Mining massive Datasets WS 2017/18

Problem Set 3

Rudolf Chrispens, Marvin, Daniela Schacherer

November 12, 2017

Exercise 02

1. MapReduce Pseudocode for k-means
   The k-means algorithm consists of two steps. The first step computes for each mean μ_i the set of points that are closest to it. In the second step new means are computed using the priorily determined sets. These two phases correspond to the Map- and the Reduce phase of MapReduce. The Map phase computes the squared distance to all means for each point x in the dataset and returns a key-value pair $(i, (x,1))$ where i is the index of the mean with the smallest distance to point x. The Reduce phase then simply computes the sum of the vector points for each key.
   Pseudocode:

   - **Map** for every point x: return $(argmin_i(||x - \mu_i||, (x,1))$
   - **Reduce** for every elements with key i: return $(i, (x + y, s + t))$ with x and y being the data points and s and t being the counts.

2. MapReduce Pseudocode for Inverted Indexing
   Inverted

   - **Map**: for every keyword in the given list the Mapper should perform the following:
     if $keyword$ in $text_i$: return $(keyword, doc_i)$
   - **Reduce**: for every keyword add the documents indices to a list and finally return $(keyword, [doc_i, doc_j, ..$

   The proposed pseudocode would probably not scale well with the number of keywords, because the mapper has to run a search algorithm for every keyword. If the number of mappers is limited, a mapper has to perform multiple searches and thus the time for the complete calculation will get larger.

3. MapReduce Pseudocode
   When one dataset is small and every mapper has access to it the joining can already be part of the mapping phase. Let R with tuples (a,b) and S with tuples (b,c) be the datasets, and R is the smaller one. We want to join on b. Every mapper gets tuples from S in this form: (S,a,b)

- **Map**: for every tuple of R:
  if b in (S,a,b): return (a,b,c)
- **Reduce**: in the reduce phase we now only have to collect all the joined tuples.

Exercise 03

a)

In Spark the input of every Transformation (map) and Action (reduce) is a RDD. With a Transformation the output is again a RDD and with an action the output can have different types. The data in the RDD can be different, key-value pairs is not mandatory.
In MapReduce the output is not a RDD but a set of key-value pairs. The output is again a set of key-value pairs. Depending on the map step the output has a different number of data. At the reduce step it is basically the same. Input and output are both key-value pairs.

b)

I wrote a spark application that counts the number of characters in a textfile. The Code is in the submitted *ex3problem3b.py* file that uses the textfile *lorem_ipsum.txt*.

Program code:

```
from pyspark import SparkContext

if __name__ == "__main__":
    sc = SparkContext(appName="PythonKMeans")

    lines = sc.textFile("lorem_ipsum.txt")
    lineLengths = lines.map(lambda s: len(s))
    totalLength = lineLengths.reduce(lambda a, b: a + b)

    print(totalLength)

    sc.stop()
```

The output of the Program

```
591
```