Itemset and Association rule mining

Itemset and Association rule mining

- Spark MLlib provides
  - An itemset mining algorithm based on the FP-growth algorithm
    - That extracts all the sets of items (of any length) with a minimum frequency
  - A rule mining algorithm
    - That extracts the association rules with a minimum frequency and a minimum confidence
    - Only the rules with one single item in the consequent of the rules are extracted

Itemset and Association rule mining

- The input dataset in this case is a set of transactions
- Each transaction is defined as a set of items
- Transactional dataset example

  A B C D

  A B

  B C

  A D E
- The example dataset contains 4 transactions
- The distinct items are A, B, C, D, E

The FP-Growth algorithm and Association rule mining

The FP-Growth algorithm

- FP-growth is one of the most popular and efficient itemset mining algorithms
- It is characterized by one single parameter
  - The minimum support threshold **(minsup)**
    - i.e., the minimum frequency of the itemset in the input transational dataset
    - It is a real value in the range (0-1]
  - The minsup threshold is used to limit the number of mined itemsets
- The input dataset is a transactional dataset

Association Rule Mining

- Given a set of frequent itemsets, the frequent association rules can be mined
- An association rule is mined if
  - Its frequency is greater than the minimum support threshold (**minsup**)
    - i.e., a minimum frequency
    - The minsup value is specified during the itemset mining step and not during the association rule mining step
  - Its confidence is greater than the minimum confidence threshold (**minconf**)
    - i.e., a minimum “correlation”
    - It is a real value in the range [0-1]

The FP-Growth algorithm

- The MLlib implementation of FP-growth is based on DataFrames
- Differently from the other algorithms, the FP-growth algorithm is not invoked by using pipelines

Itemset and Association Rule Mining

- Itemset and association rule mining
  - Instantiate an FP-Growth object
  - Invoke the fit(input data) method on the FP-Growth object
  - Retrieve the sets of frequent itemset and association rules by invoking the following methods of on the FP-Growth object
    - freqItemsets()
    - associationRules()

Itemset and Association Rule Mining: Input data

- The input of the MLlib itemset and rule mining algorithm is a DataFrame containing a column called **items**
  - Data type: array of values
- Each record of the input DataFrame contains one transaction, i.e., a set of items

Itemset and Association Rule Mining: Input data

- Example of input data

  transactions

  A B C D

  A B

  B C

  A D E

Itemset and Association Rule Mining: Input data

- Input data

transactions
A B C D
A B
B C
A D E

- The column items must be created before invoking FP-growth

items
[A, B, C, D]
[A, B]
[B, C]
[A, D, E]

Itemset and Association Rule Mining: Input data

Each input line is "stored" in an array of strings
The generated DataFrame contains a column called items, which is an ArrayType, containing the lists of items associated with the input transactions.

- Input data

transactions
A B C D
A B
B C
A D E

- The column items must be created before invoking FP-growth

items
[A, B, C, D]
[A, B]
[B, C]
[A, D, E]

Itemset and Association Rule Mining: Example

- The following slides show how to
  - Extract the set of frequent itemsets from a transactional dataset and the association rules from the extracted frequent itemsets
- The input dataset is a transactional dataset
  - Each line of the input file contains a transaction, i.e., a set of items

Itemset and Association Rule Mining: Example

- Example of input data

  transactions

  A B C D

  A B

  B C

  A D E

Itemset and Association Rule Mining: Example

```
from pyspark.ml.fpm import FPGrowth
from pyspark.ml import Pipeline
from pyspark.ml import PipelineModel
from pyspark.sql.functions import col, split

# input and output folders
transactionsData = "ex_dataitemsets/transactions.csv"
outputPathItemsets = "Itemsets/"
outputPathRules = "Rules/“

# Create a DataFrame from transactions.csv
transactionsDataDF = spark.read.load(transactionsData,\
            format="csv", header=True,\
            inferSchema=True)
```

Itemset and Association Rule Mining: Example

```
# Transform Column transactions into an ArrayType
trsDataDF = transactionsDataDF\
.selectExpr('split(transactions, " ")')\
.withColumnRenamed("split(transactions,  )", "items")
```

Itemset and Association Rule Mining: Example

```
# Transform Column transactions into an ArrayType
trsDataDF = transactionsDataDF\
.selectExpr('split(transactions, " ")')\
.withColumnRenamed("split(transactions,  )", "items")
```

This is the pyspark.sql.functions.split() function.
It returns an SQL ArrayType

Itemset and Association Rule Mining: Example

```
# Transform Column transactions into an ArrayType
trsDataDF = transactionsDataDF\
.selectExpr('split(transactions, " ")')\
.withColumnRenamed("split(transactions,  )", "items")

# Create an FP-growth Estimator
fpGrowth = FPGrowth(itemsCol="items", minSupport=0.5, minConfidence=0.6)

# Extract itemsets and rules
model = fpGrowth.fit(trsDataDF)

# Retrieve the DataFrame associated with the frequent itemsets
dfItemsets = model.freqItemsets

# Retrieve the DataFrame associated with the frequent rules
dfRules = model.associationRules
```

Itemset and Association Rule Mining: Example

```
# Save the result in an HDFS output folder
dfItemsets.write.json(outputPathItemsets)

# Save the result in an HDFS output folder
dfRules.write.json(outputPathRules)
```

The result is stored in a JSON file because itemsets and rules are stored in columns associated with the data type Array.
Hence, CSV files cannot be used to store the result.