AIM-3 Scalable Data Science (SS 2017)

Homework Assignment 1

Due on June 2, 2017 at 14:15

Instructions. For exercises 1-4, be sure to show all work, to receive full credit. For exercise 5, be sure to include the reference to your dataset (e.g., URL), a description of the dataset, and your plots, including titles, labels, and findings. For exercise 6, be sure to include your key source codes (i.e., the methods you worked on), screenshots of the unit test results, and when appropriate, the requested Tableau plot. Be aware you may be asked to meet with one of the instructors to run your codes later. **Note**: Source code stubs for exercise 6 may be found in our TU Berlin GitLab repository. You will need to upload your solutions as a single **PDF** file (HW1_lastname.pdf) to ISIS by no later than June 2, 2017 at 14:15. Also, you will need to drop off a stapled, printed copy of your homework assignment at the start of the lecture held on June 2. Lastly, be certain to work individually, you are free to drop some hints. However, you must solve these problems on your own.

1. Bonferroni Principle (Total: 5 points)

Using the information presented in Section 1.2.3 (MMDS book, page 6), **what would be the number of suspected pairs, if the following changes were made to the data and all other numbers remained as they were in that section? Note**. For the cardinality of the event space, use the exact calculation, $\binom{n}{2}$ as opposed to the approximation, $\frac{n^2}{2}$ referenced in the book.

(a) The number of days of observation was raised to 2000. (2½ points)

(b) The number of people observed was raised to 2 billion and there were therefore 200,000 hotels. (2½ points)

2. The Base of Natural Logarithms / Streaming Statistics (Total: 5 points)

(a) In terms of *e*, give approximations to: $(1.01)^{500}$, $(1.05)^{1000}$, and $(0.9)^{40}$. (2½ points)

(b) Given a stream of the following form: one 1, two 2's, three 3's, and so on, up to ten 10's. Compute the 0th frequency moment (F_0), 1st frequency moment (F_1), and the 2nd frequency moment (F_2) or *surprise number*. (2½ points)

3. The Distribution of Distances in a High-Dimensional Space (Total: 5 points)

Prove that if you choose two points uniformly and independently on a line of length 1, then the expected distance between the points is 1/3.

4. Clustering (Total: 10 points)

(a) Describe each of the following clustering algorithms in terms of the following criteria: shapes of clusters that can be determined, input parameters that must be specified, and limitations. (5 points)

Algorithm	Cluster Shapes	Input Parameters	Limitations
K-means			
DBSCAN			
BIRCH			
BFR			
CURE			

(b) 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)? (1 point)

(c) Sometimes, we decide to split a cluster 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. (2 points)

(d) 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, ..., and fifth points in what follows. The distance measure is the normal Euclidean (or L_2) norm. **Which statement listed below is true about the order in which the five points are added?** Show how you arrived at your conclusion. (2 points)

(a) c is added fifth,
(b) f is added first,
(c) f is added third,
(d) b is added second.

5. Visualization in Tableau (Total: 10 points)

In this exercise, you will gain familiarity with **Tableau**. To get you started, have a look at their tutorials: http://www.tableau.com/learn/training. You should allocate a few hours to come up to speed with this tool. Next, you will want to select a dataset (e.g., pick a dataset from among those listed in *Data Sources.pdf* file or a publicly available dataset based on a topic of interest to you). The dataset should be reasonably large (e.g., in the multi MB range). Once you have identified your dataset, explore/analyze the dataset, and prepare at least three distinct plots. For example, try to *identify discernible patterns*, *evidence of natural clusters*, and *appreciable outliers*. Be sure to incorporate an appropriate title, such as the question you aim to answer, corresponding labels to help better understand your data, and your conclusions/key findings depicted below your plot (e.g., as a text box). Also, be sure to document your data source and dataset size.

6. MapReduce Programming (Total: 15 points)

(a) WordCount - Hello World in MapReduce (5 points)

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 text file holding the following data per line: *word*[TAB]*count*. An additional requirement here is that stop words like *to*, *and*, *in* or *the* must be removed from the input data and all words must be lowercased.

(b) A Custom Writable for Prime Numbers (5 points)

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. Thus, your task is to enable this for our custom Writable by implementing *write(DataOutput out)* and *readFields(DataInput in)*. For convenience, a unit test is included to evaluate your implementation.

(c) Postprocessing of Temperature Sensor Readings (5 points)

File *src/test/resources/assignment1/**temperatures.tsv*** contains readings from a fictional temperature sensor, with the following format: *year[TAB]month[TAB]temperature[TAB]sensor-quality*. Your task is to implement a MapReduce program that computes the average temperature per month for each reported year. Also, it must ignore all records that fall below a given minimum quality threshold value (MQTV). The output of your program will be a file consisting of the *year*[TAB]*month*[TAB]*average-temperature*, corresponding to sensor readings that are equal to or exceed the MQTV. Also, be sure to include a Tableau plot of your output, listed in chronological order, with the appropriate *title*, *labels*, and an *interpretation*. Use method *de.tuberlin.dima.aim3.assignment1.**AverageTemperaturePerMonth*** as a starting point. Lastly, a unit test is included to evaluate your implementation.