Distributed Computing and Storage Architectures: Project MapReduce

Prof. Nikolaos Deligiannis
nikolaos.deligiannis{*@vub.be*}

Assistants:
Abel Díaz Berenguer Kasper Cools
{{*abel.diaz.berenguer, kasper.cools*}*@vub.be*}

2024-2025

Project Overview

In this project you will practice your skills on MapReduce algorithms in Python using MRJob[1]. The data you will be working on includes movie titles from MovieLens [1], Google web graph [2], KNN (k-nearest neighbors) [2] dataset and a large matrix. For each of the provided data you are asked to perform a set of tasks such as counting, web-link graph reversion and matrix manipulation.

MRJob

MRJob enables us to write MapReduce jobs in Python and run them on several platforms, being it locally, a Hadoop cluster, EMR, Dataproc, or spark jobs on a Hadoop cluster. For installation we refer to the documentation manual[3]. Below is an example where the mapper will receive one line from a text document and emit a lowercase version of each word in the line and a 1.

```
from mrjob.job import MRJob
import re
WORD_RE = re.compile(r"[\w']+")

class MRWordFreqCount(MRJob):
        def mapper(self, _, line):
        for word in WORD_RE.findall(line):
                yield word.lower(), 1

        def combiner(self, word, counts):
                yield word, sum(counts)

        def reducer(self, word, counts):
                yield word, sum(counts)

if __name__ == '__main__':
        MRWordFreqCount.run()
```

Using the command below we can feed the map reduce jobs our files and test our algorithm locally.

```
python3 word_counter.py in.txt
```

[1] https://github.com/Yelp/mrjob
[2] https://snap.stanford.edu/data/web-Google.html
[3] https://mrjob.readthedocs.io/en/latest/

1 The MovieLens Datasets[4]

In the **1_MovieLens** folder you will find:

1. **movies.csv**: CSV dataset of the basic information on movies. Parameters include: [movieId,title,genres]

1.1 TASK 1: Top 10 keywords for each movie genre

For each possible **movie genre** find the top 10 most common keywords within their titles using MapReduce. Try to avoid numbers (years), auxiliary verbs, prepositions, articles and conjunctions as keywords. You can use libraries like NLTK to help you with the latter part.

2 Google web graph [5]

In the **2_GoogleWebGraph** folder you will find:

1. **web-Google.txt**: TXT file of a web graph in which nodes represent web pages and directed edges represent hyperlinks between them. Parameters include: [FromNodeID, ToNodeId] [Source, Target] -> [Target, Source] -> Concat Source

2.1 TASK 2: Reverse web-link graph

Reverse the web-link graph from the given Google web graph file (web-Google.txt).

3 k-nearest neighbors classification[2]

In the **3_KNN folder** folder you will find:

1. **Iris.csv**: The data set consists of samples from each of three species of Iris (Iris setosa, Iris virginica, and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals in centimeters.

3.1 TASK 3: k-nearest neighbor

There are some iris without species in Iris.csv. Implement k-nearest neighbors (K=15) by using MapReduce to classify those unknown species of Iris based on four features. The result should contain those iris' ID and your classification. Use the Euclidean distance to measure the distance in KNN.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

Please don't forget to normalize the features before computing distance. You can store the maximum and minimum values of features/store the unknown species samples by using attributes of the Python class.

[4] https://files.grouplens.org/datasets/movielens/ml-25m-README.html

[5] https://snap.stanford.edu/data/web-Google.html

4 Frobenius norm

In the **4_MATRIX** folder you will find:

1. **A.txt**: A matrix of 1000 rows and 50 columns

4.1 TASK 4: the Frobenius Norm of a given matrix

The Frobenius norm is the matrix norm of an matrix defined as the square root of the sum of the absolute squares of its elements.

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}$$

Calculate the Frobenius norm of the given matrix. Please calculate the sum of the squares of the elements of the same row in one reducer ($\mathrm{sum}_i = \sum_{j=1}^{n} |a_{ij}|^2$), and then calculate the norm in another reducer $||A||_F = \sqrt{\sum_{i=1}^{m} \mathrm{sum}_i}$.

Comments

All of the produced code should be written in **Python** and has to be accompanied with explanatory comments. All tasks are required to be implemented within the MapReduce framework in a distributed fashion.

Requirements

This is a solo project, which means everyone has to do this project alone. Please do not copy codes or reports from other people's projects, otherwise you will get 0 point for this course.

Evaluation

For this project, you will be evaluated based on the following criteria:

- Your written report.
- Your source code (with explanatory comments).

Deliverables

Upon finishing the project, you need to submit:

- A report describing your work and results for each task in the project;
- The full source code of your project, with clear comments;
- A detailed guideline to run your code, including the dependencies, the commands to run, etc...;

Deadline

The deadline for the submission of the project report and material is **23:59 CET, Sunday 5 January 2025**. Any submission after this deadline is considered invalid. You will have to combine the deliverables in a .zip folder and submit them through email to {abel.diaz.berenguer@vub.be} and {kasper.cools@vub.be}.

References

[1] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), Dec. 2015.

[2] Wikipedia. k-nearest neighbors algorithm. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, 2021.