Data Science Assignment1 : Apriori

2016025078 강덕영

- Apriori

데이터베이스에서 frequent pattern 을 찾는 데이터 마이닝 알고리즘. Assignment1 은 apriori 를 바탕으로 frequent item 을 찾은 후 association rule 를 찾는 것.

- Source code

assignment1 directory: 2022_ITE4005_2016025078/assignment1

main code: apriori.py

input file: input.txt, input2.txt (input2.txt is for a test case)

output file: output.txt

- Summary of algorithm

frequent item 을 찾는 로직은 크게 두 부분으로 나뉩니다. 첫 번째는 데이터베이스(input.txt)에서 길이가 1인 item 들을 뽑습니다. 두 번째는 item 들을 조합해서 item 후보들을 만들어 내고, 그 중 support 가 일정 값 이상인 item 들만 선별합니다. 두 번째 과정을 더 이상 조합할 item 이 존재하지 않을 때까지 반복합니다.

이렇게 찾은 frequent item 들 중, 길이가 2 이상인 item 들에 대해 association rule 을 생성합니다. 그리고 각각의 association rule 에 대해 support 와 confidence 를 계산하고, minimum support 값보다 큰 rule 을 파일(output.txt)에 씁니다.

- **Implementation detail**

<data structure>

1. candidate_support (dictionary, key: tuple, value: int)

후보 item 들의 support 값을 저장하고 있는 딕셔너리.

ex) candidate_support[('1', '2', '3')] = 5 <- support 가 5인 {1, 2, 3} 아이템을 의미

2. frequent_item_support (dictionary, key: tuple, value: int)

frequent item 의 support 값을 저장하고 있는 딕셔너리

3. frequent_item, infrequent_item (list, element: set)

frequent, infrequent item set 을 저장하고 있는 리스트

ex) frequent_item = [{1}, {2}, {1, 2},,,,]

<code flow>

line(1~49): 길이가 1인 item 과 item 의 support 를 찾는 과정

line(50~99): 후보 item 들을 생성 후 pruning 하여 frequent item 을 찾는 과정을 반복.

line(107~122): 길이가 2 이상인 item 들마다 association rule 을 생성하는 과정. item 을 두개의 덩어리로 나누기 위해 combination 를, 두 덩어리로부터 association rule 을 만들기 위해 product 을 사용.

line(129~156): association rule 을 파일에 쓰는 과정

- Specification

rounding: support 와 confidence 는 소수 셋째 자리에서 반올림(ROUND_HALF_UP 설정).

OS: macOS or Ubuntu

코드 실행 가상환경: anaconda

코드 실행 방법: 과제 명세와 동일합니다.

(python apriori.apy 5 input.txt output.txt) or

(python3 apriori.apy 5 input.txt output.txt)