

Assignments#1 Apriori

Algorithms

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. In this case calculate confidence with given minimum support ratio and input transactions.

First, create frequency map for candidates. Candidates starts at size 1 elements and gets bigger until there's no candidate anymore. Next step candidates determined by frequency of pattern if greater than minimum support they can be candidates!

After build frequency map, find rules and calculate confidence. Each *pattern* that size bigger than 2 in frequency map separate *element* by number of cases and calculate confidence and support with *element* and *remain* which exclude element from pattern.

Calculating support and confidence is easy. Support is probability that a transaction contains combine of *element* and *remain* (it's means *pattern*, mentioned above). Confidence is conditional probability that a transaction having *element* also contains *remains*.

Implementations

Python is a simple language that is easy enough to understand directly. So it's not difficult to see and understand the code right away. But here are tips for Python beginners.

Function: apriori

This function creates frequency map from transactions. First candidate extraction is just grab unique element in transactions. In this line(L#20), generate unique elements frozenset list from flatten transactions. `scan()` function scan whole transactions and count frequency of pattern in candidates finally return pattern and frequency if frequency greater than support. Until candidate no longer exists repeat these process, `scan()` - generate candidates.

Function: define

This function generate rules with confidence from frequency map generated by `apriori()`. There are two pre-defined lambda functions to make calculations easier. Lambda `f` return frequency of item from frequency map. Lambda `r` return rounded value. It's prevent floating point error which occur frequently in python. Also, below line(L#47-48) do same thing.

Each *pattern* in frequency map which size bigger than 2 can generate rules. Separate *element* and *remain* from pattern and calculate confidence and support for element (L#48, 50).

Optional parts

- L#7-13 for argument parsing support by python default. Accept support, input, output and confidence (as optional).
- L#16 for intuitive rounding. Default python rounding function round up when it is even and round down if odd.
- L#53-54 for parsing input from input file.
- L#59-61 for print output to output file.

Requirements

Nothing, but python3 only

Tested @ python 3.5.2 in windows, macOS and Linux

Run as below

```
python3 apriori.py (min_support) (input_file_pat) (output_file_path) [min_confidence  
as optional]
```