

Programming Assignment #1

2017030328 조지훈

Compilation method and environment

사용 에디터: Visual Studio Code & VIM editor

개발 OS: Ubuntu 20.04.1 LTS (Windows 10 PRO 21H1 버전에서 WSL2 이용)

언어: Python 3.9.7

장치 사양

프로세서 Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz




RAM 16.0GB

시스템 종류 64 비트 운영 체제, x64 기반 프로세서

실행 방법

```
..05_2017030328
> python3 apriori.py 5 input.txt output.txt
/mnt/c/U/m/Pro/D/2022_ite4005_2017030328 > master +3 !1 ?2
```

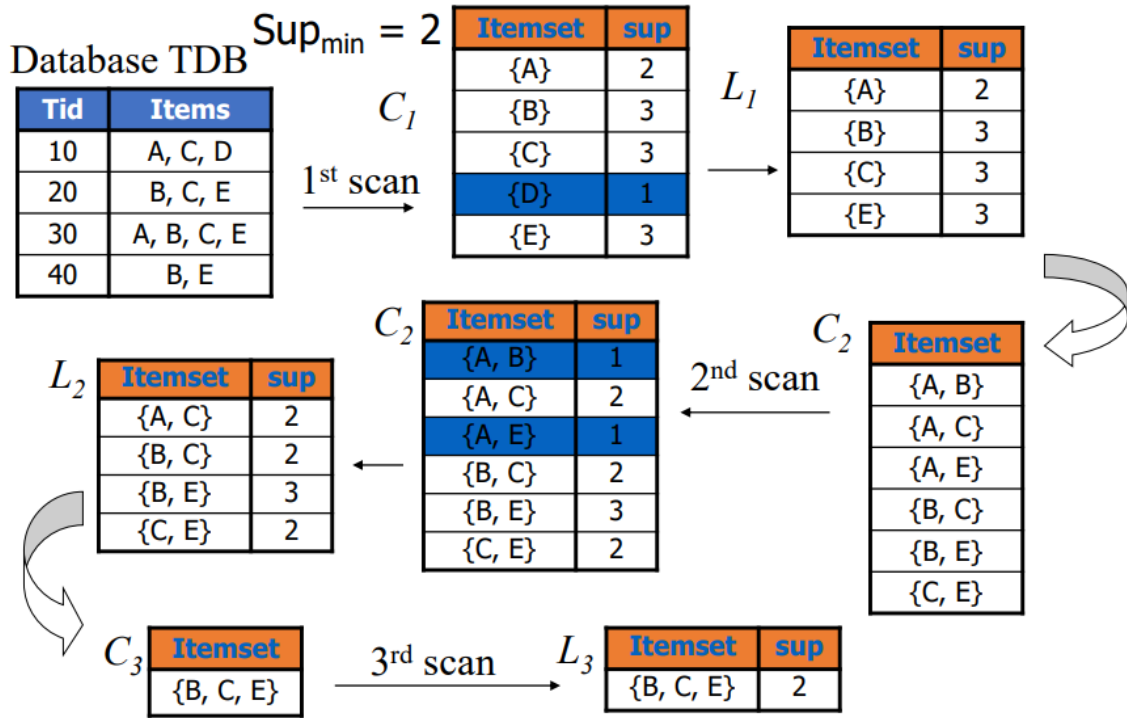
실행은 아래와 동일한 형태로 명세서와 동일하게 3 개의 arguments 를 받아 작동하고 결과물을 다음과 같이 나온다.

apriori.py	minimum support	input file name	output file name
 apriori.py		2022-03-26 오후 12:00	Python 원본 파일
 input.txt		2022-03-17 오전 8:31	텍스트 문서
 output.txt		2022-03-26 오후 12:43	텍스트 문서

{15,1}	{8}	5.40	50.00
{15,8}	{1}	5.40	43.55
{8,1}	{15}	5.40	35.06
{8}	{17,3,16}	5.80	12.83
{16}	{17,8,3}	5.80	13.68
{17}	{8,3,16}	5.80	24.37
{3}	{17,8,16}	5.80	19.33

Algorithm 설명

Apriori 알고리즘은 빈발 항목 집합을 추출하는데 이용하는 알고리즘이다. 이 때 빈발 항목 집합은 최소 지지도 이상의 가지는 항목 집합을 의미한다. 모든 집단에 대해 연산을 진행하면 계산량이 너무 많아지기 때문에 최소 지지도를 정해서 그 이상의 값을 가지는 집단에 대해서만 연관 규칙을 계산하게 된다.



알고리즘의 작동 순서는 위의 그림과 동일하다. Candidate item set 을 구해서 개수를 계산한다. 그 이후 최소 지지도에 도달하지 못하는 item set 을 제외하고 frequent item set 을 구한다. 그리고 이렇게 구해진 frequent item set 을 이용해서 한 단계 긴 Candidate item set 을 구해서 다시 알고리즘을 진행하게 된다. 새로운 Candidate item set 을 만들 수 없을 때까지 반복하게 된다.

Code 설명

해당 코드는 크게 3 가지로 구분이 된다. 2 개의 함수 부분과 그 외 부분으로 구현되어 있다.

Function: apriori

transactions 에서 frequency map 을 만드는 함수이다. 전체 transactions 을 스캔하고 candidate 의 패턴 빈도를 계산하여 빈도가 지원보다 큰 경우 최종적으로 패턴과 빈도를 반환합니다. candidate 가 더 이상 존재하지 않을 때까지 이 함수를 반복한다.

Function: define

apriori 에서 연관 규칙에 대한 패턴과 빈도를 계산하였다면, 해당 함수에서는 연관 규칙에 포함이 되어있는 item_set 과 associative_item_set 에 대하여 각각의 confidence 를 연산하는 역할을 한다. 또한, 해당 함수에서 출력 파일을 구현하기 위해서 rules 과 관련된 내용을 정리해서 반환한다.

Optional parts

7-11 번째 line 은 argument 를 입력 받기 위해서 작성되어 있고, 16 번째 line 에서 반올림이 작동할 수 있도록 설정하였다. 입력 파일을 받아들이기 위해서 50-51 번째 line 이 구현되어 있으며, 출력 형식을 맞추기 위해서 56-58 번째 line 을 작성하였다.