PROBABILISTIC MAXIMAL FREQUENT ITEMSET OVER UNCERTAIN DATABASE

# Introduction:

# Related world:

# Preliminaries and problem definition:

## Preliminaries:

### Uncertain database and possible world:

Cho 1 tập hợp các phần tử riêng biệt : 1 tập con: được gọi là **itemset**

, , ,…,

Với mỗi item ( ) trong được liên kết với một xác xuất xuất hiện ): X được gọi là **uncertain itemset**, biểu thị:

Uncertain transaction UT là một uncertain itemset đi cùng với một ID.

Uncertain database UD là một tập hợp các uncertain transaction:

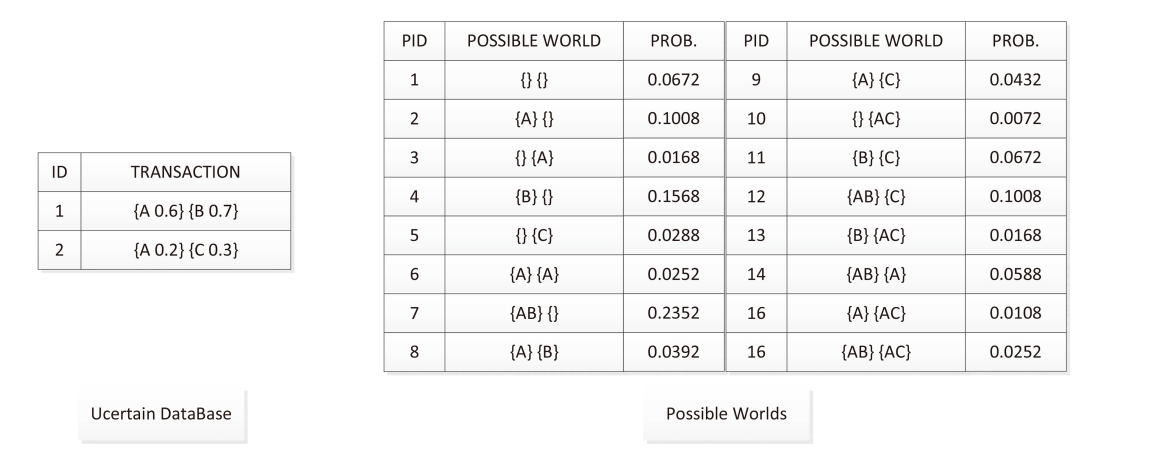
Ví dụ uncertain database:

|  |  |
| --- | --- |
| **ID** | **Transaction** |
| 1 | {A 0.6} {B 0.7} |
| 2 | {A 0.2} {C 0.3} |

Bằng cách sử dụng possible world model, một uncertain database có thể chuyển thành nhiều exact database.

**Định nghĩa 1:** (Possible World) Một possible world được chuyển đổi từ uncertain database là một exact database với transaction, nơi mà mỗi transaction là một tập con của . Chúng ta ký hiệu một possible world là PW = {, ,… , }, trong đó ⊆

Nếu chúng ta giả sử rằng các UT là độc lập thì một PW có xác suất xảy ra p(PW), có thể được tính bằng cách nhân xác suất xuất hiện của từng mục x ∈ PW nếu , cũng như một lần xác suất xuất hiện của mỗi mục nếu x thuộc , nhưng không thuộc , ký hiệu là:



Chúng ta sẽ kí hiệu tất cả possible world được phát sinh từ UD là . Tuy nhiên, số lượng phần tử của sẽ tang lên theo số mũ theo kích cỡ database gốc. Để mô tả chi tiết thì giả sử một uncertain database bao gồm m transaction, và mỗi transaction bao gồm nm item. Thì bao gồm possible world.

Số phần tử của PW theo công thức sau:

Với :

+ m là số lượng transaction

+ n là số lượng itemset trong mỗi transaction

Ví dụ: Số phần tử trong possible trong ví dụ trên là phần tử.

Mỗi phần tử trong possible world model sẽ được tính theo công thức (1).

Ví dụ:

p(PW6) =

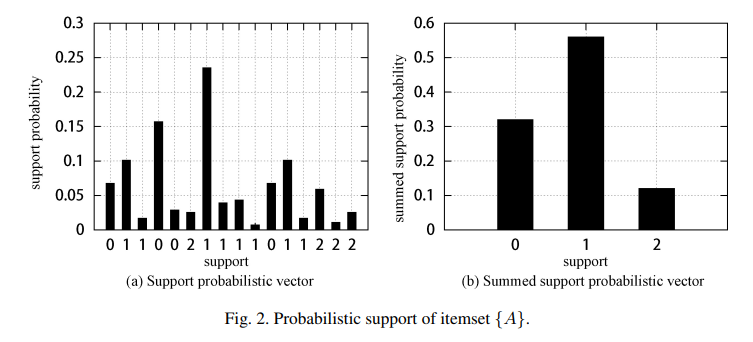
p{A}∈UT1∧{A}∈T1({A})∗p{B}∈UT1∧{B}6∈T1({B})∗p{A}∈UT2∧{A}∈T2({A})∗p{C}∈UT2∧{C}6∈T2({C}) = 0.6 \* 0.3 \* 0.2 \* 0.7 = 0.0252.

Khi tính tổng các xác suất trong PW model, chúng ta nhận thấy tổng xác suất là 1.

### Frequent itemset over uncertain databases:

Trong exact database, cho 1 ngưỡng cụ thể (minimum support) ), một frequent itemset là một itemset mà có số lần xuất hiện trong ít nhất transaction. Chúng ta có thể nói rằng một itemset là frequent nếu support không bé hơn .

Trong một uncertain database, frequent itemset được định nghĩa với Possible world model. Cho 1 itemset X trong mỗi possible world PW được phát sinh từ UD, đầu tiên chúng ta cần đạt được số lần xuất hiện cùng với xác suất xuất hiện . Chúng ta biểu thị chúng với là 2-tuple . Điều này có nghĩa rằng item X có độ support  với xác suất . Do đó, X trong UD có tuple, chúng được tổ chức lại và biểu diễn thành summed probabilistic vector, được biểu thị .



Ví dụ, trong Hình 1, itemset {A} có 16 giá trị trong probabilistic vector, được biểu thị trong Hình 2a. Chúng ta có thể thấy rằng {A} không xuất hiện trong các possible world , , , và , với xác suất tương ứng lần lượt là 0.0672, 0.1568, 0.0288 và 0.0672. Đẻ tích lũy xác suất của những phần từ có cùng support, chúng ta có thể dung summed probabilistic vector, , được trình bày trong Hình 2b. {A} không xuất hiện với xác suất 0.32 trong uncertain database.

Dựa trên summed probabilistic vector, 2 định nghĩa: expected frequent itemsets và probabilistic frequent itemset được đề xuất

**Định nghĩa 2**: (Expected Frequent Itemset) Cho một uncertain database UD, 1 itemset X là một -expected frequent itemset nếu expected support của nó không bé hơn minimum support . Với:

Trong Hình 2a, expected support của itemset {A} là . Chúng ta cũng nhận thấy rằng nó cũng có thể được tính toán bằng cách tích lũy tất cả xác suất của {A} trong uncertain database UD. Trong Hình 1, . Do đó, chúng ta đơn giản tính được expected support với công thức sau:

**Định nghĩa 3**: (Probabilistic Frequent Itemset) Cho một uncertain database UD, minimum support và minimum probabilistic confidence , một itemset X được gọi là một -probabilistic frequent itemset nếu xác suất support của nó không nhỏ hơn , biểu thị thì lớn hơn .

Probabilistic frequent itemset tập trung vào việc đạt được khả năng của 1 itemset mà nó xuất hiện ít nhất transaction.

Ví dụ: Đối với itemset {A}, cho minimum support = 2, thì . Chúng ta gọi itemset {A} là môt frequent itemset với xác suất 0.12.

Tuy nhiên, vì Định nghĩa 3 không nhất quán với định nghĩa truyền thống của frequent itemset. Nó cũng không đủ thuận tiện để định nghĩa cho những định nghĩa liên quan trong exact database, như là close and maximal itemset. Do đó, một định nghĩa mới tương đương của probabilistic frequent itemset được đề xuất.

**Định nghĩa 4**: (Probabilistic Support) Cho minimum probabilistic confidence , một uncertain database UD và một itemset X, probabilistic support của X được biểu thi , là giá trị tối đa support của itemset X với probabilistic confidence .

**Định nghĩa 5**: (Probabilistic Frequent Itemset) Cho một uncertain database UD, minimum support , minimum probabilistic confidence , một itemset X là probabilistic frequent itemset nếu .

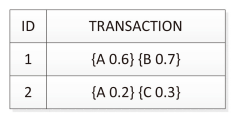
## Problem definition

**Định nghĩa 6:** (Probabilistic Maximal Frequent Itemset) Cho một uncertain database, minimum support , minimum probabilistic confidence , một itemset X là probabilistic maximal frequent itemset nếu nó là một probabilistic frequent itemset và nó không bị chưa bởi một probabilistic frequent itemset khác, biểu thị: .

Cho một probabilistic maximal frequent itemset X, bất kì itemset Y nào sao cho thỏa mãn mệnh đề sau: Xác suất của support Y không bé hơn thì lớn hơn .

**Problem statement**: Cho một uncertain database UD, minimum support , minimum probabilistic confidence , chúng ta được yêu cầu tìm ra tất cả probabilistic maximal frequent itemset trong UD.

|  |  |
| --- | --- |
| **Notation** | **Meaning** |
| UD | Uncertain database |
|  | Minimum support |
|  | Minimum probabilistic support |
|  | Độ support của itemset X (tần số xuất hiện của itemset X xuất hiện trong UD) |
|  | Expected support của itemset X |
|  | Xác suất độ support của itemset X |
|  | Xác suất độ support của itemset X không nhỏ hơn |
|  | Probabilistic support của itemset X |



**Ví dụ 1**: Cho 1 uncertain database UD như hình trên, minimum support = 1 và minimum probabilistic confidence = 0.1. Yêu cầu tìm ra tất cả Probablilistic maximal frequent itemset.

Chúng ta có thể nhận được kết quả sau: . Những probabilistic frequent itemset là {A}, {B}, {C}, {AB} và probabilistic maximal frequent itemset là {C} và {AB}.

Chúng ta cũng có thể suy ra từ những probabilistic maximal frequent itemset rằng xác suất support của itemset {A} không nhỏ hơn 1 thì lớn hơn 0.1.

# Probabilistic maximal frequent itemset method

## Data structure:

### Probabilistic maximal frequent itemset tree:

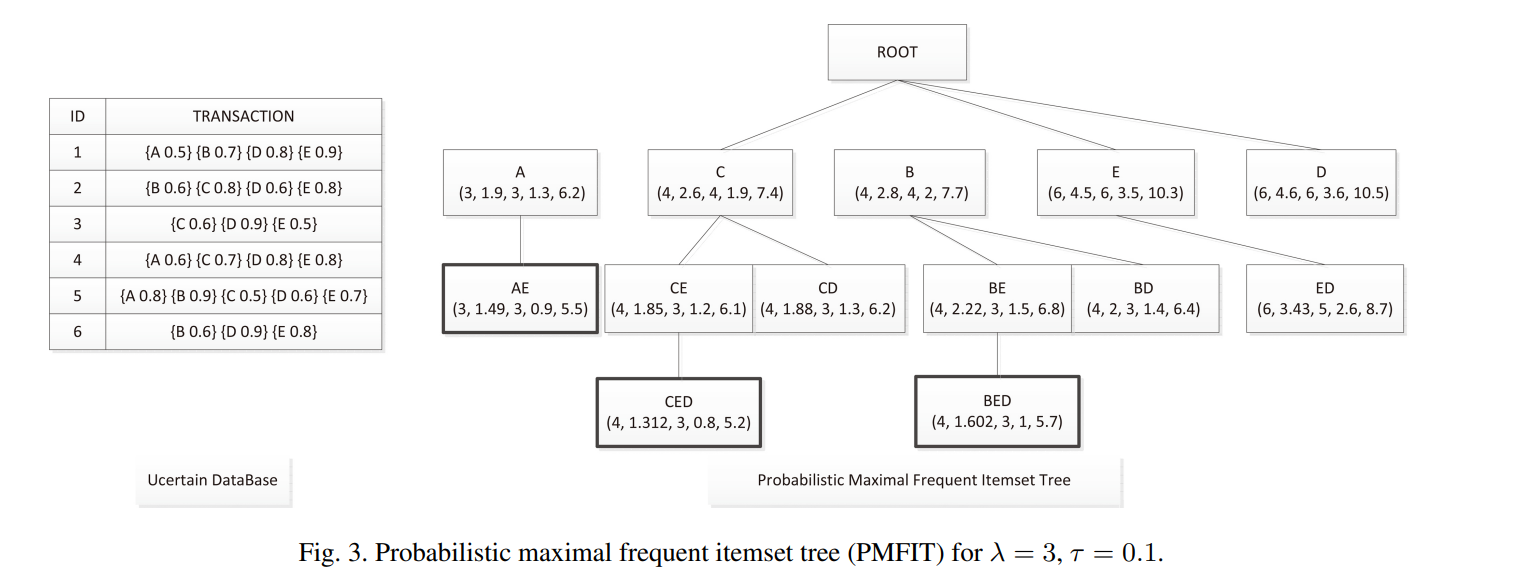
Để gia tăng tốc độ tìm kiếm và cắt tỉa, chúng tôi sử dụng Index tree hiệu quả gọi là PMFIT (Probabilistic Maximal Frequent Itemset Tree).

Mỗi node biểu thị một itemset X là một 6-tuple <item, sup, esup, psup, lb, ub>.

Trong đó item là itemset X hiện tại, sup là độ support, esup là expected support, và psup là probabilistic support. Lb và ub đại diện cho giới hạn trên và dưới của probabilistic support.

Ngoại trừ root node thì mỗi node đều có pointer tới parent node.

Ví dụ:



Trong ví dụ trên trình bày FMFIT từ 6 transaction trong uncertain database.

Itemset {A} có support là 3, expected support là 1.9, probabilistic support 3.

Ngoài ra ta có thể thấy có 13 node là probabilistic frequent itemset và chỉ có 3 node là probabilistic maximal frequent itemset.

### Probabilistic list:

Sử dụng một array trong suốt quá trình implement để duy trì danh sách xác suất của itemset xuất hiện trong tất cả transaction. Danh sách xác suất này được sử dụng để tính probabilistic support. Tuy nhiên, đây chỉ là tạm thời vì trong thuật toán thì probabilistic support không cần tính toán cho mỗi itemset.

### Probabilistic maximal frequent itemset collection:

Bởi vì kết quả cuối cùng không cần phải duy trì xác suất của mỗi item, chúng ta sử dụng một **bitmap-based collection** để lưu trữ probabilistic maximal frequent itemset. Do đó, cắt giảm chi phí bộ nhớ và gia tang tốc độ tìm kiếm.

## Probabilistic support computing:

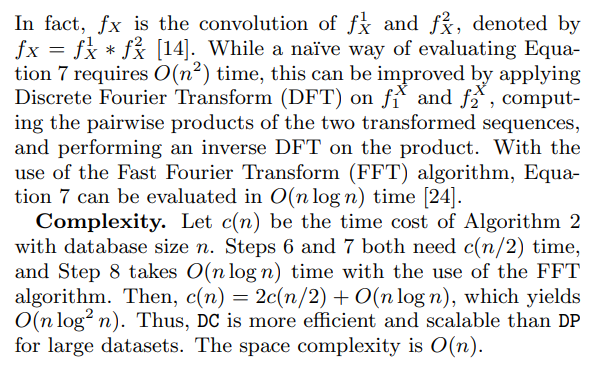
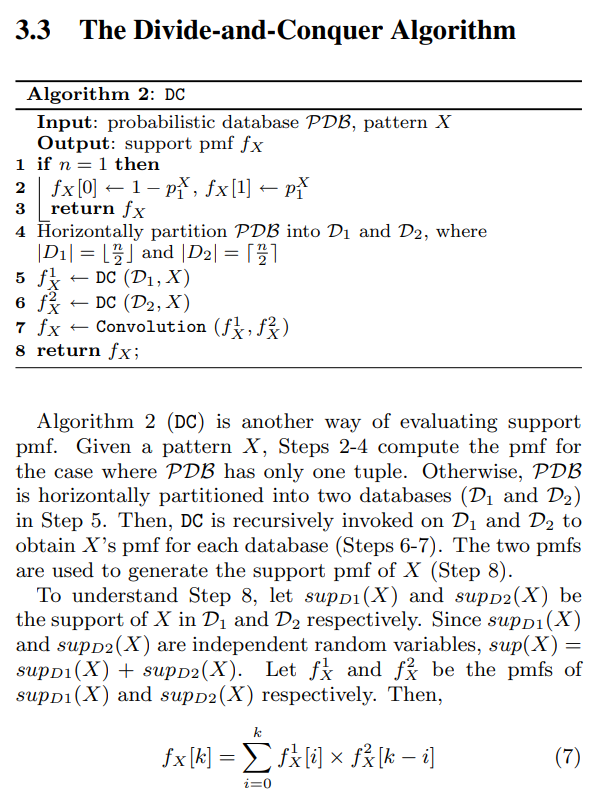
Bởi vì summed support probabilistic vector của itemset X trong 2 transaction là sự tích chập của nó trong , phương pháp chia để trị [20] được đề xuất được sử dụng trong bài báo này*. Uncertain database sẽ được chia làm 2 phần để tính toán summed support probabilistic vector, và phép toán này sẽ được tiến hành đệ quy cho đến khi sub-database chỉ còn 1 transaction. Sự tích chập này có thể được tính bằng* ***Fast Fourier Transform***, cho kích thước n của uncertain database, sẽ cắt giảm hiệu quả độ phức tạp về thời gian từ thành . Summed support probabilistic vector được tính toán , chúng ta có thể nhận được probabilistic support, nó là giá trị tối đa của t sao cho lớn hơn minimum probabilistic confidence .

Cho uncertain database UD, ta có thể tính được summed support probabilistic vector của itemset X bằng phương pháp chia để trị và tích chập bằng FFT.

Với summed support probabilistic vector của itemset X: , ta có thể tính được probabilistic support của itemset X theo công thức:

Ví dụ: Cho UD nhưu Hình 1, minimum probabilistic confidence = 0.1, summed support probabilistic vector của itemset {A} trong ví dụ trên là:

SSPV({A}) = {0.32, 0.56, 0.12}. Ta có thể tính được probabilistic support của itemset {A} là 2 (do 0.12 > 0.1 và giá trị t của 0.12 là 2}



//code convolution 2 vector

int len = f1X.length + f2X.length - 1;

double[] fX = new double[len];

for (int k = 0; k < len; k++) {

for (int i = 0; i <= k; i++) {

if (i < f1X.length && (k - i) < f2X.length) {

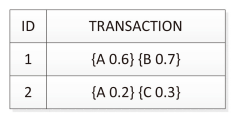
fX[k] += f1X[i] \* f2X[k - i];

}

}

}

Cho UD sau tính Summed Support Probabilistic Vector của itemset {A}:



Step 2:

D1: (A 0.6) (B 0.7)

D2: (A 0.2) (C 0.3)

double[] f1X = DC(D1, pattern);

double[] f2X = DC(D2, pattern);

Step 3:

DC(D1, 'A');

f1X = [1 - 0.6, 0.6] = [0.4, 0.6]

DC(D2, 'A')

f2X = [1 - 0.2, 0.2] = [0.8, 0.2]

length(fX) = 2 + 2 - 1 = 3

fx = new fx[3]

//convolution(f1X, f2X)

- k = 0:

i = 0; i <= k => True

ĐK: i < f1X.length && (k - i) < f2X.length

0 < 2 && 0 - 0 < 2 => True

fX[k] = f1X[i] \* f2X[k - i]

=> fX[0] = f1X[0] \* f2X[0 - 0]

= f1X[0] \* f2X[0] = 0.4 \* 0.8 = 0.32

i = 1 => break

=> fx[0] = 0.32

- k = 1

i = 0; i <= k

ĐK: i < f1X.length && (k - i) < f2X.length

0 < 2 && 1 - 0 < 2 => True

fX[k] = f1X[i] \* f2X[k - i]

=> fX[1] = f1X[0] \* f2X[1 - 0] = f1X[0] \* f2X[1] = 0.4 \* 0.2 = 0.08

i = 1; i <= k

ĐK: i < f1X.length && (k - i) < f2X.length

1 < 2 && 1 - 1 < 2 => True

fX[k] = f1X[i] \* f2X[k - i]

=> fX[1] = f1X[1] \* f2X[1 - 1] = f1X[1] \* f2X[0] = 0.6 \* 0.8 = 0.48

i = 2 => break

=> fX[1] = 0.08 + 0.48 = 0.56

- k = 2:

i = 0; i <= k => True

ĐK: i < f1X.length && (k - i) < f2X.length

0 < 2 && 2 - 0 < 2 => False

i = 1 => i <= k => True

ĐK: i < f1X.length && (k - i) < f2X.length

1 < 2 && 2 - 1 < 2 => True

fX[k] = f1X[i] \* f2X[k - i]

=> fX[2] = f1X[1] \* f2X[2 - 1] = f1X[1] \* f2X[1] = 0.6 \* 0.2 = 0.12

i = 2 => i <= k => True

ĐK: i < f1X.length && (k - i) < f2X.length

2 < 2 && 2 - 2 < 2 => False

=> end loop

=> fX[2] = 0.12

=> fx = [0.32, 0.56, 0.12]

## Items reordering

Bayardo tuyên bố rằng việc sắp xếp item theo sự tăng dần của **support** có thể cắt giảm không gian tìm kiếm.

Trong bài báo này sử dụng phương pháp tương tự, các item sẽ đươc sắp xếp theo sự tăng dần của **expected support** => ngta tin rằng điều này sẽ làm cho thuật toán hiệu quả hơn.

Mặc dù sắp xếp theo sự tăng dần của probabilistic support là phương pháp tốt nhất, nhưng không sử dụng vì nó sẽ ảnh hưởng đến hiệu suất khi mà minimum support không đủ thấp.

* Trong bài báo này sử dụng 1 **array**, gọi là **sorted items list**, để duy trì sorted items.

## Pruning strategies

Được lấy cảm hứng bởi phương pháp maximal frequent itemset mining trên regular database và đề xuất một vài chiến lược cắt tỉa đó là cung cấp bound chặc chẽ để suy ra probabilistic support hoặc ẩn đi việc tính toán probabilistic support và cải thiện hiệu năng.

### Bounds of probabilistic support

Cho một n-transactions uncertain database, một phương pháp hiệu quả để tính probabilistic support là phương pháp chia để trị; tuy nhiên, chi phí về thời gian chạy và sử dụng bộ nhớ vẫn lớn. Khi việc khai phá probabilistic maximal frequent itemset, thì probabilistic support không quan trọng cho users, vì thế họ cố gắng tìm ra phương pháp để suy ra frequency của itemset thay vì tính trực tiếp probabilistic.

**Theorem 1**: Cho một itemset X, minimum probabilistic confidence , uncertain database UD, nếu X xuất hiện trong transaction với probabilistic , thì probabilistic support đơn điệu nhất quán với bất kì nào.

**Theorem 2**: Đối với một itemset X trong uncertain database UD, cho minimum support , minimum probabilistic confidence , thì .

**Theorem 3**: Đối với một itemset X trong uncertain database UD, cho minimum support , minimum probabilistic confidence , chúng ta có thể tính được lower và upper bound của probabilistic support , biểu thị và theo công thức sau:

Theorem 3 cung cấp 2 chiến lược cắt tỉa:

Đối với một itemset X, nếu uppber bound không lớn hơn minimum support , thì X là probabilistic infrequent itemset. Cùng đó, nếu lower bound không bé hơn , thì X chắc chắn probabilistic frequent itemset.

**Ví dụ 2**: Chúng ta sử dụng uncertain database trong Hình 3 như ví dụ.

Nếu chúng ta đặt = 1 và minimum probabilistic confidence = 0.1, thì đối với itemset {A}, lower bound là 1.3, nó lớn hơn 1; do đó {A} là frequent itemset.

Nếu chúng ta đặt = 5 và minimum probabilistic confidence = 0.1, thì đối với itemset {AB}, upper bound là 4.7, nó bé hơn 5; do đó {AB} là infrequent itemset.

### Superset pruning

Để một item là probabilistic maximal frequent itemset phải thỏa hai điều kiện:

1. Probabilistic support không bé hơn một ngưỡng nhất định.
2. Nó không là tập con của bất kì probabilistic frequent itemset khác.

Cả 2 sự tính toán đều cần thiết, vì vậy việc tính toán với chi phí thấp hơn nên được tiến hành trước. Như đã đề cập ở trên, việc tính probabilistic support yêu cầu độ phức tạp về thời gian là với n là kích thước database. Mặc dù với phương pháp được đưa ra của chúng tôi thì độ phức tạp về thời gian là O(n). Tuy nhiên, để tìm được tồn tại probabilistic maximal frequent itemset, kích thước của nó giả sử là m, yêu cầu độ phức tạp về thời gian là O(m). Nhìn chung, m bé hơn nhiều so với n. Do đó, đối với itemset mới được phát sinh, chúng ta sẽ quyết định liệu là nó bị bao bởi một super itemset, nếu ko chúng ta tính bound hoặc probabilistic support. Chiến lược này có thể mở rộng them cho việc cắt tỉa. Cho một probabilistic maximal frequent itemset , nếu chúng là n item cuối cùng trong sorted item list, thì item có thể được cắt tỉa trực tiếp.

## Algorithm description:

Trong bài báo này, đề xuất thuật toán khai phá depth-first probabilistic maximal frequent itemset xây dựng cây được tổ chức bottom-up; đó là, subset sẽ được tính trước và sau đó superset sẽ được generate nếu subset của nó tất cả là frequent. Tìm ra những itemset được thực hiện theo chiều dọc bởi vì probabilistic frequent itemset cũng tuân theo apriori property.

Step 1: Chúng ta lấy tất cả item phân biệt và sắp xếp chúng theo sự tăng dần của expected support trước khi xây dựng PMFIT. Ngoài ra, item có support hoặc upper bound bé hơn minimum support sẽ bị loại bỏ.

Step 2: PMFIT được khởi tạo với chỉ root node (có giá trị là null).

Step 3: Đối với một parent node, chúng ta bắt đầu generate ra child node, và

**Algorithm 1**: PMFIM Algorithm

Yêu cầu:

* : node của PMFIT biểu thị cho itemset I
* UD: Uncertain Database
* PMFIC: Probabilistic Maximal Frequent Itemset Collection
* : minimum support
* : minimum probabilistic confidence

Lấy những itemsets J thứ tự lớn hơn I;