# Project 2, Stage 2

*Sawyer Jager, Rees Klintworth, Derek Nordgren, Brad Steiner*

CSCE 378H: Data Modeling - For Dr. Yu - 5.1.2014

## Data Representation

#### Input file

*Apriori/transactions.txt*

Sample format:

```
{116,excellence captivated excuse intelligent playful underestimate damned
missed lame disputing greed disgusted mourning panicked cruelty intimidated }
{186,tender boosting growing failed drunk alarmed charged cheater }
{136,rapturous apologised inaction sulking barrier hurts exploits }{477,weird
vested welcomes melancholy retarded liar troubled hysterics misgiving }
{568,genial absentee huge astonished empathetic hahaha bereave murders dishonest
offends treason loathing interrupted greenwashers disasters }{627,promoted
unified tout steal nonsense stressor }Monday
```

Each line of the file represents one transaction. A { } block represents one item, and lists the itemId the associated customer reviews for that item purchase. The day of purchase is displayed at the end of the line.

Transactions were output in this format so as to minimize the implementation effort of computing total sentiment for itemsets. Items could be easily tokenized from the line, from which sentiment values could be also easily tokenized and compared to a hashmap to compute total sentiment.

#### Output file

*Apriori/project_2_output.txt*

Sample format:

```
(143, 19, 876) 54 0 -44 -47 0 0 0
```

Each line of this file represents one itemset. The ( ) block contains the items that are the parts of the itemset. The following seven integers represent the total sentiment of the itemset for each

day of the week, Sunday - Saturday.

Storing the per day total sentiments in this format enabled more convenient querying of the sentiments by itemset and/or by day in our visualization scripts. We were able to easily manipulate this data using JavaScript and the D3.js library.

## Processing Time

Our input file (generated in C) takes less than one second to generate.

Processing to find triple sets and generate the output file (done in Java) averaged approximately 21.3 seconds. This processing time is of the same order of magnitude as in Stage 1 because our algorithm has the same complexity (as we will discuss below).

However, several new complexities introduced in Stage 2 have slightly increased the processing time:

- changed C input file format to make use of item sentiments - larger input file to read in (*see Apriori/transactions.txt*)
- in addition to reading in a line, now we must parse out and performs calculations required to total sentiments
- *sentiment.csv* must be read into memory
- significant hashing had to be done to make use of *sentiment.csv* in totaling consumer reviews

## Algorithm Complexity

The algorithm complexity for Stage 2 remains unchanged from Stage 1:

> *The complexity was determined to be O(T*I3), where 'I' is the total number of items, and 'T' is the number of transactions. In the average case, the complexity will be much lower due to the Apriori algorithm filtering itemsets with less than three occurrences. Additionally, 'I3' represents the number of combinations of 'I' elements, depending on the iteration of the algorithm. If our algorithm does three passes, to find the minimum support of three, our algorithm will need to make (I choose 3) combinations in the worst case. In our given assignment, the I3 portion was represented by 1000 combinations of 1, 49229 combinations of 2, and 171 combinations of 3.*

## Visualizations

### Interactive Bar Chart

The Interactive Bar Chart visualization, which can be found in the folder named as such, displays a bar for every itemset purchased on a given day. The y-axis on this visualization is the total sentiment for an itemset.

By selecting checkboxes, the user can control which days of the week they view total sentiment for. For example, by selecting Monday, Tuesday, and Wednesday, each bar will represent the total sentiment for an itemset purchased on any or all of those three days. By selecting a single day, the user can view the itemsets purchased on that day. By selecting the entire week, the user can view the total sentiment for any given itemset.

Additionally, the user can query to see in which itemsets a specific item appears via the text field at the top of the page. For example, typing '50' will red-fill any itemset bars that contain the item 50. This is especially useful for examing how often an item occurs in all of the itemsets set to display. It can also show if that item tends to be part of positively or negatively rated itemsets.

The tooltips display on mouseover of the bar displays the item triple that the itemset represents.

### Scatterplot

The Scatterplot visualization, which can be found in the folder named as such, displays a point for every itemset purchased on a given day. The x-axis is the day of week and the y-axis is the total sentiment. This visualization enables the user to quickly determine whether the total sentiment for the itemsets purchased on a given day were more negative or more positive. Additionally, users can quickly determine the relative number of itemsets purchased on a given day.

## Visualization Observations

In our Interactive Bar Chart visualization, we have observed that, in the beginning of the week, consumers typically leave negative reviews for purchases they make early in the week - specifically, Monday, Tuesday, and Wednesday. The total sentiment for any itemset purchased on one of these days is negative.

However, later in the week, consumers typically leave positive reviews for purchases they make. The total sentiment for any itemset purchased on Thursday, Friday, Saturday, or Sunday is positive.