**Association Rule Mining with FP-Growth¶**

We will be identifying frequent itemsets using the FP-Growth Algorithm in Pyspark.

First, we must import the appropriate libraries from Pyspark machine learning algorithms for frequent pattern matching.

According to the [Frequent Pattern Matching API documentation](https://spark.apache.org/docs/2.2.0/mllib-frequent-pattern-mining.html), "The FP-growth algorithm is described in the paper Han et al., Mining frequent patterns without candidate generation, where “FP” stands for frequent pattern. Given a dataset of transactions, the first step of FP-growth is to calculate item frequencies and identify frequent items."

According to [Spark documentation](https://spark.apache.org/docs/2.0.2/api/java/org/apache/spark/SparkContext.html), "A SparkContext represents the connection to a Spark cluster, and can be used to create RDDs, accumulators and broadcast variables on that cluster. **Only one SparkContext may be active per JVM.** You must stop() the active SparkContext before creating a new one. This limitation may eventually be removed; see SPARK-2243 for more details."

In [1]:

**from** **pyspark.mllib.fpm** **import** FPGrowth

**from** **pyspark** **import** SparkContext

sc = SparkContext()

Next, we first read in a text file (sample\_fpgrowth.txt) with the transactiond data we will be using for association rule mining. Every line in the dataset corresponds to a transaction or basket and items are seperated by a space:

r z h k p z y x w v u t s s x o n r x z y m t s q e z x z y r q t p

The first transaction, for example, contains the items r, z, h, k, and p.

Next, we map data into an appropriate data strcuture and assigning this to a variable called transactions. .strip() returns a copy of the string in which all chars have been stripped from the beginning and the end of the string (default whitespace characters). .split(' ') returns a list of all the words in the string, using ' ' as the separator.

In [2]:

data = sc.textFile("sample\_fpgrowth.txt")

transactions = data.map(**lambda** line: line.strip().split(' '))

Next, we use the transactions data to train an FP Growth model to identify frequent itemsets and association rules.

According to the [FP-growth documentation](https://spark.apache.org/docs/2.2.0/mllib-frequent-pattern-mining.html): spark.mllib’s FP-growth implementation takes the following (hyper-)parameters:

* minSupport: the minimum support for an itemset to be identified as frequent. For example, if an item appears 3 out of 5 transactions, it has a support of 3/5=0.6.
* numPartitions: the number of partitions used to distribute the work.

After training the model, we extract freuqent itemsets and assign them to a dictionary where we can more easily keep track of their frequency. This will be helpful for determining association rules and relevant support and confidence metrics.

In [3]:

model = FPGrowth.train(transactions, minSupport = 0.2, numPartitions = 10)

freqItemsets = model.freqItemsets().collect()

freqItemsetsDict = {}

**for** i **in** range(0, len(freqItemsets)):

freqItemsetsDict[str(freqItemsets[i].items)] = freqItemsets[i].freq

Finally, we set a minimum confidence, generate association rules from the frequent itemsets identified above, and print rules with corresponding evaluation metrics.

In [4]:

minConfidence = .6

rules = sorted(model.\_java\_model.generateAssociationRules(minConfidence).collect(),

key=**lambda** x: x.confidence(), reverse=**True**)

**for** j **in** range(0, len(rules)):

antecedent = list(rules[j].antecedent())

consequent = list(rules[j].consequent())

ruleItemset = str(antecedent + consequent)

**try**:

support = "**{0:.2f}**".format(freqItemsetsDict[ruleItemset] / transactions.count())

**except** **KeyError**:

next

confidence = "**{0:.2f}**".format(rules[j].confidence())

**try**:

print(antecedent, '=>', consequent, '(support: '+ str(support), 'confidence: '+ str(confidence) + ')')

**except** **NameError**:

next

['t', 's', 'y'] => ['x'] (support: 0.33 confidence: 1.00)

['t', 's', 'y'] => ['z'] (support: 0.33 confidence: 1.00)

['y', 'x', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['y'] => ['x'] (support: 0.50 confidence: 1.00)

['y'] => ['z'] (support: 0.50 confidence: 1.00)

['y'] => ['t'] (support: 0.50 confidence: 1.00)

['p'] => ['r'] (support: 0.33 confidence: 1.00)

['p'] => ['z'] (support: 0.33 confidence: 1.00)

['q', 't', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['q', 't', 'z'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 'y'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 'y'] => ['z'] (support: 0.33 confidence: 1.00)

['q', 'y'] => ['t'] (support: 0.33 confidence: 1.00)

['t', 's', 'x'] => ['y'] (support: 0.33 confidence: 1.00)

['t', 's', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['q', 't', 'y', 'z'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 't', 'x', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['q', 'x'] => ['y'] (support: 0.33 confidence: 1.00)

['q', 'x'] => ['t'] (support: 0.33 confidence: 1.00)

['q', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 'x', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['x', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['x', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['p', 'z'] => ['r'] (support: 0.33 confidence: 1.00)

['t'] => ['y'] (support: 0.50 confidence: 1.00)

['t'] => ['x'] (support: 0.50 confidence: 1.00)

['t'] => ['z'] (support: 0.50 confidence: 1.00)

['y', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['y', 'z'] => ['t'] (support: 0.50 confidence: 1.00)

['p', 'r'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 's'] => ['y'] (support: 0.33 confidence: 1.00)

['t', 's'] => ['x'] (support: 0.33 confidence: 1.00)

['t', 's'] => ['z'] (support: 0.33 confidence: 1.00)

['q', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['q', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['q', 'z'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 'y', 'z'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 'y', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['y', 'x'] => ['z'] (support: 0.50 confidence: 1.00)

['y', 'x'] => ['t'] (support: 0.50 confidence: 1.00)

['q', 'x', 'z'] => ['y'] (support: 0.50 confidence: 1.00)

['q', 'x', 'z'] => ['t'] (support: 0.50 confidence: 1.00)

['t', 'y', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['q', 'y', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['q', 'y', 'x'] => ['t'] (support: 0.33 confidence: 1.00)

['q', 't', 'y', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 's', 'x', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['s', 'y', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['s', 'y', 'x'] => ['t'] (support: 0.33 confidence: 1.00)

['s', 'x', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['s', 'x', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['q', 'y', 'x', 'z'] => ['t'] (support: 0.33 confidence: 1.00)

['s', 'y'] => ['x'] (support: 0.33 confidence: 1.00)

['s', 'y'] => ['z'] (support: 0.33 confidence: 1.00)

['s', 'y'] => ['t'] (support: 0.33 confidence: 1.00)

['q', 't', 'y'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 't', 'y'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 'y'] => ['x'] (support: 0.50 confidence: 1.00)

['t', 'y'] => ['z'] (support: 0.50 confidence: 1.00)

['t', 'z'] => ['y'] (support: 0.50 confidence: 1.00)

['t', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['t', 's', 'y', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 'y', 'x'] => ['z'] (support: 0.50 confidence: 1.00)

['q', 't'] => ['y'] (support: 0.33 confidence: 1.00)

['q', 't'] => ['x'] (support: 0.33 confidence: 1.00)

['q', 't'] => ['z'] (support: 0.33 confidence: 1.00)

['q'] => ['y'] (support: 0.33 confidence: 1.00)

['q'] => ['t'] (support: 0.33 confidence: 1.00)

['q'] => ['x'] (support: 0.33 confidence: 1.00)

['q'] => ['z'] (support: 0.33 confidence: 1.00)

['t', 's', 'z'] => ['y'] (support: 0.33 confidence: 1.00)

['t', 's', 'z'] => ['x'] (support: 0.33 confidence: 1.00)

['t', 'x'] => ['y'] (support: 0.33 confidence: 1.00)

['t', 'x'] => ['z'] (support: 0.50 confidence: 1.00)

['s', 'z'] => ['y'] (support: 0.50 confidence: 1.00)

['s', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['s', 'z'] => ['t'] (support: 0.50 confidence: 1.00)

['s', 'y', 'x', 'z'] => ['t'] (support: 0.50 confidence: 1.00)

['s'] => ['x'] (support: 0.50 confidence: 1.00)

['t', 's', 'y', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['s', 'y', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

['s', 'y', 'z'] => ['t'] (support: 0.50 confidence: 1.00)

['q', 't', 'x'] => ['y'] (support: 0.50 confidence: 1.00)

['q', 't', 'x'] => ['z'] (support: 0.33 confidence: 1.00)

['r', 'z'] => ['p'] (support: 0.33 confidence: 1.00)

['x'] => ['z'] (support: 0.50 confidence: 0.75)

['x'] => ['y'] (support: 0.50 confidence: 0.75)

['x'] => ['s'] (support: 0.50 confidence: 0.75)

['x'] => ['t'] (support: 0.50 confidence: 0.75)

['y', 'x', 'z'] => ['s'] (support: 0.50 confidence: 0.67)

['y', 'x', 'z'] => ['q'] (support: 0.50 confidence: 0.67)

['y'] => ['s'] (support: 0.50 confidence: 0.67)

['y'] => ['q'] (support: 0.50 confidence: 0.67)

['t', 'x', 'z'] => ['s'] (support: 0.50 confidence: 0.67)

['t', 'x', 'z'] => ['q'] (support: 0.50 confidence: 0.67)

['x', 'z'] => ['s'] (support: 0.50 confidence: 0.67)

['x', 'z'] => ['q'] (support: 0.50 confidence: 0.67)

['t'] => ['s'] (support: 0.33 confidence: 0.67)

['t'] => ['q'] (support: 0.33 confidence: 0.67)

['y', 'z'] => ['s'] (support: 0.33 confidence: 0.67)

['y', 'z'] => ['q'] (support: 0.33 confidence: 0.67)

['y', 'x'] => ['s'] (support: 0.33 confidence: 0.67)

['y', 'x'] => ['q'] (support: 0.33 confidence: 0.67)

['t', 'y', 'z'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'y', 'z'] => ['q'] (support: 0.33 confidence: 0.67)

['t', 'y'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'y'] => ['q'] (support: 0.33 confidence: 0.67)

['t', 'z'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'z'] => ['q'] (support: 0.33 confidence: 0.67)

['r'] => ['x'] (support: 0.33 confidence: 0.67)

['r'] => ['z'] (support: 0.33 confidence: 0.67)

['r'] => ['p'] (support: 0.33 confidence: 0.67)

['t', 'y', 'x'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'y', 'x'] => ['q'] (support: 0.33 confidence: 0.67)

['t', 'x'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'x'] => ['q'] (support: 0.33 confidence: 0.67)

['t', 'y', 'x', 'z'] => ['s'] (support: 0.33 confidence: 0.67)

['t', 'y', 'x', 'z'] => ['q'] (support: 0.33 confidence: 0.67)

['s'] => ['y'] (support: 0.33 confidence: 0.67)

['s'] => ['z'] (support: 0.33 confidence: 0.67)

['s'] => ['t'] (support: 0.33 confidence: 0.67)

['s', 'x'] => ['y'] (support: 0.33 confidence: 0.67)

['s', 'x'] => ['z'] (support: 0.33 confidence: 0.67)

['s', 'x'] => ['t'] (support: 0.33 confidence: 0.67)

['z'] => ['x'] (support: 0.33 confidence: 0.60)

['z'] => ['y'] (support: 0.33 confidence: 0.60)

['z'] => ['t'] (support: 0.33 confidence: 0.60)

We can interpret these association rules as follows:

['t', 'y', 'z'] => ['x'] (support: 0.50 confidence: 1.00)

* A support of .50 suggests that products t, y, z, and x appear together in 50% of the transactions.
* A confidence of 1 suggests that every transaction that contains products t, y, and z also contains product x.

['z'] => ['x'] (support: 0.33 confidence: 0.60)

* A support of .33 suggests that products z and x appear together in 33% of the transactions.
* A confidence of .6 suggests that every 100 transactions that contains product z, 60 also contains product x.