TECHNISCHE UNIVERSITÄT BERLIN
Database Systems and Information Management Group
Prof. V. Markl, C. Boden, Dr. S. Schelter, and J. Soto

AIM-3 Scalable Data Science
Summer Semester 2016

Assignment 1

Programming in Hadoop and Clustering Exercises
Due Date: June 10, 2016 at 12:00 noon

Programming in Hadoop (Total: 20 pts.)

1. **WordCount - "Hello World"of MapReduce (6 pts.)**

   We'll start with the classic MapReduce example of counting words. Your task is to complete the code in *de.tuberlin.dima.aim3.assignment1.FilteringWordCount*. The output of this job should be a textfile holding the following data per line:

   word[TAB]count.

   An additional requirement here is that stop words like *to*, *and*, *in* or *the* should be removed from the input data and all words should be lowercased.

2. **A custom Writable (6 pts.)**

   You will work on your first custom Writable object in this task. Have a look at the class *de.tuberlin.dima.aim3.assignment1.PrimeNumbersWritable*, which models a collection of prime numbers. Writable classes need to be able to serialize to and deserialize from a binary representation. Enable that for our custom Writable by implementing *write(DataOutput out)* and *readFields(DataInput in).*

3. **Average temperature per month (8 pts.)**

   Have a look at the file *src/test/resources/assignment1/temperatures.tsv.* It contains the output of a fictional temperature sensor, where each line denotes the year, the month and the temperature of a single recording. Additionally, a quality parameter is included which expresses how sure the sensor was of a single measurement:

   year[TAB]month[TAB]temperature[TAB]quality

   Your task is to implement a MapReduce program that computes the average temperature per month of year. It should ignore all records that are below a given minimum quality. The output of your program will be a textfile holding the following data per line:

   year[TAB]month[TAB]average temperature

   Use *de.tuberlin.dima.aim3.assignment1.AverageTemperaturePerMonth* as a starting point.

Clustering (Total: 20 pts.)

4. **Metrics I (3 pts.)**

   The SSE (sum squared error) is a common measure of the quality of a cluster. It is the sum of the squares of the distances between each of the points of the cluster and the centroid. What is the SSE for a cluster consisting of the following three points: (4,8), (9,5), and (2,2)? Show how you arrived at your conclusion.

5. **Metrics II (3 pts.)**

   Sometimes, we decide to split a cluster in order to reduce the SSE. Suppose a cluster consists of the following three points: (3,0), (0,7), and (6,5). Calculate the reduction in the SSE if we partition the cluster optimally into two clusters. Which of the following is the corresponding reduction: (a) 27, (b) 31, (c) 17, or (d) 36. Show how you arrived at your conclusion.

6. **CURE Algorithm (4 pts.)**

   In certain clustering algorithms, such as CURE, we need to pick a representative set of points in a supposed cluster, and these points should be as far away from each other as possible. That is, begin with the two furthest points, and at each step add the point whose minimum distance to any of the previously selected points is maximum. Suppose you are given the following points in two-dimensional Euclidean space: x = (0,0); y = (10,10), a = (1,6); b = (3,7); c = (4,3); d = (7,7), e = (8,2); f = (9,5). Obviously, x and y are furthest apart, so start with these. You must add five more points, which we shall refer to as the first, second,..., fifth points in what follows. The distance measure is the normal Euclidean L2-norm. Which of the following is true about the order in which the five points are added? (a) c is added fifth, (b) f is added first, (c) f is added third, or (d) b is added second. Show how you arrived at your conclusion.

7. **A Comparative Analysis of Clustering Algorithms (10 pts.)**

   In this exercise, you will compare and contrast the BIRCH, DBSCAN, CURE, and K-means clustering algorithms. In a table, summarize the key differences (e.g., the type of algorithm, space and time complexity, notes concerning applicability, i.e., under what conditions can one employ each algorithm, and their limitations/restrictions). For BIRCH and DBSCAN read the papers that were uploaded in ISIS on May 13th. Note: You are free to identify additional supporting reference materials. In the event you do so, be sure to include your references in your response.

General Instructions

The source code stubs for exercises 1, 2, and 3 are available in our GitLab repository. You will need to upload your results in ISIS as a zip archive (adhering to the structure and naming conventions listed below) by no later June 10, 2016 at 12:00 noon. On Friday, June 10, 2016 you will need to bring a stapled, printed copy of your homework assignment, including the key source code file(s), runs (i.e., the output), and hand-written solutions. Also, be aware that it is plausible you may be asked to meet with one of the instructors to run your codes. Lastly, be certain to work individually, you are free to drop some hints, but try your best to solve these problems on your own.

```
aim3-SS16-<name>.zip
├── author.txt (contains your name and matriculation number)
├── exercise1
│   └── patch file (see README.md file)
├── exercise2
│   └── patch file
├── exercise3
│   └── patch file
└── documentation.pdf (your solutions for exercises 4-7)
```