**Eclat**

Đầu tiên, liên kết google colab với google drive:

#Liên kết với google drive

from google.colab import drive

drive.mount('/content/drive/')

Sau đó mở file, đọc file *data.txt* với đường dẫn tới file, đọc từng dòng lưu vào datalist, sau khi đọc xong thì đóng file lại:

#Đọc file

f = open('/content/drive/My Drive/Data/data.txt', 'r')

datalist = f.readlines()

print(datalist)

f.close()

['2\n', 'A C D\n', 'B C E\n', 'A B C E\n', 'B E']

File *data.txt* có định dạng như sau:

2

A C D

B C E

A B C E

B E

Với dòng đầu tiên là *minsup*, các dòng tiếp theo là *items*

Lưu minsup ở dòng đầu tiên vào biến *minsup*:

#Minsup ở dòng đầu tiên

minsup = int(datalist[0])

Lưu các item vào data, với mỗi dòng item, ta dùng hàm split(" ")để chia ra các ký tự và hàm strip() để xóa đi các ký tự xuống dòng (\n)

#Lưu các item vào list

data = []

for i in range(1, len(datalist)):

 tmp = datalist[i].split(" ")

 tmp[-1] = tmp[-1].strip()

 data.append(tmp)

print(data)

[['A', 'C', 'D'], ['B', 'C', 'E'], ['A', 'B', 'C', 'E'], ['B', 'E']]

Tìm các item riêng biệt, lưu vào *item*:

item = []

for i in data:

  for j in i:

    item.append(j)

item = sorted(set(item))

print(item)

['A', 'B', 'C', 'D', 'E']

Tìm các support và tidset tương ứng với từng item. Với mỗi item trong *item* cho chạy vòng lặp với từng phần tử trong *data*, nếu item xuất hiện ở phần tử trong *data*, +1 support và thêm dòng đó vào tidset của item đó. Sau đó lưu vào các list *item\_sup* và *item\_tid*:

item\_sup = []

item\_tid = []

for i in item:

  count = 0

  tid = set()

  for j in range(len(data)):

    if i in data[j]:

      count = count + 1

      tid.add(j+1)

  if count < minsup:

    continue

  item\_sup.append([i, count])

  item\_tid.append([{i}, tid])

print(item\_sup)

print(item\_tid)

[['A', 2], ['B', 3], ['C', 3], ['E', 3]]

[[{'A'}, {1, 3}], [{'B'}, {2, 3, 4}], [{'C'}, {1, 2, 3}], [{'E'}, {2, 3, 4}]]

Tiếp theo là hàm eclat(P, minsup, F) thực hiện giải thuật Eclat.

* Copy tidset *P* vào *tmp* để không ảnh hưởng đến list gốc.
* Chạy vòng lặp với từng item trong tidset *P*
* Thêm vào kết quả *F* cặp item và support tương ứng.
* Tạo list *Pa* rỗng
* Xóa item ra khỏi *tmp* để tạo tổ hợp item với các item còn lại
* Với từng tổ hợp item *Xab*, xác định support *Tab* của nó
* Nếu support của tổ hợp item đó lớn hơn hoặc bằng *minsup*, thêm cặp *Xab*, *Tab* tương ứng và *Pa* đã tạo
* Nếu *Pa* không rỗng, cho chạy đệ quy với *P* thay bằng *Pa*
* Tiếp tục cho đến khi chạy hết các item trong *P* và *Pa* rỗng

#Hàm Eclat

def eclat(P, minsup, F):

  tmp = P.copy()

  for i in P:

    F.append([i[0], len(i[1])])

    Pa = []

    tmp.remove(i)

    #print(tmp)

    for j in tmp:

      Xab = sorted(i[0] | j[0])

      Tab = i[1] & j[1]

      #print(Xab, Tab)

      if len(Tab) >= minsup:

        Pa.append([set(Xab), Tab])

    if len(Pa) != 0:

      eclat(Pa, minsup, F)

Cuối cùng, thực thi hàm eclat và xem kết quả *F*:

#Thực thi

F = []

eclat(item\_tid, minsup, F)

F

[[{'A'}, 2],

[{'A', 'C'}, 2],

[{'B'}, 3],

[{'B', 'C'}, 2],

[{'B', 'C', 'E'}, 2],

[{'B', 'E'}, 3],

[{'C'}, 3],

[{'C', 'E'}, 2],

[{'E'}, 3]]

Thử tính tay bằng cây để so sánh kết quả:

23

BCE

23

CE

234

BE

BC

23

3

AE

13

AC

AB

3

A

13

B

234

123

C

1

D

E

234

Từ cây trên, ta có thể thấy kết quả giống với kết quả của hàm *eclat*.