Big data: architectures and data analytics

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 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]

7

Itemset and Association Rule Mining: example

- The following slides show how to
  - Extract the set of frequent itemsets from a transactional dataset
  - Extract the association rules from the frequent itemsets
- The input dataset is a transactional dataset

8

Itemset and Association Rule Mining: example

- Example of input file

  A B C D

  A B

  B C

  A D E

9

Itemset and Association Rule Mining: example

```
package it.polito.bigdata.spark.sparkmllib;

import java.util.ArrayList;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.mllib.fpm.AssociationRules;
import org.apache.spark.mllib.fpm.FPGrowth;
import org.apache.spark.mllib.fpm.FPGrowth.FreqItemset;
import org.apache.spark.mllib.fpm.FPGrowthModel;
```

10

Itemset and Association Rule Mining: example

```
public class SparkDriver {

  public static void main(String[] args) {
    String inputFile;
    String outputFolderItemsets;
    String outputFolderRules;
    double minSupport;
    double minConfidence;

    inputFile = args[0];
    outputFolderItemsets = args[1];
    outputFolderRules = args[2];
    minSupport = Double.parseDouble(args[3]);
    minConfidence = Double.parseDouble(args[4]);
```

11

Itemset and Association Rule Mining: example

```
// Create a configuration object and set the name of the application
SparkConf conf=new SparkConf().setAppName("Itemset and Association rule
  mining");

// Create a Spark Context object
JavaSparkContext sc = new JavaSparkContext(conf);

JavaRDD<String> inputRDD = sc.textFile(inputFile);

// Read the input data
// Each line of the input file contains one transaction,
// i.e., a list of items
// Each element of the JavaRDD represents one transaction = a list of strings
JavaRDD<ArrayList<String>>transactions =inputRDD.map(new InputRecord());
```

12

Itemset and Association Rule Mining: example

```
// Create an FPGrowth object
FPGrowth fp=new FPGrowth();

// Set the minimum support (frequency threshold)
fp.setMinSupport(minSupport);

// Run the algorithm
FPGrowthModel<String> model = fp.run(transactions);

// Retrieve the extracted itemsets
JavaRDD<FreqItemset<String>>freqItemsets=model.freqItemsets().toJavaRDD();
```

13

Itemset and Association Rule Mining: example

```
// FreqItemset<String> is a complex data type. It contains a list of items
// and the frequency of that set
// The following map is used to map each FreqItemset object to a String
// containing the set of items and the support using the format
  // [list of items], frequency=frequency
JavaRDD<String> freqItemsetsString=freqItemsets.map(
                                        new ItemsetAsSingleString());

freqItemsetsString.saveAsTextFile(outputFolderItemsets);
```

14

Itemset and Association Rule Mining: example

```
// Association rule mining from the set of frequent itemsets mined
// by using FPGrowth
// Instance an AssociationRules object
AssociationRules arules = new AssociationRules();

// Set the minimum confidence threshold
arules.setMinConfidence(minConfidence);

// Extract the rules
JavaRDD<AssociationRules.Rule<String>> assRules = arules.run(freqItemsets);
```

15

Itemset and Association Rule Mining: example

```
package it.polito.bigdata.spark.sparkmllib;
import java.util.ArrayList;
import org.apache.spark.api.java.function.Function;

@SuppressWarnings("serial")
public class InputRecord implements Function<String, ArrayList<String>> {

    public ArrayList<String> call(String line) {
        ArrayList<String> itemsList=new ArrayList<String>();
        String[] items = line.split(" ");

        for (String item: items) {
                itemsList.add(item);
        }
        return itemsList;
    }
}
```

16

Itemset and Association Rule Mining: example

```
package it.polito.bigdata.spark.sparkmllib;

import org.apache.spark.api.java.function.Function;
import org.apache.spark.mllib.fpm.FPGrowth.FreqItemset;

@SuppressWarnings("serial")
public class ItemsetAsSingleString implements Function<FreqItemset<String>,
    String> {

    @Override
    public String call(FreqItemset<String> itemset) {
        return new String("["+itemset.javaItems()+"], frequency="+itemset.freq());
    }

}
```

17

Itemset and Association Rule Mining: example

```
package it.polito.bigdata.spark.sparkmllib;

import org.apache.spark.api.java.function.Function;
import org.apache.spark.mllib.fpm.AssociationRules.Rule;

@SuppressWarnings("serial")
public class AssRuleToString implements Function<Rule<String>, String> {

    public String call(Rule<String> rule) throws Exception {

        return new String("{"+rule.javaAntecedent()+"} -> {"+
                          rule.javaConsequent()+"} "+
                          "confidence="+rule.confidence());
    }
}
```

18