AIM-3 Scalable Data Science (SS 2017)

Homework Assignment 2

Due on June 30, 2017 at 14:15

Instructions. For all exercises, be sure to show all work to receive full credit, where appropriate follow the additional instructions specific to the individual exercise. Be aware that you may be asked to meet with an instructor to run your codes later. Upload your solutions as a single **PDF** file (HW2_lastname.pdf) to ISIS by no later than June 30, 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 30. Lastly, be certain to work individually, you are free to drop some hints. However, you must solve these problems on your own.

1. Network Analysis (Total: 10 points)

(a) Solve exercise 5.1.1 in the MMDS book, on page 175. Compute the PageRank of each page in Fig. 5.7, assuming no taxation. Show your work including the adjacency matrix, the initial PageRank vector, and the values of the PageRank vector for the first three iterations, as well as an approximation of the converged result rounded to two significant figures. (5 points)

(b) Solve exercise 5.1.2 in the MMDS book, on page 176. Compute the PageRank of each page in Fig. 5.7, assuming $\beta = 0.8$. Show your work including the adjacency matrix, the initial PageRank vector, and the values of the PageRank vector for the first three iterations, as well as an approximation of the converged result rounded to two significant figures. (5 points)

2. Support Vector Machines (Total: 10 points)

(a) What are support vector machines (SVMs)? When are they typically used? (2 points)

(b) Can SVMs be employed for multiclass classification? Discuss the difference between the *one-against-rest* (a.k.a. *one-versus-all*) approach and the *one-against-one* (a.k.a. *one-versus-one*) approach? (4 points)

(c) Suppose that a classification training algorithm requires O(n^r) time for training on a data set of size *n*. Here *r* is assumed to be larger than 1. Consider a data set *D* with exactly even distribution across *k* different classes. Compare the running time of the *one-against-rest* approach with the *one-against-one* approach. (4 points)

3. Specialised Systems (Total: 10 points)

(a) Summarize the key ideas of bulk synchronous parallel (BSP) processing models. This includes the *three* main components in BSP algorithms and *two* key features in BSP models. (5 points)

(b) Compare *vertex centric* vs. *graph-centric* processing models. What are the *key differences*? (5 points)

4. Stream Processing in Apache Flink (Total: 10 points)

In this exercise, you will familiarise yourself with streaming in Apache Flink version 1.3.0.

I. Preparation

As a starting point, read both the Flink Basic API Concepts[1] and the Flink Streaming Guide[2].

II. Foundations

Prior to answering the following questions, compare the Word Count example referenced in the Batch API[3] with the corresponding one referenced in the Streaming API[4].

a. What are the key differences between batch and stream data sources? (1 point)

b. What problem will we encounter, if blocking operators, such as aggregations are used in streaming queries? (1 point)

c. How do streaming engines address the problem referenced just above? Hint: Examine the aforementioned Word Count examples. (1 point)

d. Batch processing engines like Apache Spark simulate streaming behavior using discretized streams (D-Streams). What is the main idea behind D-Streams? (1 point)

e. What are the disadvantages attributed to using D-Streams on a batch processing engine, as opposed to using a runtime engine that offers native streaming support? (1 point)

III. Monitoring a Fleet of Taxis

f. Examine the *TopSpeedWindowing*[5] example and illustrate the query in a flow graph. Describe which data are transferred along the different edges. (2 points)

g. Conduct an experiment based on the **Advanced DataStream API – Sorting Car Events** Exercise found at http://dataartisans.github.io/flink-training/exercises/carSort.html. Although the reference solution is available, try to solve this problem on your own. You should submit both your program, a snapshot of the execution of the program (just a short block will suffice), and a snapshot of the flow graph as depicted in the Flink Dashboard[6]. (3 points)

[1] Basic API Concepts - https://ci.apache.org/projects/flink/flink-docs-release-1.3/dev/api_concepts.html.
[2] Flink DataStream API Programming Guide - https://ci.apache.org/projects/flink/flink-docs-release-1.3/dev/datastream_api.html.
[3] Word Count Batch - https://github.com/apache/flink/blob/master/flink-examples/flink-examples-batch/src/main/java/org/apache/flink/examples/java/wordcount/WordCount.java.
[4] Word Count Stream - https://github.com/apache/flink/blob/master/flink-examples/flink-examples-streaming/src/main/java/org/apache/flink/streaming/examples/windowing/WindowWordCount.java.
[5] TopSpeedWindowing - https://github.com/apache/flink/blob/master/flink-examples/flink-examples-streaming/src/main/java/org/apache/flink/streaming/examples/windowing/TopSpeedWindowing.java.
[6] Apache Flink Dashboard - https://ci.apache.org/projects/flink/flink-docs-release-1.2/quickstart/run_example_quickstart.html.

5. PageRank (Total: 20 points)

(a) In this exercise, you will gain familiarity with **GraphLab Create**.

- First, download the tool. Visit https://turi.com. Under the GRAPHLAB CREATE menu, select *Register for Academic Use*. Complete the form. Return to the GRAPHLAB CREATE menu, then select *Install* to download the tool.

- Next, you will want to get acquainted with the **GraphLab Create API Documentation**, https://turi.com/products/create/docs/index.html.

- Then, you will use its **Graph Analytics Toolkit** to conduct a series of experiments, namely, involving **Connected Components** and **PageRank**.

- As a *warm-up exercise* find the connected components for the graph of the 2014 World Cup knockout state results. How many connected components are there?

- Compute the PageRank score of every node in the Stanford Web Graph, http://snap.stanford.edu/data/web-Stanford.html. You will need to submit your GraphLab program, a screenshot of the execution of the program, and a histogram of the PageRank scores. For this exercise, you are free to use any visualization tool (e.g., the *GraphLab* visualization capabilities to generate an image of the graph, *Gephi*: An Open Graph Viz Platform, https://gephi.org, or *Tableau*). (10 points)

(b) Repeat (a) using Apache Flink[7] version 1.3.0. Are the results consistent between the two systems? If not, can you explain why this is so? (10 points)

[7] PageRank - https://ci.apache.org/projects/flink/flink-docs-release-1.3/dev/batch/examples.html#page-rank.