Scott Johnson
Michael Hollman
Cassey Lottman
Darren Johnson

We read the sentiment file from csv and represented sentiments as a dictionary where the word as the key and the sentiment score was the value.

We represented baskets and itemsets the same way as we did for the first part of the project. As you may recall, we represented the item sets we found and their respective counts as a dictionary with keys of tuples containing the item ids (a single item ID on the first pass, a tuple of two items on the second pass, and a tuple of three items on the third pass). The value of the dictionary was the number of times that set had appeared in the processed baskets. For this phase of the project, we also kept track of the basket that each item had appeared in so that we could go back later and get the reviews from that basket.

After finding the frequent itemsets using the apriori algorithm, we constructed a new dictionary to represent the frequent item sets. Instead of having four values where the first three are item numbers and the last number is the number of baskets it appeared in, this dictionary had a 3-item tuple of item numbers as its key and a list of baskets that that item set appeared in. With the data represented in this way, it was very easy to output the data for the report in the format that was defined in the project assignment, and any other format that was useful to our visualization.

For the visualization, we found that it was easiest to only show one itemset, day, and sentiment score per line in a tab-separated format. This allowed us to read the tsv file in d3 and bind each row to a circle on our graph.

Processing time:
File read time: 11,777 milliseconds
Data analysis runtime: 224 ms
Total execution time: 12,001 ms

Complexity Analysis:
The worst case complexity of our algorithm is O(b*i^3 + f) where b is the number of baskets, i is the number of items, and f is the number of frequent itemsets. This complexity is basically the same as the a priori algorithm, with the only additional complexity resulting from the traversal of the frequent itemset list at the end to generate the reports. This complexity is in regards to the entire algorithm, but does not include file I/O nor exhaustively include the (generally negligible) complexity discrepancies introduced by using some of C#'s helper methods, notably IEnumerable extension methods.

Visualization Insight:
We were able to find clear patterns in the data showing that people had much more negative sentiments about their purchases in the beginning of the week, especially Monday, Tuesday, and Wednesday.