DATASCI W261: Machine Learning at Scale

Gopala Tumuluri's Submission for HW3

Start yarn and hdfs

In [3]:
```
!/usr/local/Cellar/hadoop/2.7.0/sbin/start-yarn.sh
!/usr/local/Cellar/hadoop/2.7.0/sbin/start-dfs.sh
```

```
starting yarn daemons
starting resourcemanager, logging to /usr/local/Cellar/hadoop/2.7.0/li
bexec/logs/yarn-gtumuluri-resourcemanager-GTA2.out
localhost: starting nodemanager, logging to /usr/local/Cellar/hadoop/2
.7.0/libexec/logs/yarn-gtumuluri-nodemanager-GTA2.out
15/09/22 19:53:07 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/Cellar/hadoop/2.7.
0/libexec/logs/hadoop-gtumuluri-namenode-GTA2.out
localhost: starting datanode, logging to /usr/local/Cellar/hadoop/2.7.
0/libexec/logs/hadoop-gtumuluri-datanode-GTA2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/Cellar/hado
op/2.7.0/libexec/logs/hadoop-gtumuluri-secondarynamenode-GTA2.out
15/09/22 19:53:23 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
```

Create a folder in HDFS

In [4]:
```
!hdfs dfs -mkdir -p /user/gtumuluri
```

```
15/09/22 19:53:27 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
```

Problem - 3.0.1

What is a merge sort? Where is it used in Hadoop?

Merge sort is a classic divide and conquer algorithm that is used to efficiently sort a list of items (in ascending or descending order) based on defined ordering criteria. The algorithm has a runtime of $\mathbf{O}(n \times log(n))$ and is the best achievable for any sorting algorithm in the best case. The way the algorithm works is that a given list of items is split into half each time until there is nothing to split (single item) left. Each of these components is then 'merged' one step at a time in such a way that the merge results in ordered (sorted) sub lists until the last merge happens. Given n items in the list, there can be log(n) merges, and the merge operation itself is $\mathbf{O}(n)$, which results in a total run time of $\mathbf{O}(n \times log(n))$ for the algorithm.

In Hadoop, sorting is one of the key steps that happens in the shuffle phase where the shuffle step (implemented as part of the Hadoop framework) 'merges' the output coming through from different mappers into a sorted list of key/value pairs. Sorting is essential to ensures that a given reducer receives all output from all the mappers for a given key.

Problem - 3.0.2

What is a combiner function in the context of Hadoop? Give an example where it can be used and justify why it should be used in the context of this problem.

A combiner - typically an optional program - is an intermediate function/program that is designed to improve the efficiency of a given map/reduce impelementation by reducing network traffic, disk writing and memory usage.

The combiner program acts on output from the mapper, and 'combines' values of same keys into a single key value pair. Without the combiner, the mapper will output one line (key value pair) per occurrence of they key. With the combiner, the reducer will only receive one or few occurrences of they key and its corresponding aggregate / combined value. This process often severely reduces the network traffic between the mapper and the reducer.

The combiner can be implemented in one of three ways: 1) As part of the mapper - called in-mapper combiner. 2) As a separate program that is invoked between a mapper and a reducer. 3) As part of the reducer (although a rare choice).

The combiner is also optional. One condition of a combiner is that it not alter the data contract between the mapper and the reducer. In other words, the reducer should receive the output in a format/format it would have received had the combiner not been present in the first place.

In the problem at hand of computing frequent item sets, we will be producing counts of individual items and co-occurring pairs from the mapper side. In the event items occur numerous times, and also co-occur with other items numerous times, each mapper will output several lines per item and item pair. To efficiently transfer this output to the reducer, it is beneficial to use a combiner that aggregates the counts per item and per item pair, and only sends aggregated counts to the reducer.

Problem - 3.0.3

What is the Hadoop shuffle?

Hadoop Shuffle is the heart of the entire map/reduce framework available as part of Hadoop. It is the machinery that ensures output from mappers is gathered, aggregated, sorted, and delivered to the appropriate reducers. One big contractual commitment the Hadoop map/reduce framework has is that each reducer will see all of the output for a given key from all the mappers. To keep this commitment, the Shuffle process merges the output from mappers, sorts the output in the order of keys, and sends all output for a given key to a single reducer. This whole process is complex given the number of mappers, reducers and data at play, and also given the points of failure in any of these components including the Hadoop management system.

Problem - 3.0.4

What is the Apriori algorithm? Describe an example use in your domain of expertise. Define confidence and lift.

ABOUT APRIORI

The Apriori algorithm is used to find 'frequent itemsets' defined as co-occurring items in a set of baskets. An item can be physical, like one bought in a store, or abstract as a word in a document or care at a hospital.

One of the biggest challenges to find frequent itemsets is that the search space of the combinations is exhaustive - meaning it is of the order $\mathbf{O}(2^n)$. This has to do with the fact that the summation of N choose 1, N choose 2, ... N choose N adds to 2^n. There are no known algorithms that solve such computational complexity for any meaningful size N. The classi traveling salesman problem with some heuristics can perhaps solve for N = 30 or tere abouts.

To overcome this challenge, the Apriori algortithm was developed with a key insight that for an item set to be frequent (above some threshold of occurrences), all its subsets must be frequent also. Applying this criteria will dramatically reduce the brute-force search space necessary to find the frequent item sets of a given size. Using this algorithm, we can first find frequent items of set size 1, which hopefully is a small fraction of total item set size of N (call it k). To find frequent items of set size 2, now we only have to search the combinatorial space of order $\mathbf{O}(k^2)$. Even if k is of $\mathbf{O}(n)$, we still have an $\mathbf{O}(n^2)$ run time (of course, for just 2 itemsets). Repeating this process for larger itemsets will at least ensure a computationally 'plausible' algorithm for finding larger and larger itemsets in polynomial time, unlike the original $\mathbf{O}(2^n)$ algorithm, which is intractable.

EXAMPLE IN MY DOMAIN

I work in the healthcare industry where frequent itemsets can be very applicable in assessing the care a patient receives and how often they receive care for another related issue, or how frequently combination tests are recommended on patients.

CONFIDENCE

In the context of the Apriori algorithm, confidence is defined as the ratio of the frequency of the set occurring relative to the frequency of the subset occurring. For example, if we have milk and orange juice in our frequent itemset, the confidence of this rule will be the ratio: $(number of times milk and orange juice occur)/(number of times milk alone occurs)$. This ratio gives us a better sense for how much we can believe in the rule. Orange juice may happen to appear frequently enough, but if milk's frequency is orders of magnitude higher, this may just be a coincidence and nothing more. Confidence measure ensures we don't put too much faith into such scenarios.

LIFT

Lift is an additional measure to create even stronger association rules beyond confidence measures. A confidence measure can be high just because the component items have very high support. To overcome these, perhaps 'spurious' association rules, *LIFT* is used. It is defined as the ratio of confidences - confidence as measured above divided by the confidence if the items were completely independent. For an association rule A => B, we will have lift as (# A and B) / (# A) whole thing divided by (# B) / (# Baskets).

Problem 3.1 Exploratory Analysis of Data

For this problem, I am exploring unique item counts and basket size frequencies. To accomlish this task, I am using a simple mapper and reducer - with no combiner (in-memory or otherwise). The code outputs each item and the corresponding basket size in which the item was found. The reducer will then be able to count unique items and also the basket size frequencies simultaneously.

An alternative approach could use an in-memory/in-mapper combiner with stripes of unique items and basket size frequencies, which the reducer will aggregate from multiple mappers. I did not use this approach below because such a choice won't generalize to any dataset. For example, a large number of items would need onordinate amount of memory on the mapper side, which may be infeasible. A simple mapper, while not efficient, is guaranteed to generalize. A more specialized mapper/combiner/reducer can be written if more information is available about the dataset.

Problem 3.1.1 - Mapper for Exploratory Analysis of Number of Items and Basket Sizes

In [14]:
```
%%writefile 3.1_mapper.py
#!/usr/bin/python
import sys

# Take input from the standard input, split the line
# and output the item and the corresponding basket
# size it was found in
for line in sys.stdin:
    line = line.strip()
    items = line.split(' ')

    # We don't want to output the basket size for each
    # item found in the basket. Just for the first item.
    # For other items, output -1 for basket size.
    itemnum = 1
    for item in items:
        if itemnum == 1:
            print '%s\t%s' % (item, len(items))
        else:
            print '%s\t%s' % (item, -1)
        itemnum += 1
```

```
Overwriting 3.1_mapper.py
```

Problem 3.1.1 - Reducer for Exploratory Analysis of Number of Items and Basket Sizes

In [15]:
```
%%writefile 3.1_reducer.py
#!/usr/bin/python
import sys
import matplotlib.pyplot as plt

# Initialize basket size frequency dictionary,
# number of items, and a temp variable to detect
# item changes
basket_size_freq = {}
nitems = 0
prev_item = None

# Read standard input, extract item number (key)
# and basket size, and build up a count of unique
# items and the frequency distribution of the basket
# sizes.
for line in sys.stdin:
    line = line.strip()
    item, basket_size = line.split('\t')
```

```
    # If the item changed, increment unique item count
    if item != prev_item:
        nitems += 1
    prev_item = item

    # Increment the number of times a given basket size
    # occurred, after initializing default if the size
    # is found for the first time.
    basket_size_freq.setdefault(int(basket_size), 0)
    basket_size_freq[int(basket_size)] += 1

# Remove -1 from the basket sizes.
# It was just a place holder value.
basket_size_freq.pop(-1)

# Print some key statistics
print '%s\t%s' % ('Number of Unique Items:', nitems)
print '%s\t%s' % ('Largest Basket Size:', max(basket_size_freq))
print '%s\t%s' % ('Smallest Basket Size:', min(basket_size_freq))
print '%s\t%s' % ('Most Frequent Size:',
                  max(basket_size_freq, key = basket_size_freq.get))
print '%s\t%s' % ('Number of Baskets of Most Frequent Size:',
                  basket_size_freq[max(basket_size_freq,
                                       key = basket_size_freq.get)])

# Plot the histogram of the basket sizes (frequency distribution)
plt.hist(basket_size_freq.keys(), len(basket_size_freq), weights = bas
ket_size_freq.values())
plt.xlabel('Basket Size')
plt.ylabel('Number of Baskets')
plt.ylim(0, 3000)
plt.title('Distribution of Basket Sizes')
plt.show()
```

```
Overwriting 3.1_reducer.py
```

In [16]:
```
!hdfs dfs -put ProductPurchaseData.txt /user/gtumuluri
```

```
15/09/22 19:56:21 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
put: `/user/gtumuluri/ProductPurchaseData.txt': File exists
```

In [17]:
```
!hadoop jar /usr/local/Cellar/hadoop/2.7.0/libexec/share/hadoop/tools/
lib/hadoop-streaming-2.7.0.jar -mapper 3.1_mapper.py -reducer 3.1_redu
cer.py -input ProductPurchaseData.txt -output productPurchaseExploreOu
tput
```

```
15/09/22 19:56:35 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
```

```
licable
15/09/22 19:56:36 INFO Configuration.deprecation: session.id is deprec
ated. Instead, use dfs.metrics.session-id
15/09/22 19:56:36 INFO jvm.JvmMetrics: Initializing JVM Metrics with p
rocessName=JobTracker, sessionId=
15/09/22 19:56:36 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics w
ith processName=JobTracker, sessionId= - already initialized
15/09/22 19:56:36 INFO mapred.FileInputFormat: Total input paths to pr
ocess : 1
15/09/22 19:56:36 INFO mapreduce.JobSubmitter: number of splits:1
15/09/22 19:56:36 INFO mapreduce.JobSubmitter: Submitting tokens for j
ob: job_local58429723_0001
15/09/22 19:56:37 INFO mapreduce.Job: The url to track the job: http:/
/localhost:8080/
15/09/22 19:56:37 INFO mapreduce.Job: Running job: job_local58429723_0
001
15/09/22 19:56:37 INFO mapred.LocalJobRunner: OutputCommitter set in c
onfig null
15/09/22 19:56:37 INFO mapred.LocalJobRunner: OutputCommitter is org.a
pache.hadoop.mapred.FileOutputCommitter
15/09/22 19:56:37 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 19:56:37 INFO mapred.LocalJobRunner: Waiting for map tasks
15/09/22 19:56:37 INFO mapred.LocalJobRunner: Starting task: attempt_l
ocal58429723_0001_m_000000_0
15/09/22 19:56:37 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 19:56:37 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcess
Tree currently is supported only on Linux.
15/09/22 19:56:37 INFO mapred.Task:  Using ResourceCalculatorProcessTr
ee : null
15/09/22 19:56:37 INFO mapred.MapTask: Processing split: hdfs://localh
ost:9000/user/gtumuluri/ProductPurchaseData.txt:0+3458517
15/09/22 19:56:37 INFO mapred.MapTask: numReduceTasks: 1
15/09/22 19:56:37 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857
584)
15/09/22 19:56:37 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
15/09/22 19:56:37 INFO mapred.MapTask: soft limit at 83886080
15/09/22 19:56:37 INFO mapred.MapTask: bufstart = 0; bufvoid = 1048576
00
15/09/22 19:56:37 INFO mapred.MapTask: kvstart = 26214396; length = 65
53600
15/09/22 19:56:37 INFO mapred.MapTask: Map output collector class = or
g.apache.hadoop.mapred.MapTask$MapOutputBuffer
15/09/22 19:56:37 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.1_mapper.py]
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.tip.id is dep
recated. Instead, use mapreduce.task.id
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.local.dir is
deprecated. Instead, use mapreduce.cluster.local.dir
```

```
15/09/22 19:56:37 INFO Configuration.deprecation: map.input.file is de
precated. Instead, use mapreduce.map.input.file
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.skip.on is de
precated. Instead, use mapreduce.job.skiprecords
15/09/22 19:56:37 INFO Configuration.deprecation: map.input.length is
deprecated. Instead, use mapreduce.map.input.length
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.work.output.d
ir is deprecated. Instead, use mapreduce.task.output.dir
15/09/22 19:56:37 INFO Configuration.deprecation: map.input.start is d
eprecated. Instead, use mapreduce.map.input.start
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.job.id is dep
recated. Instead, use mapreduce.job.id
15/09/22 19:56:37 INFO Configuration.deprecation: user.name is depreca
ted. Instead, use mapreduce.job.user.name
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.task.is.map i
s deprecated. Instead, use mapreduce.task.ismap
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.task.id is de
precated. Instead, use mapreduce.task.attempt.id
15/09/22 19:56:37 INFO Configuration.deprecation: mapred.task.partitio
n is deprecated. Instead, use mapreduce.task.partition
15/09/22 19:56:37 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
15/09/22 19:56:37 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 19:56:37 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 19:56:37 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 19:56:37 INFO streaming.PipeMapRed: Records R/W=1216/1
15/09/22 19:56:37 INFO streaming.PipeMapRed: R/W/S=10000/120338/0 in:N
A [rec/s] out:NA [rec/s]
15/09/22 19:56:38 INFO mapreduce.Job: Job job_local58429723_0001 runni
ng in uber mode : false
15/09/22 19:56:38 INFO mapreduce.Job:  map 0% reduce 0%
15/09/22 19:56:38 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 19:56:38 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 19:56:38 INFO mapred.LocalJobRunner:
15/09/22 19:56:38 INFO mapred.MapTask: Starting flush of map output
15/09/22 19:56:38 INFO mapred.MapTask: Spilling map output
15/09/22 19:56:38 INFO mapred.MapTask: bufstart = 0; bufend = 4559291;
bufvoid = 104857600
15/09/22 19:56:38 INFO mapred.MapTask: kvstart = 26214396(104857584);
kvend = 24691104(98764416); length = 1523293/6553600
15/09/22 19:56:38 INFO mapred.MapTask: Finished spill 0
15/09/22 19:56:38 INFO mapred.Task: Task:attempt_local58429723_0001_m_
000000_0 is done. And is in the process of committing
15/09/22 19:56:39 INFO mapred.LocalJobRunner: Records R/W=1216/1
15/09/22 19:56:39 INFO mapred.Task: Task 'attempt_local58429723_0001_m
_000000_0' done.
15/09/22 19:56:39 INFO mapred.LocalJobRunner: Finishing task: attempt_
```

```
local58429723_0001_m_000000_0
15/09/22 19:56:39 INFO mapred.LocalJobRunner: map task executor comple
te.
15/09/22 19:56:39 INFO mapred.LocalJobRunner: Waiting for reduce tasks
15/09/22 19:56:39 INFO mapred.LocalJobRunner: Starting task: attempt_l
ocal58429723_0001_r_000000_0
15/09/22 19:56:39 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 19:56:39 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcess
Tree currently is supported only on Linux.
15/09/22 19:56:39 INFO mapred.Task:  Using ResourceCalculatorProcessTr
ee : null
15/09/22 19:56:39 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@34ebd7e
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: MergerManager: memoryL
imit=334338464, maxSingleShuffleLimit=83584616, mergeThreshold=2206633
92, ioSortFactor=10, memToMemMergeOutputsThreshold=10
15/09/22 19:56:39 INFO reduce.EventFetcher: attempt_local58429723_0001
_r_000000_0 Thread started: EventFetcher for fetching Map Completion E
vents
15/09/22 19:56:39 INFO reduce.LocalFetcher: localfetcher#1 about to sh
uffle output of map attempt_local58429723_0001_m_000000_0 decomp: 5320
941 len: 5320945 to MEMORY
15/09/22 19:56:39 INFO reduce.InMemoryMapOutput: Read 5320941 bytes fr
om map-output for attempt_local58429723_0001_m_000000_0
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: closeInMemoryFile -> m
ap-output of size: 5320941, inMemoryMapOutputs.size() -> 1, commitMemo
ry -> 0, usedMemory ->5320941
15/09/22 19:56:39 INFO reduce.EventFetcher: EventFetcher is interrupte
d.. Returning
15/09/22 19:56:39 INFO mapred.LocalJobRunner: 1 / 1 copied.
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: finalMerge called with
1 in-memory map-outputs and 0 on-disk map-outputs
15/09/22 19:56:39 INFO mapred.Merger: Merging 1 sorted segments
15/09/22 19:56:39 INFO mapred.Merger: Down to the last merge-pass, wit
h 1 segments left of total size: 5320930 bytes
15/09/22 19:56:39 INFO mapreduce.Job:  map 100% reduce 0%
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: Merged 1 segments, 532
0941 bytes to disk to satisfy reduce memory limit
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: Merging 1 files, 53209
45 bytes from disk
15/09/22 19:56:39 INFO reduce.MergeManagerImpl: Merging 0 segments, 0
bytes from memory into reduce
15/09/22 19:56:39 INFO mapred.Merger: Merging 1 sorted segments
15/09/22 19:56:39 INFO mapred.Merger: Down to the last merge-pass, wit
h 1 segments left of total size: 5320930 bytes
15/09/22 19:56:39 INFO mapred.LocalJobRunner: 1 / 1 copied.
15/09/22 19:56:39 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.1_reducer.py]
15/09/22 19:56:39 INFO Configuration.deprecation: mapred.job.tracker i
```

```
s deprecated. Instead, use mapreduce.jobtracker.address
15/09/22 19:56:39 INFO Configuration.deprecation: mapred.map.tasks is
deprecated. Instead, use mapreduce.job.maps
15/09/22 19:56:39 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
15/09/22 19:56:39 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 19:56:39 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 19:56:39 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 19:56:39 INFO streaming.PipeMapRed: R/W/S=10000/0/0 in:NA [re
c/s] out:NA [rec/s]
15/09/22 19:56:40 INFO streaming.PipeMapRed: R/W/S=100000/0/0 in:10000
0=100000/1 [rec/s] out:0=0/1 [rec/s]
15/09/22 19:56:40 INFO streaming.PipeMapRed: R/W/S=200000/0/0 in:20000
0=200000/1 [rec/s] out:0=0/1 [rec/s]
15/09/22 19:56:41 INFO streaming.PipeMapRed: R/W/S=300000/0/0 in:30000
0=300000/1 [rec/s] out:0=0/1 [rec/s]
15/09/22 19:56:45 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 19:56:45 INFO mapreduce.Job:  map 100% reduce 100%
15/09/22 19:57:21 INFO streaming.PipeMapRed: Records R/W=380824/1
15/09/22 19:57:21 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 19:57:21 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 19:57:22 INFO mapred.Task: Task:attempt_local58429723_0001_r_
000000_0 is done. And is in the process of committing
15/09/22 19:57:22 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 19:57:22 INFO mapred.Task: Task attempt_local58429723_0001_r_
000000_0 is allowed to commit now
15/09/22 19:57:22 INFO output.FileOutputCommitter: Saved output of tas
k 'attempt_local58429723_0001_r_000000_0' to hdfs://localhost:9000/use
r/gtumuluri/productPurchaseExploreOutput/_temporary/0/task_local584297
23_0001_r_000000
15/09/22 19:57:22 INFO mapred.LocalJobRunner: Records R/W=380824/1 > r
educe
15/09/22 19:57:22 INFO mapred.Task: Task 'attempt_local58429723_0001_r
_000000_0' done.
15/09/22 19:57:22 INFO mapred.LocalJobRunner: Finishing task: attempt_
local58429723_0001_r_000000_0
15/09/22 19:57:22 INFO mapred.LocalJobRunner: reduce task executor com
plete.
15/09/22 19:57:23 INFO mapreduce.Job: Job job_local58429723_0001 compl
eted successfully
15/09/22 19:57:23 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=10854042
                FILE: Number of bytes written=16756355
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

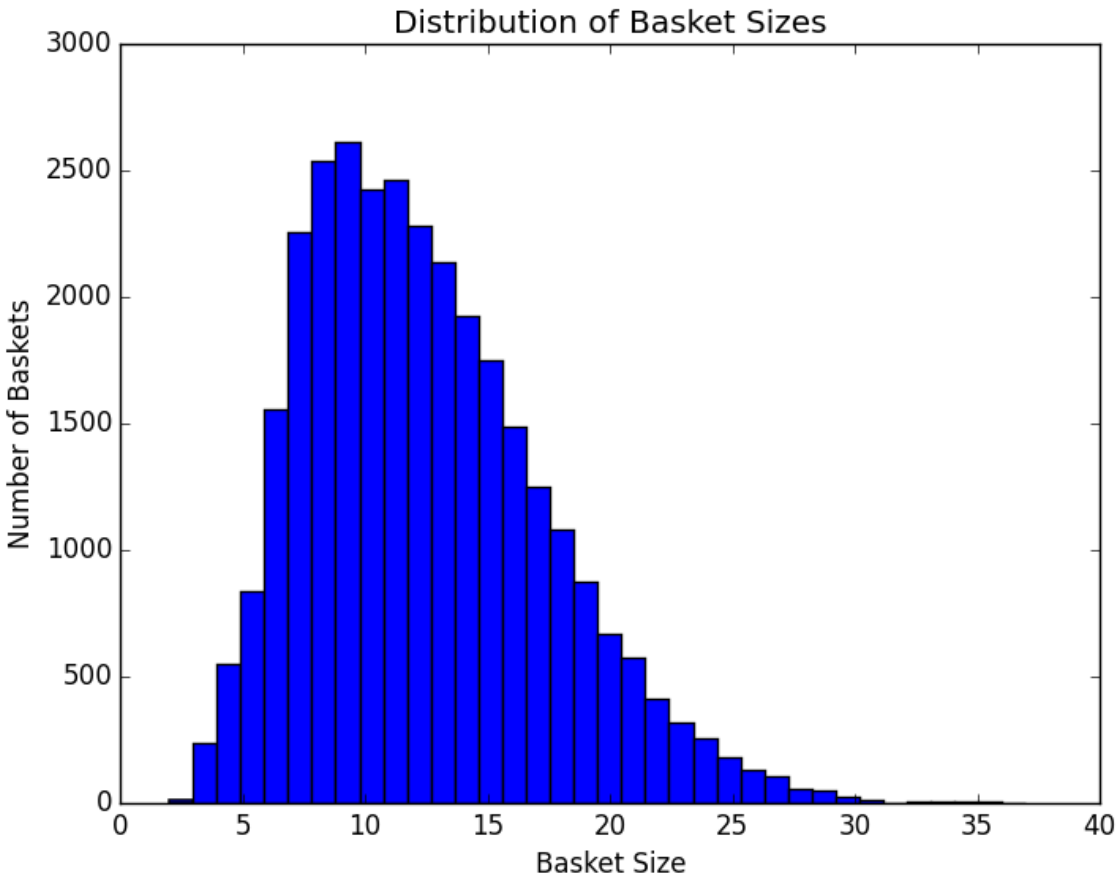
```
                HDFS: Number of bytes read=6917034
                HDFS: Number of bytes written=146
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=31101
                Map output records=380824
                Map output bytes=4559291
                Map output materialized bytes=5320945
                Input split bytes=112
                Combine input records=0
                Combine output records=0
                Reduce input groups=12592
                Reduce shuffle bytes=5320945
                Reduce input records=380824
                Reduce output records=5
                Spilled Records=761648
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=569376768
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=3458517
        File Output Format Counters
                Bytes Written=146
15/09/22 19:57:23 INFO streaming.StreamJob: Output directory: productP
urchaseExploreOutput
```

In [18]:
```
!hdfs dfs -cat productPurchaseExploreOutput/part-00000
```

```
15/09/22 19:57:31 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
Number of Unique Items: 12592
Largest Basket Size:    37
Smallest Basket Size:   2
Most Frequent Size:     9
Number of Baskets of Most Frequent Size:        2611
```

In [19]:
```
from IPython.display import Image
Image(filename='figure_1.png')
```

Out[19]:



Problem 3.2 Apriori 1-Itemsets, 2-Itemsets and Top 5 Rules

To find 1- and 2-itemsets and the rules, I am using a mapper, a combiner, and a reducer. The mapper uses 'pairs' and order inversion concepts to communicate its output to the combiner and the reducer (if the combiner does not run or is not used). The mapper outputs a two-tuple as the key, and 1 as the value. The two-tuple is the pair of items co-occurring in the lexicographic order. To leverage order inversion to count 1-itemsets, a two-tuple of each item with special peer word '*' is also output by the mapper.

Note that the output of the mapper is further optimized by avoiding outputting co-occurrences in the opposite lexicographical order since the problem statement specifically requires lexical ordering of two-item sets and their corresponding confidence computations.

For example, the mapper will only output (a, b), 1 and never (b, a), 1.

The combiner in this case is an external combiner, and it simply aggregates counts of identical keys - both order inversions, and co-occurring item pairs. The combiner is used in the output below, but the reducer can function and provide accurate output even if the combiner fails to run.

Problem 3.2.1 - Mapper to Find 1- and 2-Itemsets, and Top 5 Rules

In [20]:
```
%%writefile 3.2_pairs_mapper.py
#!/usr/bin/python

import sys

# Take input from the standard input with each line
# representing a basket. Split the basket and output
# the order inversion key with count 1 as value, and also
# co-occurring terms with count 1 as value. Only output
# lexicographically ordered pairs.
for line in sys.stdin:
    line = line.strip()
    items = line.split(' ')

    # Sort the items so we can benefit from
    # lexical ordering.
    items = sorted(items)

    # Double loop to output order inversions in
    # first one, and co-occurring terms in the second.
    for item in items:
        print '%s\t%s' % ((item, '*'), 1)
        # Sorted order means, we don't need to
        # iterate over whole list. Only current index
        # to last.
        for peer in items[items.index(item) + 1:]:
            print '%s\t%s' % ((item, peer), 1)
```

```
Overwriting 3.2_pairs_mapper.py
```

Problem 3.2.2 - Combiner to Find 1- and 2-Itemsets, and Top 5 Rules

In [21]:
```
%%writefile 3.2_pairs_combiner.py
#!/usr/bin/python
import sys
import ast

# Temp variables to track
# key changes and new counts
prev_key = None
count = 0

# Read output from standard input, aggregate counts
# of common keys (including order inversions), and
# output back the key and aggregate count.
for line in sys.stdin:
    line = line.strip()
    items = line.split('\t')

    key = ast.literal_eval(items[0])
    value = int(items[1])
    if key == prev_key:
        count += value
    else:
        # Key changed, time to output the previous one.
        if prev_key is not None:
            print '%s\t%s' % (prev_key, count)
        count = value

    prev_key = key

# Don't forget to output the last key that we just counted.
print '%s\t%s' % (prev_key, count)
```

```
Overwriting 3.2_pairs_combiner.py
```

Problem 3.2.3 - Reducer to Find 1- and 2-Itemsets, and Top 5 Rules

In [22]:
```
%%writefile 3.2_pairs_reducer.py
#!/usr/bin/python
import sys
import ast

# Maintain an item frequency map using dict.
item_freq = {}


# Temp variable to track key changes.
prev_item = None
```

```
# Read standard input, extract the 2-tuple and the
# count of occurrences, and aggregte them in the
# above dict.
for line in sys.stdin:
    line = line.strip()
    items = line.split('\t')

    # Convert two-tuple string into a
    # python tuple and the count value.
    key = ast.literal_eval(items[0])
    value = int(items[1])

    # If key is same, keep adding up the count.
    # Otherwise, start a new counter after initializing
    # a new key in the dict.
    if key == prev_item:
        item_freq[key[0]][key[1]] += value
    else:
        item_freq.setdefault(key[0], {})
        item_freq[key[0]][key[1]] = value

    # Keep track of key changes.
    prev_item = key

# Walk through the item frequency map and
# build a list of frequent 1-itemsets (items
# with 100 or more occurrences)
freq_itemsets_one = []
for key in item_freq:
    if item_freq[key]['*'] >= 100:
        freq_itemsets_one.append(key)
print 'Number of frequent 1-item sets:', len(freq_itemsets_one)

# Walk through the 1-item sets, and review the full
# n-squared possible two item sets and see if they
# have sufficient frequncy of co-occurrence (100) and
# add them to the 2-itemset if they do.
freq_itemsets_two = []
for item in freq_itemsets_one:
    for peer in freq_itemsets_one:
        if (item == peer or
            peer not in item_freq[item] or
            item_freq[item][peer] < 100):
            continue
        if ((item, peer) in freq_itemsets_two or
           (peer, item) in freq_itemsets_two):
            continue

        if item < peer:
```

```
            freq_itemsets_two.append((item, peer, item_freq[item][peer
]))
        else:
            freq_itemsets_two.append((peer, item, item_freq[peer][item
]))
print 'Number of frequent 2-item sets:', len(freq_itemsets_two)

# Walk through the 2-itemset list and compute confidence.
# Remember, the item pairs are already in the lexical order
# since our mapper did that job for us.
freq_itemsets_two_conf = []
for item in freq_itemsets_two:
    conf = round(item[2] * 1.0 / item_freq[item[0]]['*'], 2)
    freq_itemsets_two_conf.append((item[0], item[1], conf))

# Sort the 2-itemset confidence map by confidence score first in
# descending order, and by item ID next, in lexical order.
freq_itemsets_two_conf = sorted(freq_itemsets_two_conf,
                                key = lambda(x): (-x[2], x[0]))

# Output the first five entries in the list.
print '\nTop Five Rules with Confidence Scores:'
for item in freq_itemsets_two_conf[:5]:
    print item
```

```
Overwriting 3.2_pairs_reducer.py
```

In [23]:
```
!hadoop jar /usr/local/Cellar/hadoop/2.7.0/libexec/share/hadoop/tools/
lib/hadoop-streaming-2.7.0.jar -mapper 3.2_pairs_mapper.py -combiner 3
.2_pairs_combiner.py -reducer 3.2_pairs_reducer.py -input ProductPurch
aseData.txt -output oneAndTwoItemsetsOutput
```

```
15/09/22 20:00:32 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
15/09/22 20:00:32 INFO Configuration.deprecation: session.id is deprec
ated. Instead, use dfs.metrics.session-id
15/09/22 20:00:32 INFO jvm.JvmMetrics: Initializing JVM Metrics with p
rocessName=JobTracker, sessionId=
15/09/22 20:00:32 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics w
ith processName=JobTracker, sessionId= - already initialized
15/09/22 20:00:33 INFO mapred.FileInputFormat: Total input paths to pr
ocess : 1
15/09/22 20:00:33 INFO mapreduce.JobSubmitter: number of splits:1
15/09/22 20:00:33 INFO mapreduce.JobSubmitter: Submitting tokens for j
ob: job_local2065773600_0001
15/09/22 20:00:33 INFO mapreduce.Job: The url to track the job: http:/
/localhost:8080/
15/09/22 20:00:33 INFO mapred.LocalJobRunner: OutputCommitter set in c
onfig null
```

```
15/09/22 20:00:33 INFO mapred.LocalJobRunner: OutputCommitter is org.a
pache.hadoop.mapred.FileOutputCommitter
15/09/22 20:00:33 INFO mapreduce.Job: Running job: job_local2065773600
_0001
15/09/22 20:00:33 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 20:00:33 INFO mapred.LocalJobRunner: Waiting for map tasks
15/09/22 20:00:33 INFO mapred.LocalJobRunner: Starting task: attempt_l
ocal2065773600_0001_m_000000_0
15/09/22 20:00:33 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 20:00:33 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcess
Tree currently is supported only on Linux.
15/09/22 20:00:33 INFO mapred.Task:  Using ResourceCalculatorProcessTr
ee : null
15/09/22 20:00:33 INFO mapred.MapTask: Processing split: hdfs://localh
ost:9000/user/gtumuluri/ProductPurchaseData.txt:0+3458517
15/09/22 20:00:33 INFO mapred.MapTask: numReduceTasks: 1
15/09/22 20:00:34 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857
584)
15/09/22 20:00:34 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
15/09/22 20:00:34 INFO mapred.MapTask: soft limit at 83886080
15/09/22 20:00:34 INFO mapred.MapTask: bufstart = 0; bufvoid = 1048576
00
15/09/22 20:00:34 INFO mapred.MapTask: kvstart = 26214396; length = 65
53600
15/09/22 20:00:34 INFO mapred.MapTask: Map output collector class = or
g.apache.hadoop.mapred.MapTask$MapOutputBuffer
15/09/22 20:00:34 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.2_pairs_mapper.py]
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.tip.id is dep
recated. Instead, use mapreduce.task.id
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.local.dir is
deprecated. Instead, use mapreduce.cluster.local.dir
15/09/22 20:00:34 INFO Configuration.deprecation: map.input.file is de
precated. Instead, use mapreduce.map.input.file
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.skip.on is de
precated. Instead, use mapreduce.job.skiprecords
15/09/22 20:00:34 INFO Configuration.deprecation: map.input.length is
deprecated. Instead, use mapreduce.map.input.length
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.work.output.d
ir is deprecated. Instead, use mapreduce.task.output.dir
15/09/22 20:00:34 INFO Configuration.deprecation: map.input.start is d
eprecated. Instead, use mapreduce.map.input.start
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.job.id is dep
recated. Instead, use mapreduce.job.id
15/09/22 20:00:34 INFO Configuration.deprecation: user.name is depreca
ted. Instead, use mapreduce.job.user.name
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.task.is.map i
s deprecated. Instead, use mapreduce.task.ismap
```

```
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.task.id is de
precated. Instead, use mapreduce.task.attempt.id
15/09/22 20:00:34 INFO Configuration.deprecation: mapred.task.partitio
n is deprecated. Instead, use mapreduce.task.partition
15/09/22 20:00:34 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
15/09/22 20:00:34 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 20:00:34 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 20:00:34 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 20:00:34 INFO streaming.PipeMapRed: Records R/W=1216/1
15/09/22 20:00:34 INFO mapreduce.Job: Job job_local2065773600_0001 run
ning in uber mode : false
15/09/22 20:00:34 INFO mapreduce.Job:  map 0% reduce 0%
15/09/22 20:00:36 INFO streaming.PipeMapRed: R/W/S=10000/974771/0 in:5
000=10000/2 [rec/s] out:487385=974771/2 [rec/s]
15/09/22 20:00:39 INFO mapred.MapTask: Spilling map output
15/09/22 20:00:39 INFO mapred.MapTask: bufstart = 0; bufend = 52006641
; bufvoid = 104857600
15/09/22 20:00:39 INFO mapred.MapTask: kvstart = 26214396(104857584);
kvend = 18244540(72978160); length = 7969857/6553600
15/09/22 20:00:39 INFO mapred.MapTask: (EQUATOR) 59995777 kvi 14998940
(59995760)
15/09/22 20:00:39 INFO mapred.LocalJobRunner: Records R/W=1216/1 > map
15/09/22 20:00:40 INFO mapreduce.Job:  map 48% reduce 0%
15/09/22 20:00:42 INFO mapred.LocalJobRunner: Records R/W=1216/1 > map
15/09/22 20:00:43 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.2_pairs_combiner.py
]
15/09/22 20:00:43 INFO Configuration.deprecation: mapred.skip.map.auto
.incr.proc.count is deprecated. Instead, use mapreduce.map.skip.proc-c
ount.auto-incr
15/09/22 20:00:43 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
15/09/22 20:00:43 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 20:00:43 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 20:00:43 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 20:00:43 INFO streaming.PipeMapRed: Records R/W=9896/1
15/09/22 20:00:43 INFO streaming.PipeMapRed: R/W/S=10000/2432/0 in:NA
[rec/s] out:NA [rec/s]
15/09/22 20:00:43 INFO mapreduce.Job:  map 58% reduce 0%
15/09/22 20:00:45 INFO streaming.PipeMapRed: R/W/S=100000/34618/0 in:5
0000=100000/2 [rec/s] out:17309=34618/2 [rec/s]
15/09/22 20:00:45 INFO mapred.LocalJobRunner: Records R/W=9896/1 > map
15/09/22 20:00:47 INFO streaming.PipeMapRed: R/W/S=200000/73487/0 in:5
```

```
0000=200000/4 [rec/s] out:18371=73487/4 [rec/s]
15/09/22 20:00:48 INFO mapred.LocalJobRunner: Records R/W=9896/1 > map
15/09/22 20:00:49 INFO streaming.PipeMapRed: R/W/S=300000/109311/0 in:
50000=300000/6 [rec/s] out:18218=109311/6 [rec/s]
15/09/22 20:00:51 INFO mapred.LocalJobRunner: Records R/W=9896/1 > map
15/09/22 20:00:52 INFO streaming.PipeMapRed: R/W/S=400000/137841/0 in:
50000=400000/8 [rec/s] out:17230=137841/8 [rec/s]
15/09/22 20:00:53 INFO streaming.PipeMapRed: Records R/W=470166/149349
15/09/22 20:00:54 INFO streaming.PipeMapRed: R/W/S=500000/164546/0 in:
50000=500000/10 [rec/s] out:16454=164546/10 [rec/s]
15/09/22 20:00:54 INFO mapred.LocalJobRunner: Records R/W=470166/14934
9 > map
15/09/22 20:00:56 INFO streaming.PipeMapRed: R/W/S=600000/190039/0 in:
50000=600000/12 [rec/s] out:15836=190039/12 [rec/s]
15/09/22 20:00:57 INFO mapred.LocalJobRunner: Records R/W=470166/14934
9 > map
15/09/22 20:00:58 INFO streaming.PipeMapRed: R/W/S=700000/221595/0 in:
46666=700000/15 [rec/s] out:14773=221595/15 [rec/s]
15/09/22 20:01:00 INFO streaming.PipeMapRed: R/W/S=800000/251941/0 in:
47058=800000/17 [rec/s] out:14820=251941/17 [rec/s]
15/09/22 20:01:00 INFO mapred.LocalJobRunner: Records R/W=470166/14934
9 > map
15/09/22 20:01:02 INFO streaming.PipeMapRed: R/W/S=900000/285340/0 in:
47368=900000/19 [rec/s] out:15017=285340/19 [rec/s]
15/09/22 20:01:03 INFO streaming.PipeMapRed: Records R/W=936739/295660
15/09/22 20:01:03 INFO mapred.LocalJobRunner: Records R/W=936739/29566
0 > map
15/09/22 20:01:04 INFO streaming.PipeMapRed: R/W/S=1000000/319354/0 in
:47619=1000000/21 [rec/s] out:15207=319354/21 [rec/s]
15/09/22 20:01:06 INFO streaming.PipeMapRed: R/W/S=1100000/356423/0 in
:47826=1100000/23 [rec/s] out:15496=356423/23 [rec/s]
15/09/22 20:01:06 INFO mapred.LocalJobRunner: Records R/W=936739/29566
0 > map
15/09/22 20:01:09 INFO streaming.PipeMapRed: R/W/S=1200000/389221/0 in
:48000=1200000/25 [rec/s] out:15568=389221/25 [rec/s]
15/09/22 20:01:09 INFO mapred.LocalJobRunner: Records R/W=936739/29566
0 > map
15/09/22 20:01:11 INFO streaming.PipeMapRed: R/W/S=1300000/428165/0 in
:48148=1300000/27 [rec/s] out:15857=428165/27 [rec/s]
15/09/22 20:01:12 INFO mapred.LocalJobRunner: Records R/W=936739/29566
0 > map
15/09/22 20:01:13 INFO streaming.PipeMapRed: R/W/S=1400000/462216/0 in
:48275=1400000/29 [rec/s] out:15938=462216/29 [rec/s]
15/09/22 20:01:13 INFO streaming.PipeMapRed: Records R/W=1411398/46465
5
15/09/22 20:01:15 INFO streaming.PipeMapRed: R/W/S=1500000/496308/0 in
:46875=1500000/32 [rec/s] out:15509=496308/32 [rec/s]
15/09/22 20:01:15 INFO mapred.LocalJobRunner: Records R/W=1411398/4646
55 > map
15/09/22 20:01:17 INFO streaming.PipeMapRed: R/W/S=1600000/529179/0 in
```

```
:47058=1600000/34 [rec/s] out:15564=529179/34 [rec/s]
15/09/22 20:01:18 INFO mapred.LocalJobRunner: Records R/W=1411398/4646
55 > map
15/09/22 20:01:19 INFO streaming.PipeMapRed: R/W/S=1700000/560854/0 in
:47222=1700000/36 [rec/s] out:15579=560854/36 [rec/s]
15/09/22 20:01:21 INFO streaming.PipeMapRed: R/W/S=1800000/589507/0 in
:47368=1800000/38 [rec/s] out:15513=589507/38 [rec/s]
15/09/22 20:01:21 INFO mapred.LocalJobRunner: Records R/W=1411398/4646
55 > map
15/09/22 20:01:23 INFO streaming.PipeMapRed: Records R/W=1889061/61392
8
15/09/22 20:01:23 INFO streaming.PipeMapRed: R/W/S=1900000/619436/0 in
:47500=1900000/40 [rec/s] out:15485=619436/40 [rec/s]
15/09/22 20:01:24 INFO mapred.LocalJobRunner: Records R/W=1889061/6139
28 > map
15/09/22 20:01:25 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 20:01:25 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 20:01:25 INFO mapred.MapTask: Finished spill 0
15/09/22 20:01:25 INFO mapred.MapTask: (RESET) equator 59995777 kv 149
98940(59995760) kvi 13008708(52034832)
15/09/22 20:01:25 INFO streaming.PipeMapRed: Records R/W=26917/2490024
15/09/22 20:01:27 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 20:01:27 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 20:01:27 INFO mapred.LocalJobRunner: Records R/W=1889061/6139
28 > map
15/09/22 20:01:27 INFO mapred.MapTask: Starting flush of map output
15/09/22 20:01:27 INFO mapred.MapTask: Spilling map output
15/09/22 20:01:27 INFO mapred.MapTask: bufstart = 59995777; bufend = 8
4025236; bufvoid = 104857600
15/09/22 20:01:27 INFO mapred.MapTask: kvstart = 14998940(59995760); k
vend = 11309268(45237072); length = 3689673/6553600
15/09/22 20:01:27 INFO mapred.LocalJobRunner: Records R/W=26917/249002
4 > sort
15/09/22 20:01:28 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.2_pairs_combiner.py
]
15/09/22 20:01:28 INFO Configuration.deprecation: mapred.skip.reduce.a
uto.incr.proc.count is deprecated. Instead, use mapreduce.reduce.skip.
proc-count.auto-incr
15/09/22 20:01:28 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
15/09/22 20:01:28 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 20:01:28 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 20:01:28 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 20:01:28 INFO streaming.PipeMapRed: Records R/W=9907/1
15/09/22 20:01:28 INFO streaming.PipeMapRed: R/W/S=10000/2431/0 in:NA
[rec/s] out:NA [rec/s]
```

```
15/09/22 20:01:28 INFO mapreduce.Job:  map 67% reduce 0%
15/09/22 20:01:30 INFO streaming.PipeMapRed: R/W/S=100000/47422/0 in:5
0000=100000/2 [rec/s] out:23711=47422/2 [rec/s]
15/09/22 20:01:30 INFO mapred.LocalJobRunner: Records R/W=9907/1 > sor
t
15/09/22 20:01:32 INFO streaming.PipeMapRed: R/W/S=200000/86913/0 in:5
0000=200000/4 [rec/s] out:21728=86913/4 [rec/s]
15/09/22 20:01:33 INFO mapred.LocalJobRunner: Records R/W=9907/1 > sor
t
15/09/22 20:01:34 INFO streaming.PipeMapRed: R/W/S=300000/127623/0 in:
50000=300000/6 [rec/s] out:21270=127623/6 [rec/s]
15/09/22 20:01:36 INFO streaming.PipeMapRed: R/W/S=400000/165901/0 in:
50000=400000/8 [rec/s] out:20737=165901/8 [rec/s]
15/09/22 20:01:36 INFO mapred.LocalJobRunner: Records R/W=9907/1 > sor
t
15/09/22 20:01:38 INFO streaming.PipeMapRed: Records R/W=481576/202398
15/09/22 20:01:38 INFO streaming.PipeMapRed: R/W/S=500000/209696/0 in:
50000=500000/10 [rec/s] out:20969=209696/10 [rec/s]
15/09/22 20:01:39 INFO mapred.LocalJobRunner: Records R/W=481576/20239
8 > sort
15/09/22 20:01:40 INFO streaming.PipeMapRed: R/W/S=600000/252888/0 in:
50000=600000/12 [rec/s] out:21074=252888/12 [rec/s]
15/09/22 20:01:42 INFO streaming.PipeMapRed: R/W/S=700000/297979/0 in:
50000=700000/14 [rec/s] out:21284=297979/14 [rec/s]
15/09/22 20:01:42 INFO mapred.LocalJobRunner: Records R/W=481576/20239
8 > sort
15/09/22 20:01:45 INFO streaming.PipeMapRed: R/W/S=800000/342503/0 in:
50000=800000/16 [rec/s] out:21406=342503/16 [rec/s]
15/09/22 20:01:45 INFO mapred.LocalJobRunner: Records R/W=481576/20239
8 > sort
15/09/22 20:01:47 INFO streaming.PipeMapRed: R/W/S=900000/380486/0 in:
50000=900000/18 [rec/s] out:21138=380486/18 [rec/s]
15/09/22 20:01:47 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 20:01:47 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 20:01:47 INFO mapred.MapTask: Finished spill 1
15/09/22 20:01:47 INFO mapred.Merger: Merging 2 sorted segments
15/09/22 20:01:47 INFO mapred.Merger: Down to the last merge-pass, wit
h 2 segments left of total size: 29812435 bytes
15/09/22 20:01:48 INFO mapred.Task: Task:attempt_local2065773600_0001_
m_000000_0 is done. And is in the process of committing
15/09/22 20:01:48 INFO mapred.LocalJobRunner: Records R/W=481576/20239
8 > sort
15/09/22 20:01:48 INFO mapred.Task: Task 'attempt_local2065773600_0001
_m_000000_0' done.
15/09/22 20:01:48 INFO mapred.LocalJobRunner: Finishing task: attempt_
local2065773600_0001_m_000000_0
15/09/22 20:01:48 INFO mapred.LocalJobRunner: map task executor comple
te.
15/09/22 20:01:48 INFO mapred.LocalJobRunner: Waiting for reduce tasks
15/09/22 20:01:48 INFO mapred.LocalJobRunner: Starting task: attempt_l
```

```
ocal2065773600_0001_r_000000_0
15/09/22 20:01:48 INFO output.FileOutputCommitter: File Output Committ
er Algorithm version is 1
15/09/22 20:01:48 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcess
Tree currently is supported only on Linux.
15/09/22 20:01:48 INFO mapred.Task:  Using ResourceCalculatorProcessTr
ee : null
15/09/22 20:01:48 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@1911483f
15/09/22 20:01:48 INFO reduce.MergeManagerImpl: MergerManager: memoryL
imit=334338464, maxSingleShuffleLimit=83584616, mergeThreshold=2206633
92, ioSortFactor=10, memToMemMergeOutputsThreshold=10
15/09/22 20:01:48 INFO reduce.EventFetcher: attempt_local2065773600_00
01_r_000000_0 Thread started: EventFetcher for fetching Map Completion
Events
15/09/22 20:01:48 INFO reduce.LocalFetcher: localfetcher#1 about to sh
uffle output of map attempt_local2065773600_0001_m_000000_0 decomp: 29
812465 len: 29812469 to MEMORY
15/09/22 20:01:48 INFO reduce.InMemoryMapOutput: Read 29812465 bytes f
rom map-output for attempt_local2065773600_0001_m_000000_0
15/09/22 20:01:48 INFO reduce.MergeManagerImpl: closeInMemoryFile -> m
ap-output of size: 29812465, inMemoryMapOutputs.size() -> 1, commitMem
ory -> 0, usedMemory ->29812465
15/09/22 20:01:48 INFO reduce.EventFetcher: EventFetcher is interrupte
d.. Returning
15/09/22 20:01:48 INFO mapred.LocalJobRunner: 1 / 1 copied.
15/09/22 20:01:48 INFO reduce.MergeManagerImpl: finalMerge called with
1 in-memory map-outputs and 0 on-disk map-outputs
15/09/22 20:01:48 INFO mapred.Merger: Merging 1 sorted segments
15/09/22 20:01:48 INFO mapred.Merger: Down to the last merge-pass, wit
h 1 segments left of total size: 29812445 bytes
15/09/22 20:01:48 INFO mapreduce.Job:  map 100% reduce 0%
15/09/22 20:01:49 INFO reduce.MergeManagerImpl: Merged 1 segments, 298
12465 bytes to disk to satisfy reduce memory limit
15/09/22 20:01:49 INFO reduce.MergeManagerImpl: Merging 1 files, 29812
469 bytes from disk
15/09/22 20:01:49 INFO reduce.MergeManagerImpl: Merging 0 segments, 0
bytes from memory into reduce
15/09/22 20:01:49 INFO mapred.Merger: Merging 1 sorted segments
15/09/22 20:01:49 INFO mapred.Merger: Down to the last merge-pass, wit
h 1 segments left of total size: 29812445 bytes
15/09/22 20:01:49 INFO mapred.LocalJobRunner: 1 / 1 copied.
15/09/22 20:01:49 INFO streaming.PipeMapRed: PipeMapRed exec [/Users/g
tumuluri/Documents/BerkeleyMIDS/ScalableML/HW3/./3.2_pairs_reducer.py]
15/09/22 20:01:49 INFO Configuration.deprecation: mapred.job.tracker i
s deprecated. Instead, use mapreduce.jobtracker.address
15/09/22 20:01:49 INFO Configuration.deprecation: mapred.map.tasks is
deprecated. Instead, use mapreduce.job.maps
15/09/22 20:01:49 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s]
out:NA [rec/s]
```

```
15/09/22 20:01:49 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s
] out:NA [rec/s]
15/09/22 20:01:49 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/
s] out:NA [rec/s]
15/09/22 20:01:49 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec
/s] out:NA [rec/s]
15/09/22 20:01:49 INFO streaming.PipeMapRed: R/W/S=10000/0/0 in:NA [re
c/s] out:NA [rec/s]
15/09/22 20:01:51 INFO streaming.PipeMapRed: R/W/S=100000/0/0 in:50000
=100000/2 [rec/s] out:0=0/2 [rec/s]
15/09/22 20:01:53 INFO streaming.PipeMapRed: R/W/S=200000/0/0 in:50000
=200000/4 [rec/s] out:0=0/4 [rec/s]
15/09/22 20:01:54 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:01:54 INFO mapreduce.Job:  map 100% reduce 75%
15/09/22 20:01:55 INFO streaming.PipeMapRed: R/W/S=300000/0/0 in:50000
=300000/6 [rec/s] out:0=0/6 [rec/s]
15/09/22 20:01:57 INFO streaming.PipeMapRed: R/W/S=400000/0/0 in:50000
=400000/8 [rec/s] out:0=0/8 [rec/s]
15/09/22 20:01:57 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:01:57 INFO mapreduce.Job:  map 100% reduce 80%
15/09/22 20:01:59 INFO streaming.PipeMapRed: R/W/S=500000/0/0 in:50000
=500000/10 [rec/s] out:0=0/10 [rec/s]
15/09/22 20:02:00 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:02:00 INFO mapreduce.Job:  map 100% reduce 85%
15/09/22 20:02:01 INFO streaming.PipeMapRed: R/W/S=600000/0/0 in:50000
=600000/12 [rec/s] out:0=0/12 [rec/s]
15/09/22 20:02:03 INFO streaming.PipeMapRed: R/W/S=700000/0/0 in:50000
=700000/14 [rec/s] out:0=0/14 [rec/s]
15/09/22 20:02:03 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:02:03 INFO mapreduce.Job:  map 100% reduce 89%
15/09/22 20:02:05 INFO streaming.PipeMapRed: R/W/S=800000/0/0 in:50000
=800000/16 [rec/s] out:0=0/16 [rec/s]
15/09/22 20:02:06 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:02:06 INFO mapreduce.Job:  map 100% reduce 94%
15/09/22 20:02:07 INFO streaming.PipeMapRed: R/W/S=900000/0/0 in:50000
=900000/18 [rec/s] out:0=0/18 [rec/s]
15/09/22 20:02:09 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:02:09 INFO streaming.PipeMapRed: R/W/S=1000000/0/0 in:5000
0=1000000/20 [rec/s] out:0=0/20 [rec/s]
15/09/22 20:02:09 INFO mapreduce.Job:  map 100% reduce 99%
15/09/22 20:02:11 INFO streaming.PipeMapRed: Records R/W=1030795/1
15/09/22 20:02:11 INFO streaming.PipeMapRed: MRErrorThread done
15/09/22 20:02:11 INFO streaming.PipeMapRed: mapRedFinished
15/09/22 20:02:11 INFO mapred.Task: Task:attempt_local2065773600_0001_
r_000000_0 is done. And is in the process of committing
15/09/22 20:02:11 INFO mapred.LocalJobRunner: reduce > reduce
15/09/22 20:02:11 INFO mapred.Task: Task attempt_local2065773600_0001_
r_000000_0 is allowed to commit now
15/09/22 20:02:11 INFO output.FileOutputCommitter: Saved output of tas
k 'attempt_local2065773600_0001_r_000000_0' to hdfs://localhost:9000/u
```

```
ser/gtumuluri/oneAndTwoItemsetsOutput/_temporary/0/task_local206577360
0_0001_r_000000
15/09/22 20:02:11 INFO mapred.LocalJobRunner: Records R/W=1030795/1 >
reduce
15/09/22 20:02:11 INFO mapred.Task: Task 'attempt_local2065773600_0001
_r_000000_0' done.
15/09/22 20:02:11 INFO mapred.LocalJobRunner: Finishing task: attempt_
local2065773600_0001_r_000000_0
15/09/22 20:02:11 INFO mapred.LocalJobRunner: reduce task executor com
plete.
15/09/22 20:02:11 INFO mapreduce.Job:  map 100% reduce 100%
15/09/22 20:02:11 INFO mapreduce.Job: Job job_local2065773600_0001 com
pleted successfully
15/09/22 20:02:11 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=119462040
                FILE: Number of bytes written=149863529
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=6917034
                HDFS: Number of bytes written=276
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=31101
                Map output records=2914884
                Map output bytes=76036100
                Map output materialized bytes=29812469
                Input split bytes=112
                Combine input records=2914884
                Combine output records=1030795
                Reduce input groups=889690
                Reduce shuffle bytes=29812469
                Reduce input records=1030795
                Reduce output records=9
                Spilled Records=3092385
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=8
                Total committed heap usage (bytes)=488636416
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
```

```
        File Input Format Counters
                Bytes Read=3458517
        File Output Format Counters
                Bytes Written=276
15/09/22 20:02:11 INFO streaming.StreamJob: Output directory: oneAndTw
oItemsetsOutput
```

In [24]:
```
!hdfs dfs -cat oneAndTwoItemsetsOutput/part-00000
```

```
15/09/22 20:02:18 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
Number of frequent 1-item sets: 647
Number of frequent 2-item sets: 1334

Top Five Rules with Confidence Scores:
('DAI93865', 'FRO40251', 1.0)
('DAI88079', 'FRO40251', 0.99)
('ELE12951', 'FRO40251', 0.99)
('DAI43868', 'SNA82528', 0.97)
('DAI23334', 'DAI62779', 0.95)
```

Problem 3.3 - pyFIM

This problem has been cancelled, and so it is not presented here.

Problem 3.4 - Apriori Algorithm with MapReduce to Find High Confidence 3-Itemsets

From the key foundational principle of Apriori algorithm on frequent item subsets, we know that all 3-itemsets will only be made up of items in the 2-itemsets, and further more, a 2-itemset has to be in the 3-itemset. So, we would have the mapper take the list of 2-itemsets, and generate the combinations of possible 3-itemsets.

For example, we (a, b), (c, d) and (e, f) were the 2-itemsets, then the possible 3-itemsets are:

(a, b) (c) (a, b) (d) (a, b) (e) (a, b) (f) (c, d) (a) and so forth.....

After generating the combinations, we can send them to another set of mappers in a separate file. The mappers will then use this information and re-process the original basket data to compare against possible 3-itemset candidates. When a matching 3-itemset is found, the mapper will output the occurrence and count (of 1). A combiner can be used again to aggregate counts for each 3-itemset occurrence. The reducer will take each candidate 3-itemset found, and create aggregate counts and do support/confidence computations. Possible combinations of 3-itemset candidates can be stored in a dict, which can be looked up as the mapper re-processes the basket by looking up the existence of the combination. Mapper will have a 'triple' loop for each basket to generate 3-itemset candidates, which will then be looked up in the dict that has the 'subset' of candidates that have met the frequent 2-itemset and 1-itemset criteria in the original map/reduce step.

Stop Yarn and HDFS

In [25]:
```
!/usr/local/Cellar/hadoop/2.7.0/sbin/stop-yarn.sh
!/usr/local/Cellar/hadoop/2.7.0/sbin/stop-dfs.sh
```

```
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
no proxyserver to stop
15/09/22 20:23:25 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
15/09/22 20:23:44 WARN util.NativeCodeLoader: Unable to load native-ha
doop library for your platform... using builtin-java classes where app
licable
```

In [ ]:
```

```