ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN

KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG
Bài tập Thực hành 2

KHAI THÁC LUẬT KẾT HỢP

Nhóm thực hiện

1. Hồng Thanh Hoài 1612855
2. Huỳnh Minh Huấn 1612858

Giáo viên lý thuyết
PGS.TS Lê Hoài Bắc

Giáo viên hướng dẫn
Nguyễn Ngọc Thảo, Lê Ngọc Thành

Tháng 04 năm 2019

Mục lục

1 Giới thiệu nhóm và phân công công việc

1.1 Giới thiệu nhóm

Nhóm gồm 2 thành viên.

STT	Họ và tên	MSSV	Email	SĐT
1	Hồng Thanh Hoài	1612855	hthoai1006@gmail.com	0965596807
2	Huỳnh Minh Huấn	1612858	minhhuanhuynh289@gmail.com	0824540646

1.2 Phân công công việc

STT	Họ và tên	Công việc
1	Hồng Thanh Hoài	Câu B1, B2, B3, báo cáo.
2	Huỳnh Minh Huấn	Câu A1, A2, B4, B5.

2 Lý thuyết

2.1 Phương pháp cải tiến quá trình tìm luật kết hợp từ tập phổ biến

- Tập phổ biến có thể được lưu trữ bằng bảng băm cùng với *supp* của chúng để truy cập nhanh chóng.
  - Xây dưng bảng băng:
    + Chọn hàm băm h phù hợp.
    + Chọn phương thức xử lý *collison* phù hợp.
    + Băm các tập phổ biến tìm được theo hàm băng vừa xây dựng vào bảng băm (bảng băm chứa *k-item* và *supp*).
      Ví dụ: A - 2, AB - 2 (cấu trúc lưu trữ tùy thuộc vào người lập trình).
  - Với mỗi luật $X \to Y$ $(X, Y \subset FI)$, ta có $conf(X \to Y) = \frac{supp(XY)}{supp(X)}$.
    Ta tìm *supp* của X, XY bằng cách *lookup* trong bảng băm.
- Thay vì khai thác luật truyền thống, ta khai thác luật thu gọn (chứa tất cả các luật như khi khai thác truyền thống, trừ những luật suy ra từ tính bắc cầu) để tăng tốc độ.

```
SHORTEN_AR()
    SORT (FI) // Sắp xếp tập FI tăng theo k-itemset
    AR = ∅
    // Với mọi phần tử Y trong FI, tìm ra các phần tử X sao cho
    // |X| = |Y| - 1
    for each Y ∈ FI with |Y| > 1 do
        for each X ∈ FI with |X| = |Y| - 1 do
            // Nếu X là tập con của Y, ta tính conf(X → Y − X)
            if X ⊂ Y then
                conf = Sup(Y) / Sup(X)
            // Nếu conf của luật vừa tính lớn hơn minConf thì thỏa
            if conf ≥ minConf then
                AR = AR ∪ X → Y \X (Sup(Y), conf)
    return AR
```

→ **Đánh giá:** Khi tìm kiếm *k-item* (để tìm *supp(k-item)*) trên bảng băm sẽ nhanh hơn so với cách thông thường là tìm trực tiếp trong FI (nếu tìm được hàm băm và phương pháp xử lý *collision* phù hợp). Cộng thêm việc khai thác luật thu gọn sẽ giảm đáng kể thời gian so với cách thông thường.

2.2 Áp dụng thuật toán với CSDL

2.2.1 Tìm tập phổ biến

Bảng 1: Bảng dữ liệu.

TID	Item bought
100	I, B, F, D, E, C, H, J
200	F, C, F, G, A, D, C
300	B, J, D, A, H
400	E, A, B, E, G

- **Apriori**

Bảng 2: Tập Large 1-item L1.

(a) C1

Itemset	Count
{A}	3
{B}	3
{C}	2
{D}	3
{E}	2
{F}	2
{G}	2
{H}	2
{I}	1
{J}	2

(b) L1

Itemset	Count
{A}	3
{B}	3
{C}	2
{D}	3
{E}	2
{F}	2
{G}	2
{H}	2
{J}	2

Bảng 3: Tập Large 2-item L2.

(a) C2

Itemset	Count	Itemset	Count
AB	2	CF	2
AC	1	CG	1
AD	2	CH	1
AE	1	CJ	1
AF	1	DE	1
AG	2	DF	2
AH	1	DG	1
AJ	1	DH	2
BC	1	DJ	2
BD	2	EF	1
BE	2	EG	1
BF	1	EH	1
BG	1	EJ	1
BH	2	FG	1
BJ	2	FH	1
CD	2	FJ	1
CE	1	HJ	2

(b) L2

Itemset	Count
AB	2
AD	2
AG	2
BD	2
BE	2
BH	2
BJ	2
CD	2
CF	2
DF	2
DH	2
DJ	2
HJ	2

Bảng 4: Tập Large 3-item L3.

(a) C3

Itemset	Count
ABD	1
BDH	2
BDJ	2
BHJ	2
CDF	2
DHJ	2

(b) L3

Itemset	Count
BDH	2
BDJ	2
BHJ	2
CDF	2
DHJ	2

Bảng 5: Tập Large 4-item L4.

(a) C4

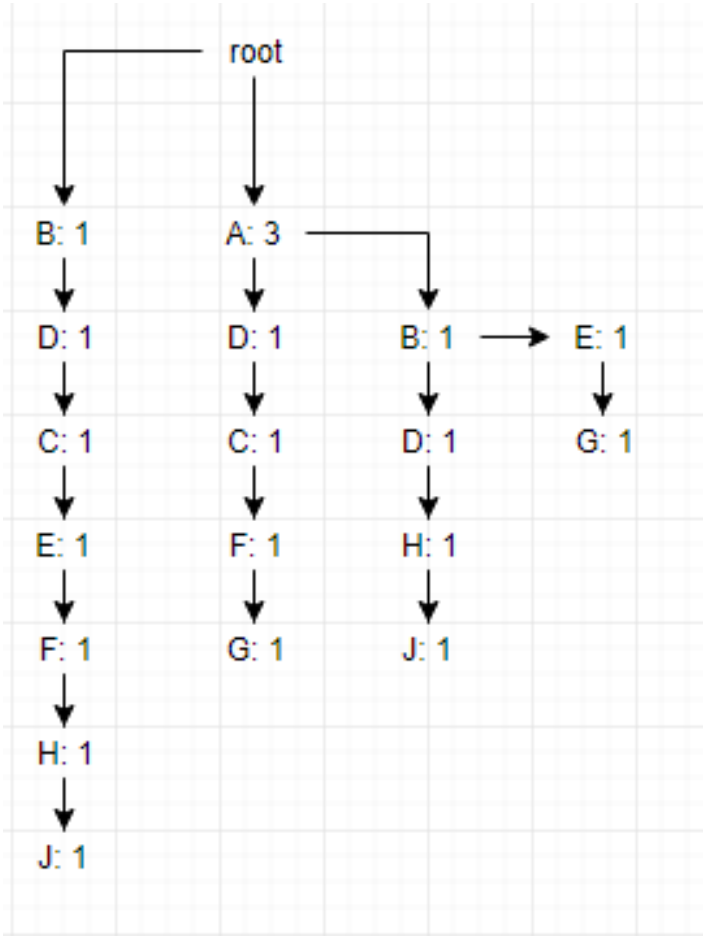
Itemset	Count
BDHJ	2

(b) L4

Itemset	Count
BDHJ	2

→ Tất cả các tập phổ biến tìm được với thuật toán Apriori:
`{A, B, D, C, E, F, G, H, J, AB, AD, AG, BD, BH, BE, BJ, CD, CF, DF, DH, DJ, HJ, BDH, BDJ, BHJ, CDF, DHJ, BDHJ}`.

- **FP-Growth**



Hình 1: Cây FP.

Bảng 6: Bảng xây dựng cây cơ sở mẫu điều kiện và cây FP điều kiện, mẫu phổ biến.

Item	**Cơ sở mẫu điều kiện**	**FP-Tree điều kiện**	**Các mẫu phổ biến**
J	{BDCEFH: 1, ABDH: 1}	{B: 2, D: 2, H: 2} - J	J, BJ, DJ, HJ, BDJ, BHJ, DHJ, BDHJ
H	{BDCEF: 1, ABD: 1}	{B: 2, D: 2} - H	H, BH, DH, BDH
G	{ADCF: 1, ABE: 1}	{A: 2} - G	G, AG
F	{BDCE: 1, ADC: 1}	{C: 2, D: 2} - F	F, CF, DF, CDF
E	{BDC: 1, AB: 1}	{B: 2} - E	E, BE
C	{BD: 1, AD: 1}	{D: 2} - C	C, CD
D	{B: 1, A: 1, AB: 1}	{B: 2, A: 2} - D	D, BD, AD
B	{A: 2}	∅	B, AB
A	∅	∅	A

→ Tất cả các tập phổ biến tìm được với thuật toán FP-Growth:
`{A, B, D, C, E, F, G, H, J, AB, AD, AG, BD, BH, BE, BJ, CD, CF, DF, DH, DJ, HJ, BDH, BDJ, BHJ, CDF, DHJ, BDHJ}`.

- **So sánh kết quả**
  Kết quả tìm được của 2 thuật toán là **như nhau**.
- **Tập phổ biến tối đại**
  Các tập phổ biến tối đại (Maximal frequent itemsets): `AB, AD, AG, BE, CDF, BDHJ`.
- **Tập phổ biến đóng**
  Các tập phổ biến đóng (Closed frequent Itemsets):
  - A: $c(A) = i(t(A)) = i(200, 300, 400) = A$.
  - B: $c(B) = i(t(B)) = i(100, 300, 400) = B$.
  - D: $c(D) = i(t(D)) = i(100, 200, 300) = D$.
  - AB: $c(AB) = i(t(AB)) = i(300, 400) = AB$.
  - AD: $c(AD) = i(t(AD)) = i(200, 300) = AD$.
  - AG: $c(AG) = i(t(AG)) = i(200, 400) = AG$.
  - BE: $c(BE) = i(t(BE)) = i(100, 400) = BE$.
  - CDF: $c(CDF) = i(t(CDF)) = i(100, 200) = CDF$.
  - BDHJ: $c(BDHJ) = i(t(BDHJ)) = i(100, 300) = BDHJ$.

  → Các tập phổ biến đóng: `A, B, D, AB, AD, AG, BE, CDF, BDHJ`.

2.2.2 Luật kết hợp có dạng $(item1 \wedge item2 \rightarrow item3)$

Tất cả các luật kết hợp có dạng $(item1 \wedge item2 \rightarrow item3)$ thỏa mãn *minsupp* và *minconf*: `BDH, BDJ, BHJ, CDF, DHJ, BDHJ`.

- BD → H: $conf(BD \rightarrow H) = 2/2 = 1$.
- BH → D: $conf(BH \rightarrow D) = 2/2 = 1$.
- DH → B: $conf(DH \rightarrow B) = 2/2 = 1$.
- BD → J: $conf(BD \rightarrow J) = 2/2 = 1$.
- BJ → D: $conf(BJ \rightarrow D) = 2/2 = 1$.
- DJ → B: $conf(DJ \rightarrow B) = 2/2 = 1$.
- BH → J: $conf(BH \rightarrow J) = 2/2 = 1$.

- BJ $\rightarrow$ H: $conf(BJ \rightarrow H) = 2/2 = 1$.
- HJ $\rightarrow$ B: $conf(HJ \rightarrow B) = 2/2 = 1$.
- CD $\rightarrow$ F: $conf(CD \rightarrow F) = 2/2 = 1$.
- CF $\rightarrow$ D: $conf(CF \rightarrow D) = 2/2 = 1$.
- DF $\rightarrow$ C: $conf(DF \rightarrow C) = 2/2 = 1$.
- DH $\rightarrow$ J: $conf(DH \rightarrow J) = 2/2 = 1$.
- DJ $\rightarrow$ H: $conf(DJ \rightarrow H) = 2/2 = 1$.
- HJ $\rightarrow$ D: $conf(HJ \rightarrow D) = 2/2 = 1$.

2.2.3 Ứng dụng cải tiến của câu 1

Ta xây dựng bảng băm cho tập phổ biến đã tìm được. Xét k-itemset $X_k = x_1 x_2 \dots x_k$,

$$h(X_k) = \sum_{i=1}^{k} order(x_i) \, mod \, |L_1|$$

trong đó $order(x_i)$ là thứ bậc của item x_i trong tập L_1 và $|L_1|$ là số lượng phần tử một hạng mục trong tập phổ biến.

Bảng 7: Ta có bảng băm.

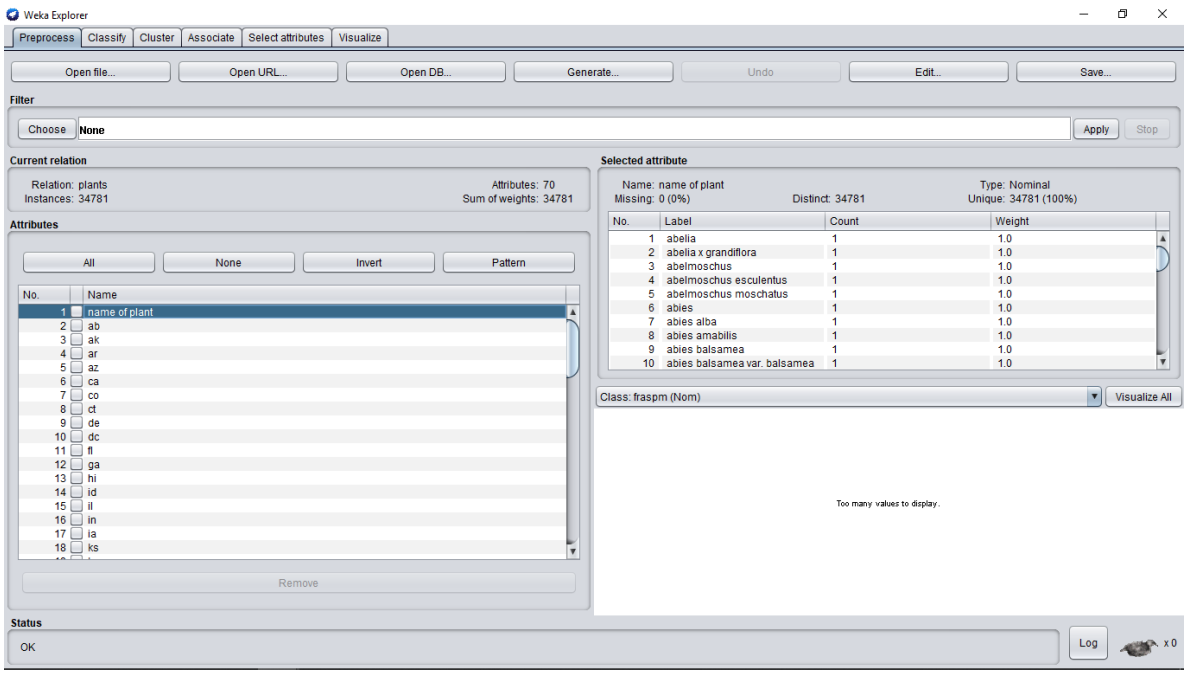
0	A - 3	BJ - 2	DHJ - 2		
1	B - 3	AB - 2	DH - 2	CDF - 2	BDHJ - 2
2	C - 2	DJ - 2	BDH - 2		
3	D - 3	AD - 2	BDJ - 2		
4	E -2	BD - 2			
5	F -2	BE - 2	CD - 2		
6	G -2	AG - 2	HJ - 2		
7	H -2	CF - 2	BHJ - 2		
8	J -2	BH - 2	DF - 2		

Vì luật có dạng ($item1 \wedge item2 \rightarrow item3$) (3 items) nên khi dùng phương pháp truyền thống sẽ có những luật dư thừa và tốn thời gian tính *conf* như là H $\rightarrow$ DJ. Việc dùng phương pháp khai thác luật rút gọn giúp ta chỉ cần tính conf của những luật cần quan tâm (có dạng ($item1 \wedge item2 \rightarrow item3$)) của tập 3-items. Cộng thêm việc *lookup* conf từ hash-table sẽ giảm đáng kể thời gian so với kết quả ở câu b.

3 Thực hành

3.1 Chuyển đổi dữ liệu

Doạn code dùng để chuyển dữ liệu từ dạng giao dịch sang dạng nhị phân được lưu ở file `convertCSV.py`. Sau khi thực hiện chuyển đổi, ta có file `plants.csv`.



Hình 2: File `plants.csv` trên tab *Explorer* của WEKA.

3.2 Trả lời câu hỏi

Về loài cây và vùng phân bố, nhóm sử dụng thông tin trong tab *Explorer* của WEKA. Các câu còn lại được phát sinh từ file `statistic.py`.

- Có tất cả 34781 loài cây.
- Có tất cả 69 vùng phân bố.
- Số loài cây trên mỗi vùng phân bố:

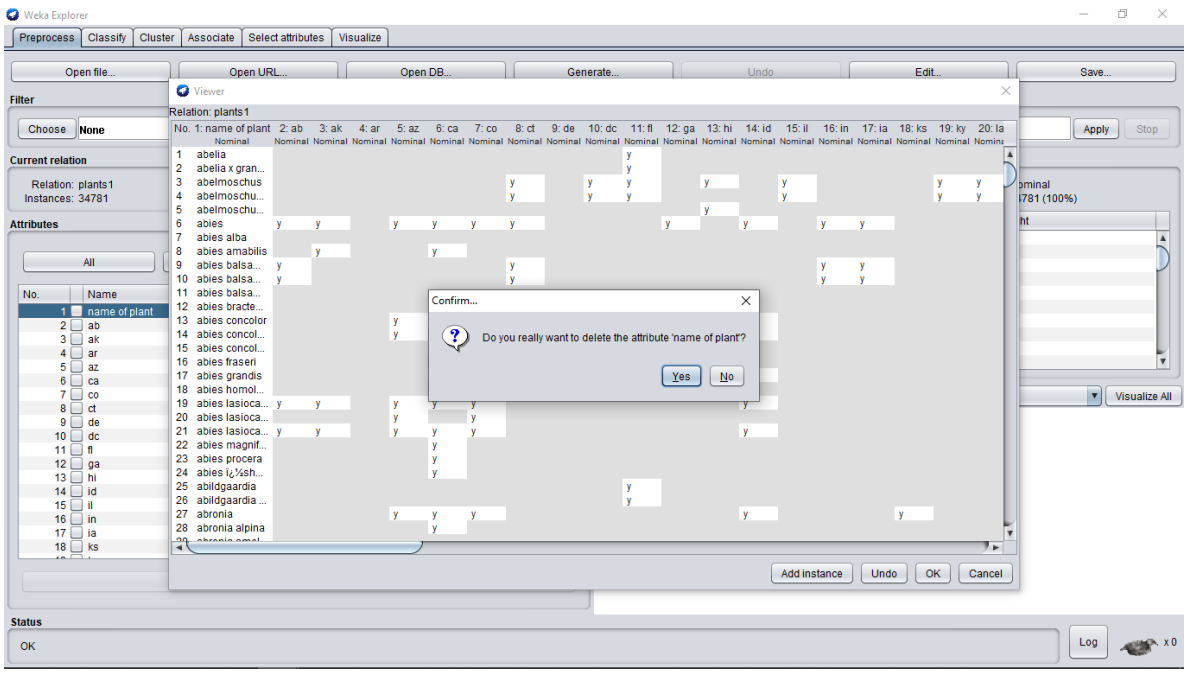
ak : 2969	ks : 3869	nm : 6403	vt : 3713	nu : 979
ar : 4610	ky : 4555	ny : 5773	va : 5638	on : 5068
az : 6778	la : 5154	nc : 5926	vi : 2185	pe : 1841
ca : 11676	me : 3969	nd : 2682	wa : 5654	qc : 4272
co : 5465	md : 5108	oh : 4772	wv : 4062	sk : 2846
ct : 4391	ma : 4963	ok : 4651	wi : 4321	yt : 2100
de : 3630	mi : 4734	or : 7028	wy : 4710	dengl : 479
dc : 3080	mn : 3929	pa : 5474	al : 5702	fraspm : 1210
fl : 6621	ms : 4815	pr : 4781	bc : 4875	
ga : 5942	mo : 4638	ri : 3295	mb : 3023	
hi : 3804	mt : 4800	sc : 5432	nb : 2856	
id : 5129	ne : 3281	sd : 3185	lb : 1433	
il : 5167	nv : 5670	tn : 4900	nf : 2188	
in : 4440	nh : 3635	tx : 8483	nt : 2024	
ia : 3652	nj : 4822	ut : 6041	ns : 2844	

Hình 3: Thống kê số loài cây trên mỗi vùng phân bố.

- Vùng phân bố có ít loài cây nhất là *dengl*, có 479 loài cây, chiếm tỉ lệ 1.38% trên tổng số loài.
- Vùng phân bố có nhiều loài cây nhất là *ca*, có 11676 loài cây, chiếm tỉ lệ 33.57% trên tổng số loài.
- Trung bình một vùng phân bố có khoảng 4370.33 loài cây.

3.3 Chuẩn bị dữ liệu cho thuật giải Apriori

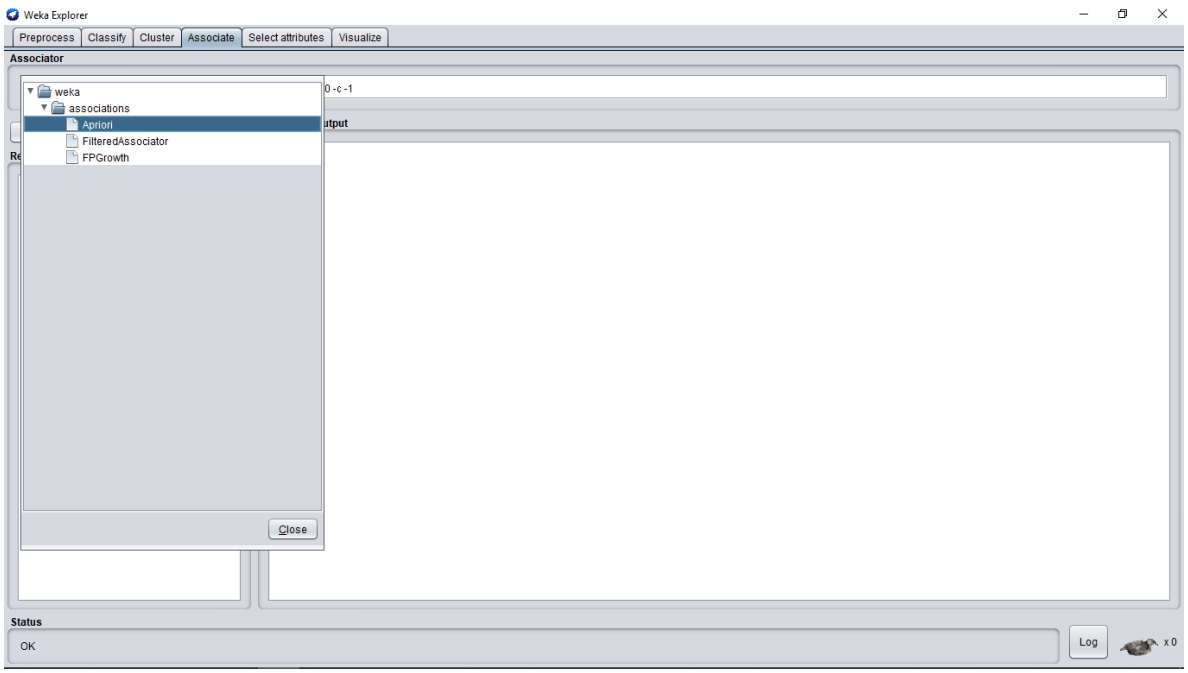
Ta tiến hành thay thế toàn bộ giá trị 'n' thành '?' để loại các giá trị 'n' ra khỏi dữ liệu. Sau đó xóa huộc tính đầu tiên (tên loài cây) không cần thiết trong bài toán khai thác tập phổ biến.



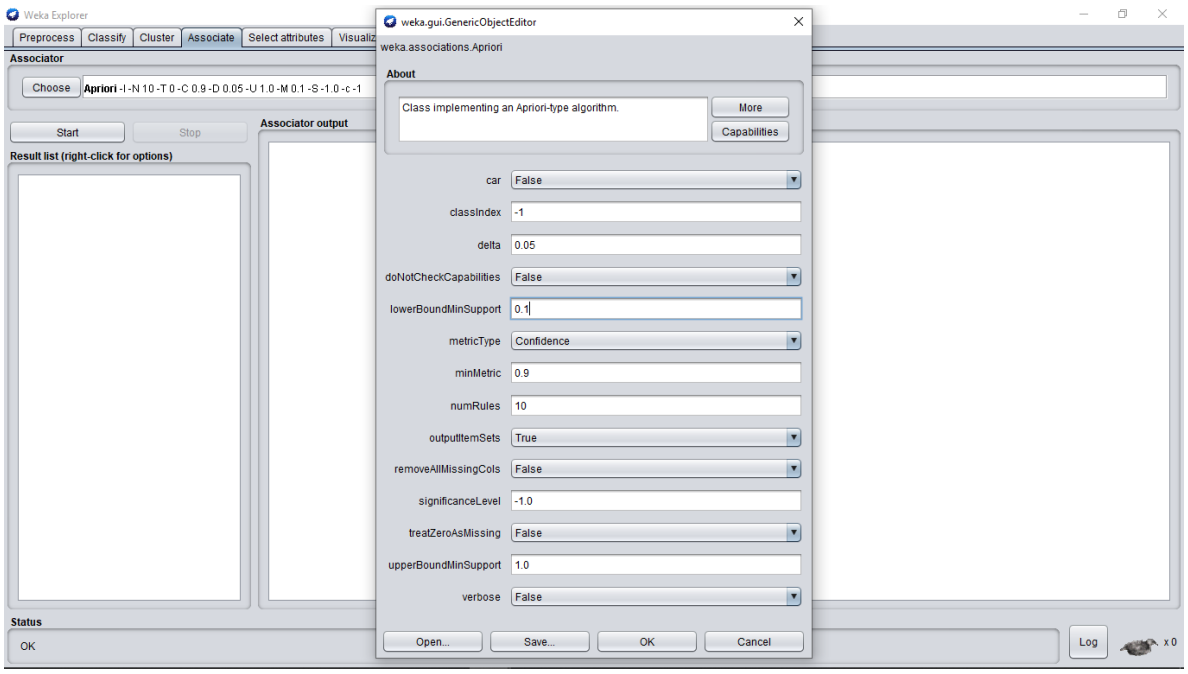
Hình 4: Thay thế 'n' ở các thuộc tính thành '?' và xóa bỏ thuộc tính 'name of plant'.

3.4 Khai thác tập phổ biến

Trong tab *Associate*, ta chọn phương pháp *Apriori*.



Hình 5: Chọn phương pháp Apriori.



Hình 6: Chọn `minSup=0.1`

Sau đó, ta nhấn *Start* để bắt đầu chạy thuật toán.

Bảng 8: Bảng các tập phổ biến được phát sinh.

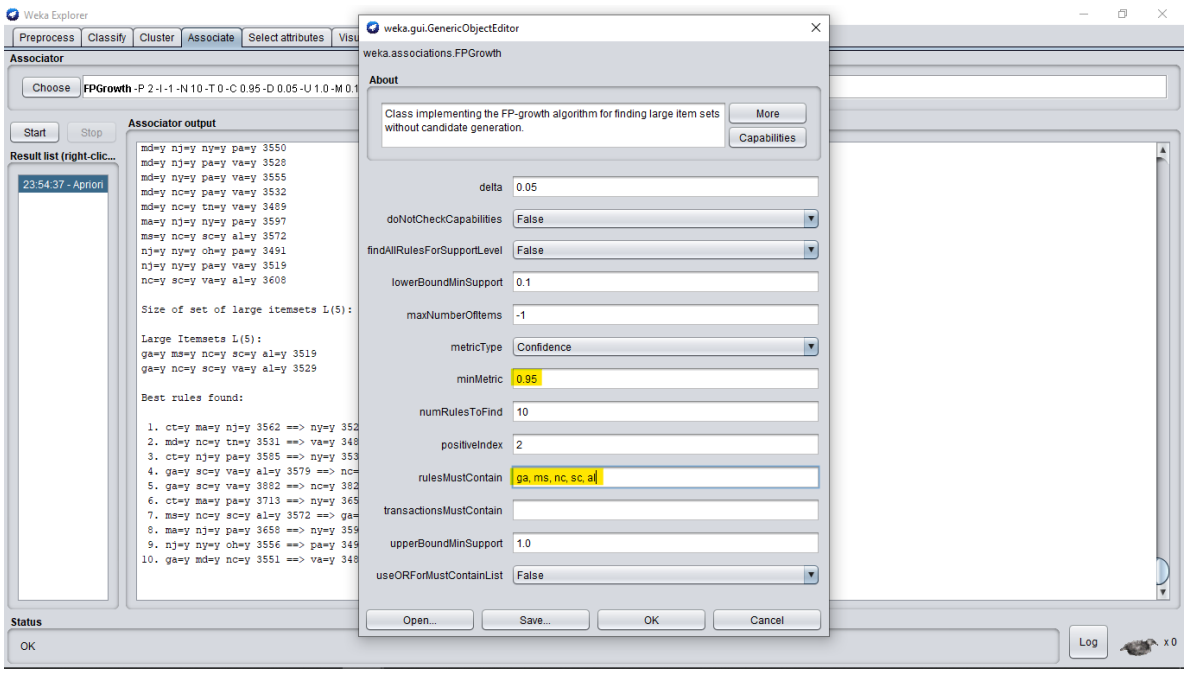
Kích thước	**Số lượng**
1 hạng mục	49
2 hạng mục	167
3 hạng mục	116
4 hạng mục	25
5 hạng mục	2

3.5 Khai thác luật kết hợp

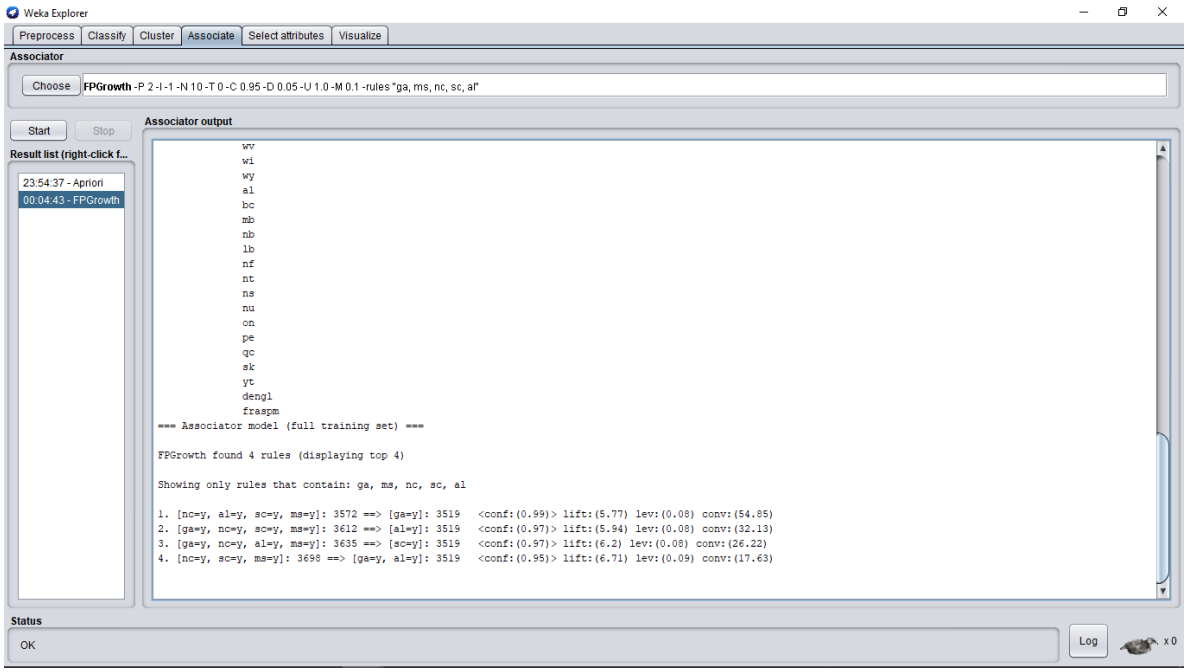
Tập hạng mục phổ biến có kích thước lớn nhất theo kết quả của câu trước là 5. Có tất cả 2 tập:

- `ga=y ms=y nc=y sc=y al=y`
- `ga=y nc=y sc=y va=y al=y`

Ta tiến hành khai thác tất cả luật kết hợp có độ tin cậy (Confidence) từ 0.95 trở lên của 2 tập này.



Hình 7: Lần lượt điền độ tin cậy và tập hạng mục phổ biến cần khai thác vào. Sau đó nhấn *OK* và *Start* để tiến hành khai thác luật.



Hình 8: Kết quả, có 4 luật kết hợp với độ tin cậy từ 0.95 trở lên ở tập hạng mục thứ nhất.

Tương tự với tập hạng mục thứ hai, ta có kết quả sau:

Bảng 9: Bảng kết quả định lượng.

Tập hạng mục phổ biến	**Số lượng luật**
ga=y ms=y nc=y sc=y al=y	4
ga=y nc=y sc=y va=y al=y	4

4 Đánh giá

STT	Nội dung	Hoàn thành
1	Phần lý thuyết.	100%
2	Phần thực hành.	100%
Mức độ hoàn thành tổng thể của bài tập:		**100%**

Tài liệu

[1] Slide lý thuyết.

[2] Trang chủ của Weka: http://www.cs.waikato.ac.nz/ml/weka/

[3] J. Han and M. Kamber, *Data Mining, Concepts and Techniques, Second Edition*