với mỗi giao dịch t trong cơ sở dữ liệu:
5.          Xác định Ct là tập các ứng viên trong Ck xuất hiện trong t.
6.          Với mỗi ứng viên c trong Ct:
7.              Tăng bộ đếm số lần xuất hiện của c.
8.      Lk = tập các ứng viên trong Ck có độ hỗ trợ không nhỏ hơn minsup.
9.  Hợp tất cả các Lk là kết quả cần tìm.
```

2.5 Sinh tập ứng viên

Quá trình sinh Ck từ L(k-1) gồm hai bước chính: bước nối (join) và bước loại bỏ (prune).

2.5.1 Bước Join

Hai phần tử p và q trong L(k-1) được nối với nhau để tạo thành một ứng viên mới c thuộc Ck nếu:

- p và q trùng nhau ở k-2 phần tử đầu tiên,
- phần tử cuối của p có giá trị nhỏ hơn phần tử cuối của q (để tránh trùng lặp).

Mô tả join trong slide:

```
Chèn vào Ck:
Chọn p.item1, p.item2, ..., p.item(k-1), q.item(k-1)
từ L(k-1) p và L(k-1) q
với điều kiện:
  p.item1 = q.item1, ..., p.item(k-2) = q.item(k-2)
  và p.item(k-1) < q.item(k-1)
```

2.5.2 Bước Prune

Tập ứng viên được tạo ra sẽ tiếp tục được lọc nhờ tính chất Apriori. Một ứng viên c trong Ck sẽ bị loại bỏ nếu tồn tại bất kỳ tập con có kích thước k-1 của c không nằm trong L(k-1).

Mô tả prune:

```
Với mỗi itemset c trong Ck:
    Với mỗi tập con s của c có kích thước k-1:
        Nếu s không nằm trong L(k-1), loại c khỏi Ck.
```

Đây là một trong những bước giúp Apriori vận hành hiệu quả.

2.6 Ví dụ minh họa

Xét cơ sở dữ liệu giao dịch sau:

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Giả sử ngưỡng hỗ trợ tối thiểu minsup là 0.5, tương đương cần xuất hiện ít nhất 2 lần trong 4 giao dịch.

2.6.1 Bước 1: Tìm frequent 1-itemsets

Đếm số lần xuất hiện:

- Item 1: 2 lần
- Item 2: 3 lần
- Item 3: 3 lần
- Item 4: 1 lần (loại)
- Item 5: 3 lần

Do đó L1 gồm: {1}, {2}, {3}, {5}.

2.6.2 Bước 2: Sinh frequent 2-itemsets

Join L1 để tạo các ứng viên 2-itemset:

- {1,2}, {1,3}, {1,5}
- {2,3}, {2,5}
- {3,5}

Đếm số lần xuất hiện:

- {1,2}: 1 lần (loại)
- {1,3}: 2 lần
- {1,5}: 1 lần (loại)
- {2,3}: 2 lần
- {2,5}: 3 lần
- {3,5}: 2 lần

Do đó L2 gồm: {1,3}, {2,3}, {2,5}, {3,5}.

2.6.3 Bước 3: Sinh frequent 3-itemsets

Join L2 tạo ứng viên:

- {1,2,3}
- {1,2,5}
- {1,3,5}
- {2,3,5}

Prune:

- {1,2,3}: loại vì {1,2} không thuộc L2
- {1,2,5}: loại vì {1,2} không thuộc L2
- {1,3,5}: loại vì {1,5} không thuộc L2
- {2,3,5}: hợp lệ

Đếm:

{2,3,5}: xuất hiện 2 lần nên thuộc L3.

2.6.4 Kết quả

Frequent itemsets thu được:

- Kích thước 1: {1}, {2}, {3}, {5}
- Kích thước 2: {1,3}, {2,3}, {2,5}, {3,5}
- Kích thước 3: {2,3,5}

2.7 Kết luận

Thuật toán Apriori là một phương pháp kinh điển trong khai phá dữ liệu nhờ dựa trên tính chất mạnh mẽ về quan hệ giữa tập và tập con. Mặc dù phải quét dữ liệu nhiều lần, Apriori vẫn hoạt động hiệu quả trong nhiều trường hợp nhờ cơ chế sinh ứng viên theo từng mức và khả năng loại bỏ các ứng viên không cần thiết ngay từ khi sinh ra. Các thuật toán cải tiến như AprioriTid và AprioriHybrid được xây dựng dựa trên Apriori nhằm giảm số lần quét cơ sở dữ liệu và rút ngắn thời gian xử lý.

Chương 3

Thuật toán AprioriTid

Chương 4

Thuật toán AprioriHybrid

Chương 5

Đánh giá hiệu năng và so sánh các thuật toán

5.1 Các yếu tố ảnh hưởng hiệu năng

5.1.1 Số lượng giao dịch |D|

5.1.2 Số lượng mặt hàng |I|

5.1.3 Độ dài trung bình giao dịch |T|

5.1.4 Kích thước ứng viên và số vòng lặp

5.2 So sánh Apriori, AprioriTid và AprioriHybrid

5.2.1 Số lần quét CSDL

5.2.2 Tốc độ thực thi

5.2.3 Sử dụng bộ nhớ

5.2.4 Độ hiệu quả khi minsup thấp / cao

5.3 Kết quả thực nghiệm và biểu đồ minh hoạ

5.3.1 Biểu đồ số ứng viên theo vòng

5.3.2 Biểu đồ thời gian chạy ba thuật toán

5.3.3 Nhận xét kết quả

Kết luận

Tài liệu tham khảo