

|  |
| --- |
| Assignment1 |
| #Apriori algorithm |

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **제출일** | 2023. 04. 03 |  | **전공** | 컴퓨터소프트웨어학부 |
| **과목** | 데이터 사이언스 |  | **학번** | 2019041703 |
| **담당교수** | 채동규 교수님 |  | **이름** | 김경민 |

Environment

OS: Windows 10

Runtime: JDK 17.0.2

IDE: IntelliJ Ultimate 2021.03

Usage



apriori.exe {minSupport} {inputfile} {outputfile}

terminal에서 apriori.exe 파일과 input.txt 파일이 있는 경로로 이동하여 위와 같은 명령어를 수행합니다.

Main idea

input.txt에서 transactions를 받아와서 List<Set<String>>에 저장한다

그리고 그 중에서 minimum Support이상인 요소 쌍들을 Set<Set<String>>에 저장한다.

그 후 요소가 한 개 보다 많은 것을 골라 그것 자신을 뺀 서브셋에 대하여 support와 confidence를 구해준다.

1) getFrequentItemsets()

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

모든 아이템의 개수를 세기 위해 Map<String, Integer> itemCounts를 생성한다.

각 트랜잭션을 반복하면서, 각 아이템이 몇 번 등장했는지 itemCounts에 저장한다.

후보 아이템셋을 저장하기 위해 Set<Set<String>> candidateItemsets를 생성한다.

itemCounts의 각 키를 기반으로, 후보 아이템셋을 생성하여 candidateItemsets에 추가하고

while 루프를 실행한다.

각 후보 아이템셋이 포함된 트랜잭션의 개수를 세기 위해 Map<Set<String>, Integer> candidateCounts를 생성한다.

모든 후보 아이템셋에 대해, 모든 트랜잭션을 반복하면서 후보 아이템셋이 포함된 트랜잭션의 개수를 candidateCounts에 저장한다.

후보 아이템셋 중, minimumSupport를 만족하는 아이템셋을 frequentItemsets에 추가한다.

다음 후보 아이템셋을 생성하기 위해 Set<Set<String>> nextCandidateItemsets를 생성한다.

frequentItemsets를 반복하면서, 두 아이템셋을 결합하여 후보 아이템셋을 생성한다.

후보 아이템셋의 길이가 k+1인 경우, nextCandidateItemsets에 추가한다.

k를 1 증가시키고, 다음 루프를 실행한다.

frequentItemsets를 반환한다.

2) calculateSupport()

텍스트이(가) 표시된 사진

자동 생성된 설명

support값을 저장하기 위한 double 변수인 support를 0.0으로 초기화한다.

모든 트랜잭션을 반복하면서, 각 트랜잭션이 itemset을 포함하는 경우 support를 1.0 증가시킨다.

모든 트랜잭션의 수로 support를 나누어, itemset의 support를 계산한다.

3) calculateConfidence

텍스트이(가) 표시된 사진

자동 생성된 설명

두 항목들이 함께 등장한 횟수를 저장하기 위한 double 변수인 jointSupport를 0.0으로 초기화한다.

Antecedent와 Consequent가 모두 포함된 트랜잭션의 수를 계산하여 jointSupport를 증가시킨다.

Antecedent가 포함된 트랜잭션의 수를 저장하기 위한 double 변수인 antecedentSupport를 0.0으로 초기화한다.

모든 트랜잭션을 반복하면서, 각 트랜잭션이 Antecedent를 포함하는 경우 antecedentSupport를 1.0 증가시킨다.

Antecedent가 등장한 트랜잭션의 수가 0보다 큰 경우 confidence를 계산합니다.

4) getSubsets()

텍스트이(가) 표시된 사진

자동 생성된 설명

1) getSubset()

입력된 set을 List<String> itemList으로 변환한다.

부분집합을 저장하기 위한 Set<Set<String>> subsets을 생성한다.

getSubsetsHelper 메서드를 호출하여 부분집합을 생성한다.

2) getSubsetHelper()

현재 인덱스가 itemList의 길이와 같은 경우, currentSubset이 빈 집합이 아닌 경우에 subsets에 추가하고 반환한다.

itemList에서 현재 인덱스의 아이템을 가져온다.

currentSubset을 그대로 사용하여, itemList의 다음 인덱스부터의 모든 부분집합을 추출한다.

currentSubset에 현재 아이템을 추가하고, itemList의 다음 인덱스(index+1)부터의 모든 부분집합을 추출한다.

currentSubset에서 현재 아이템(item)을 제거한다.