Big Data Analytics LAB1

Nicolas Taba (nicta839) & Tim Yuki Washio (timwa902)

5/5/2021

Assignment 1

```
from pyspark import SparkContext

sc = SparkContext(appName = "BDA1_1")

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")

# Map the data, split by lines
lines = temperature_file.map(lambda line: line.split(";"))

# Year and temp (think about adding stations here as well?)
year_temperature = lines.map(lambda x: (x[1][0:4], float(x[3])))

# Filter year
year_temperature = year_temperature.filter(lambda x: int(x[0])>=1950 and int(x[0])<=2014)

# Max temp
max_temperature = year_temperature.reduceByKey(lambda a, b: a if a >= b else b)
max_temperature_sorted = max_temperature.sortBy(ascending=False, keyfunc=lambda k: k[1])

# Min temp
min_temperature = year_temperature.reduceByKey(lambda a, b: a if a < b else b)
min_temperature_sorted = min_temperature.sortBy(ascending=False, keyfunc=lambda k: k[1])

# save the data
max_temperature_sorted.saveAsTextFile("BDA/output/LAB1_MAX")
min_temperature_sorted.saveAsTextFile("BDA/output/LAB1_MIN")
```

Output

```
# Maximum temperatures
(u'1975', 36.1)
(u'1992', 35.4)
(u'1994', 34.7)
(u'2014', 34.4)
(u'2010', 34.4)
(u'1989', 33.9)
(u'1982', 33.8)
(u'1968', 33.7)
(u'1966', 33.5)
(u'2002', 33.3)
```

```
(u'1983', 33.3)

# Minimum temperatures
(u'1966', -49.4)
(u'1999', -49.0)
(u'1978', -47.7)
(u'1987', -47.3)
(u'1967', -45.4)
(u'1980', -45.0)
(u'1956', -45.0)
(u'1971', -44.3)
(u'1986', -44.2)
(u'2001', -44.0)
```

Assignment 2

```
from pyspark import SparkContext

sc = SparkContext(appName = "BDA1_2")

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")

# Map the data, split by lines
lines = temperature_file.map(lambda line: line.split(";"))

# (Year, month) key, (station, temp) value
yearmonth_temp = lines.map(lambda x: ((x[1][0:4], x[1][5:7]), (x[0], float(x[3]))))

# above 10 degrees filter
yearmonth_temp = yearmonth_temp.filter(lambda x: int(x[0][0]) >= 1950 and int(x[0][0]) <= 2014 and x[1]

# count stations
count_above10 = yearmonth_temp.map(lambda x: (x[0], 1))
count_above10 = count_above10.reduceByKey(lambda a, b: a+b)

# Save data
count_above10.saveAsTextFile("BDA/output/LAB1_2")

### Uniques
count_unique = yearmonth_temp.map(lambda x: (x[0], (x[1][0], 1))).distinct()

count_unique = count_unique.map(lambda x: (x[0], 1))
count_unique = count_unique.reduceByKey(lambda a, b: a+b)

# Save data
count_unique.saveAsTextFile("BDA/output/LAB1_2_UNIQUE")
```

Output

```
# Temperature counts above 10 degrees between 1950 and 2014
((u'2008', u'11'), 1494)
((u'1971', u'06'), 45789)
((u'1989', u'12'), 23)
```

```
((u'1966', u'11'), 169)
((u'1956', u'05'), 12050)
((u'1998', u'07'), 120230)
((u'1975', u'02'), 21)
((u'1983', u'10'), 11157)
((u'1992', u'05'), 33745)
((u'1969', u'10'), 12604)

# Distinct temperature counts above 10 degrees between 1950 and 2014
((u'2008', u'11'), 106)
((u'1992', u'05'), 311)
((u'1971', u'06'), 374)
((u'1989', u'12'), 8)
((u'1966', u'11'), 70)
((u'1956', u'05'), 125)
((u'1998', u'07'), 326)
((u'1975', u'02'), 14)
((u'1983', u'10'), 246)
((u'1978', u'09'), 358)
```

Assignment 3

```
from pyspark import SparkContext

sc = SparkContext(appName = "BDA1_3")

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")

# Map the data, split by lines
lines = temperature_file.map(lambda line: line.split(";"))

# (Year, month, station) key, (temp) value
daily_station_temp = lines.map(lambda x: ((x[1][0:4], x[1][5:7], x[1][8:10], x[0]), (float(x[3]))))

# Filter time period 1960-2014
daily_station_temp = daily_station_temp.filter(lambda x: int(x[0][0]) >= 1960 and int(x[0][0]) <= 2014)

# Get Max and Min temperature for each day
max_temperature = daily_station_temp.reduceByKey(lambda a, b: a if a >= b else b)
min_temperature = daily_station_temp.reduceByKey(lambda a, b: a if a < b else b)

# Join Max and Min temperatures
joined_rdd = max_temperature.join(min_temperature)

# Map to ((year, month, station_nb), daily_avg_temperature)
yearmonthstation_avgtemp = joined_rdd.map(lambda x: ((x[0][0], x[0][1], x[0][3]), (x[1][0] + x[1][1]) /

# Get the average monthly temperature
count_values_rdd = yearmonthstation_avgtemp.mapValues(lambda v: (v, 1))
reduced_rdd = count_values_rdd.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))
avg_by_key_rdd = reduced_rdd.mapValues(lambda v: v[0]/v[1])

avg_by_key_rdd.saveAsTextFile("BDA/output/LAB1_3")
```

Output

```
# Average monthly temperature for each station between 1960 and 2014
((u'1989', u'06', u'92400'), 14.686666666666667)
((u'1982', u'09', u'107530'), 11.171666666666669)
((u'2002', u'11', u'136360'), -5.861666666666667)
((u'1967', u'08', u'98170'), 15.408064516129032)
((u'2002', u'08', u'181900'), 15.598387096774191)
((u'1996', u'08', u'96190'), 17.100000000000005)
((u'1994', u'06', u'71180'), 13.036666666666665)
((u'1995', u'06', u'62400'), 16.001666666666665)
((u'1972', u'10', u'64130'), 7.666129032258066)
((u'1985', u'02', u'81130'), -7.678571428571429)
```

Assignment 4

```
from pyspark import SparkContext

sc = SparkContext(appName = "BDA1_4")

# Temperature and precipitation
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")

# Map the data, split by lines
lines_temp = temperature_file.map(lambda line: line.split(";"))
lines_precip = precipitation_file.map(lambda line: line.split(";"))

# Map to (station_nb, temperature_value)
station_temp = lines_temp.map(lambda x: (x[0], float(x[3])))

# Get maximum temperature per station
max_temperature = station_temp.reduceByKey(lambda a, b: a if a >= b else b)

# Filter all entries outside of 25-30 degrees
station_temp = station_temp.filter(lambda x: x[1] > 25 and x[1] < 30)

station_precip = lines_precip.map(lambda x: ((x[0], x[1][0:4], x[1][5:7], x[1][8:10]), float(x[3])))

# Calculate the sum of precipitation per day
station_precip = station_precip.reduceByKey(lambda a, b: a + b)

# Map to (station_nb, precipitation_value)
station_precip = station_precip.map(lambda x: (x[0][0], x[1]))

# Get maximum precipitation per station
max_precipitation = station_precip.reduceByKey(lambda a, b: a if a >= b else b)

# Filter all entries outside of 100-200mm
station_precip = station_precip.filter(lambda x: x[1] > 100 and x[1] < 200)

# Join
joined_rdd = max_temperature.join(max_precipitation)
```

```
joined_rdd.saveAsTextFile("BDA/output/LAB1_4")
```

Output

```
# Maximum measured temperature and precipitation for each station where
# 25 degrees <= temperature <= 30 degrees and
# 100mm <= precipitation <= 200mm

# 0 stations found!
```

Assignment 5

```
from pyspark import SparkContext

sc = SparkContext(appName = "BDA1_5")

# Precipitation and Ostergotland
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
oster_station = sc.textFile("BDA/input/stations-Ostergotland.csv")

# Map the data, split by lines
lines_precip = precipitation_file.map(lambda line: line.split(";"))
lines_oster = oster_station.map(lambda line: line.split(";"))

# Create tuple ((year, month, station_num), precipitation)
yearmonthstation_precip = lines_precip.map(lambda x: ((x[1][0:4], x[1][5:7], x[0]), float(x[3])))
yearmonthstation_precip_filtered = yearmonthstation_precip.filter(lambda x: int(x[0][0]) >= 1993 and in

# Collecting RDD and broadcasting the local python object to all nodes
station_ids_oster = lines_oster.map(lambda x: x[0])
stations_ids = sc.broadcast(station_ids_oster.collect()).value

# Keep only stations within Ostergotland
filtered_rdd = yearmonthstation_precip_filtered.filter(lambda x: x[0][2] in stations_ids)

# Sum all values by year, month and station
filtered_rdd = filtered_rdd.reduceByKey(lambda a, b: a + b)

# Map to ((year, month), (precipation_value, 1))
count_values_rdd = filtered_rdd.map(lambda x: ((x[0][0], x[0][1]), (x[1], 1)))

# Reduce to ((year, month), (precipitation_value_sum, count))
reduced_rdd = count_values_rdd.reduceByKey(lambda a, b: (a[0]+b[0], a[1]+b[1]))

# Map to ((year, month), average_precipitation)
avg_by_key_rdd = reduced_rdd.mapValues(lambda v: v[0]/v[1]).sortByKey()

avg_by_key_rdd.saveAsTextFile("BDA/output/LAB1_5")
```

Output

```
# Average monthly precipitation for each month, year and station in Ostergotland

((u'1993', u'04'), 0.0)
((u'1993', u'05'), 21.100000000000005)
((u'1993', u'06'), 56.5)
((u'1993', u'07'), 95.39999999999999)
((u'1993', u'08'), 80.70000000000005)
((u'1993', u'09'), 40.6)
((u'1993', u'10'), 43.2)
((u'1993', u'11'), 42.80000000000003)
((u'1993', u'12'), 37.10000000000003)
((u'1994', u'01'), 22.099999999999994)
```