Chương 1

1. Tìm hiểu về HOIM (High Occupancy Itemset Mining)
   1. **Khái niệm**

**HOIM (High Occupancy Itemset Mining)** là một phương pháp khai phá dữ liệu nhằm tìm ra các **tập mục (itemset)** có mức độ **"chiếm dụng cao" (occupancy)** trong cơ sở dữ liệu.  
Không giống các thuật toán truyền thống như **Apriori** hay **FP-Growth** chỉ dựa vào tần suất xuất hiện (support), HOIM đánh giá **mức độ bao phủ** của một tập mục trong toàn bộ dữ liệu.

Mục tiêu: Khai thác các tập mục không chỉ xuất hiện thường xuyên mà còn **có mặt trong nhiều phần của các giao dịch lớn hoặc quan trọng.**

* 1. **Định nghĩa độ chiếm dụng (Itemset Occupancy – IO)**

Độ chiếm dụng đo mức độ "phủ sóng" của một tập mục X trong cơ sở dữ liệu:

Trong đó:

* : là tập giao dịch
* : là một giao dịch
* : số lượng mục trong tập X
* : số lượng mục trong giao dịch T
* : tức là tập mục X xuất hiện trong T

**Ý nghĩa**: IO càng cao → tập mục X càng “chiếm dụng” nhiều trong CSDL → có giá trị khai thác cao hơn.

* 1. Điều kiện chọn tập mục

Một tập mục X được xem là tập mục chiếm dụng cao (High-Occupancy Itemset – HOI) nếu:

Trong đó **MinIO** là ngưỡng do người dùng đặt.

* 1. **Ưu điểm của HOIM**
* Xem xét **mức độ quan trọng thực tế** của tập mục trong dữ liệu thay vì chỉ đếm số lần xuất hiện.
* Hiệu quả hơn trong các trường hợp dữ liệu **không đồng đều**, nhiều giao dịch ngắn/dài khác nhau.
* Tập trung vào các mẫu **có ý nghĩa thực tiễn hơn**, nhất là trong hệ thống thương mại hoặc giám sát.
  1. Nhược điểm của HOIM
* **Không phân biệt các giao dịch** – mọi giao dịch đều được đối xử như nhau.
* **Không tối ưu tìm kiếm**: Không có giới hạn trên (upper bound) để loại bỏ sớm các tập mục không tiềm năng.
* **Chi phí tính toán cao** với tập dữ liệu lớn do phải duyệt tất cả các giao dịch.
  1. **Ví dụ minh họa (giản lược)**

Giả sử có 3 giao dịch:

|  |  |
| --- | --- |
| **Giao dịch** | **Mục trong giao dịch** |
| T1 | A, B, C |
| T2 | A, B |
| T3 | A, D |

Tính IO của tập mục **{A, B}**:

* Xuất hiện trong T1 và T2 → 2 giao dịch
* Tổng số mục trong T1 và T2 = 3 + 2 = 5
* Tổng số mục toàn CSDL = 3 + 2 + 2 = 7

Nếu MinIO = 0.5 thì {A, B} là tập mục chiếm dụng cao.

1. **Tìm hiểu về HOIMTO (High Occupancy Itemset Mining with Transaction Occupancy)**
   1. **Khái niệm**

**HOIMTO** là phiên bản cải tiến của thuật toán **HOIM**, được đề xuất nhằm khắc phục một số hạn chế của HOIM. Cụ thể, HOIMTO không chỉ dựa trên **độ chiếm dụng tập mục (IO)** mà còn kết hợp thêm **độ chiếm dụng giao dịch (TO)** để đánh giá chính xác hơn mức độ quan trọng của các tập mục trong cơ sở dữ liệu.

Mục tiêu: Ưu tiên các tập mục xuất hiện trong **giao dịch lớn/quan trọng**, từ đó lọc ra các mẫu có ý nghĩa thực tiễn cao hơn.

* 1. **Thành phần chính**

1. **Transaction Occupancy (TO)**

TO đo độ quan trọng của từng giao dịch dựa trên số lượng mục bên trong nó:

Trong đó:

* : số lượng mục trong giao dịch TT
* Mẫu số là tổng số mục trong toàn bộ tập dữ liệu

Giao dịch dài → TO cao → giao dịch quan trọng hơn → được ưu tiên khi đánh giá tập mục.

1. **Itemset Occupancy (IO) (có điều chỉnh)**

Tức là thay vì đếm cứng như trong HOIM, HOIMTO **cộng dồn TO** của các giao dịch chứa tập mục X.

1. **IOUB (Itemset Occupancy Upper Bound)**

Giới hạn trên giúp loại bỏ sớm các tập mục không tiềm năng:

Nếu ⇒ Không cần mở rộng tập X, vì không thể đạt yêu cầu.

* Cơ chế này giúp **cắt tỉa cây tìm kiếm** rất hiệu quả.
  1. **Điều kiện chọn tập mục**

Tương tự như HOIM, một tập mục XX được chọn nếu:

Tuy nhiên, IO ở đây là tổng các TO giao dịch chứa XX.

* 1. **Ưu điểm của HOIMTO**
* **Chính xác hơn**: Giao dịch quan trọng được xem xét ưu tiên.
* **Loại bỏ tập mục yếu sớm** nhờ IOUB.
* **Hiệu suất cải thiện rõ rệt** trên tập dữ liệu thực tế.
  1. **Nhược điểm**
* **Không xử lý trọng số từng mục (chỉ trọng số giao dịch).**
* **Chi phí tính toán vẫn còn cao** khi làm việc với dữ liệu lớn.
* IOUB vẫn có thể bị lỏng → cần tối ưu them.
  1. **Ví dụ minh họa đơn giản**

Giả sử:

* Dữ liệu có 3 giao dịch:
  + T1 = {A, B, C} → TO(T1) = 3/7
  + T2 = {A, B} → TO(T2) = 2/7
  + T3 = {A, D} → TO(T3) = 2/7

→ Tổng TO = 1

Xét tập mục {A, B}:

* Xuất hiện trong T1 và T2
* IO({A, B}) = TO(T1) + TO(T2) = 3/7 + 2/7 = **5/7 ≈ 0.714**

Nếu đặt MinIO = 0.6 → {A, B} được chọn là tập mục chiếm dụng cao.

1. Tìm hiểu về Cơ sở dữ liệu có trọng số (Weighted Database - WD)
   1. **Khái niệm**

**Cơ sở dữ liệu có trọng số (Weighted Database)** là dạng cơ sở dữ liệu trong đó **mỗi mục (item)** hoặc **mỗi giao dịch (transaction)** được gán một **trọng số (weight)** phản ánh **mức độ quan trọng**, **giá trị**, hoặc **độ ưu tiên** của chúng.

Trọng số có thể đại diện cho:

* **Giá trị tiền tệ** (ví dụ: giá sản phẩm)
* **Mức độ ưu tiên** trong phân tích
* **Tầm quan trọng** của mục đối với người dùng hoặc hệ thống
  1. **Các loại trọng số**

|  |  |
| --- | --- |
| **Loại** | **Mô tả** |
| **Trọng số mục (item weight)** | Gán trọng số cho từng mặt hàng hoặc mục dữ liệu (ví dụ: A = 5$, B = 10$) |
| **Trọng số giao dịch (transaction weight)** | Giao dịch tổng thể có trọng số riêng (ví dụ: giao dịch của khách VIP) |

* 1. **Ứng dụng thực tiễn**
* **Phân tích giỏ hàng bán lẻ**: Ưu tiên các mục có giá trị cao để tối ưu hóa lợi nhuận.
* **Khai phá dữ liệu giao dịch**: Phát hiện các tổ hợp mục có "giá trị" thực sự, thay vì chỉ xuất hiện nhiều.
* **Thương mại điện tử**: Gợi ý sản phẩm không chỉ dựa vào tần suất, mà còn dựa trên trọng số (lợi nhuận, xu hướng...)
  1. **Thuật toán liên quan**

Một số thuật toán đã được đề xuất để khai thác dữ liệu có trọng số:

|  |  |
| --- | --- |
| **Thuật toán** | **Mô tả** |
| **FWI (Frequent Weighted Itemset)** | Dựa trên ngưỡng **weighted support** để lọc tập mục |
| **HUIM (High Utility Itemset Mining)** | Khai phá tập mục có **lợi ích cao**, tính theo trọng số và số lượng |
| **NFWI (Weighted N-list)** | Tối ưu hóa FWI bằng cách dùng cấu trúc N-list có trọng số |

* 1. **Cách tính “weighted support”**

Ví dụ: Nếu mục A có trọng số 5, xuất hiện 3 lần trong các giao dịch:

→ Có thể đặt một ngưỡng để quyết định tập mục có đủ quan trọng hay không.

* 1. **Thách thức**
* **Trọng số không đồng đều** → cần phương pháp lọc hiệu quả để tránh nhiễu.
* **Không thể áp dụng trực tiếp các thuật toán phổ biến như Apriori** nếu không cải tiến chúng.
* **Cần thiết kế lại cách tính độ quan trọng của tập mục** (utility/occupancy) sao cho phù hợp với dữ liệu có trọng số.
  1. **Liên hệ với đề tài**

HOIMTO (High-Occupancy Itemset Mining with Transaction Occupancy) hiện tại **chưa xét trọng số**, tức là mặc định mọi mục trong giao dịch đều có "trọng lượng bằng nhau".  
=> Đây là **khoảng trống lý thuyết** để đề xuất thuật toán mới **HOWI-MTO (High-Occupancy Weighted Itemset Mining with Transaction Occupancy)** nhằm:

* Tích hợp trọng số vào TO và IO
* Đánh giá chính xác hơn vai trò của từng mục trong khai thác

Chương 2

* 1. Hiểu lý thuyết của HOIM và HOIMTO