Jacob Shelley

Frequent Item Set Mining with Apriori Algorithm

I implemented my Apriori Algorithm in Python. I used Counters, a special type of dictionary in python that can be found in the “collection” library. Dictionaries work like hash tables. Each element contains a key, which can be used to immediately jump to that element, and a value, which is stored along with its corresponding key.

I used Counters to optimize the run time. The counter works like a hash table. The key would be the item set and the value would be the amount of times it appears. I would take each possible item set from a line and hash them into the counter. The counter would then increment the value by 1. If a key is not found the counter automatically creates a new key:value pair with a value of 1.

There were 1073 items and item sets in total that were above a minimum support value of 500. 569 of those were single numbers, 342 were item sets of 2, 43 were item sets of 4, and 4 were item sets of 5.

From implementing this algorithm I’ve learned how powerful a tool a hash table is. Using hash tables to quickly look up an object had a dramatic impact on the runtime of my program. I also became very conscious of how lean my program was, especially when I was using an implementation with linked lists. I was careful to stop and think about each loop or search iteration I planned to add.