**Data Science Project Report**

**1. Principal Investigator**

Christopher Tam – [tamc1@hawmail.newpaltz.edu](mailto:tamc1@hawmail.newpaltz.edu) (1)

Ryan Berry - [berryr1@hawmail.newpaltz.edu](mailto:berryr1@hawmail.newpaltz.edu) (2)

Jeremy Falstein - [falsteij1@hawmail.newpaltz.edu](mailto:falsteij1@hawmail.newpaltz.edu) (3)

1.1 Individual Contribution Breakdown (list the percentage)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Member 1 | Member 2 | Member 3 | Total |
| Introduction | 60 | 30 | 10 | 100% |
| Background | 10 | 20 | 70 | 100% |
| Implementation | 50 | 30 | 20 | 100% |
| Experiment Results and Discussion | 40 | 40 | 20 | 100% |
| Conclusion | 30 | 40 | 30 | 100% |
| Other contribution and explain | 20 | 60 | 20 | 100% |

**2. Title of Project**

ANALYZING DATASETS WITH K-MEANS ALGORITHM

**3. Mentoring**

Professor Min Chen, Department of Computer Science, SUNY-New Paltz

**chenm@newpaltz.edu**

**4. Introduction**

4.1 Project Motivation:

Initially, the goal of the team was to conduct an experiment in which we analyse some arbitrary dataset with the machine learning algorithm that we all agreed on.

4.2 Aims and Objectives:

Given that this is an introduction to Data Science course, our objectives made very clear by our mentor:

* Design a MapReduce based Algorithm that has the capability of handling huge datasets, by parallelizing it.
* Finding the proper tools to aid in the development of our algorithm.
* Finding the proper dataset that is able to demonstrate the effectiveness and efficiency of our algorithm.

4.3 Report Structure:

In this following report we will begin with the background of our study, as well as how we implemented our algorithm. We will then explain the dataset we have inquired through our research and explain the results we have retrieved from performing our experiment.

**5. Background/History of the Study**

Finding diabetic studies of hospitalized patients is extremely rare. In order to effect some type of change in this area of patient care, it would deem necessary to perform such a study. Analyzing a data set would provide information needed for a study which would provide information on how to positively affect and improve care of diabetic patients. K-means will be used to find a relationship between number of lab procedures vs number of medications.

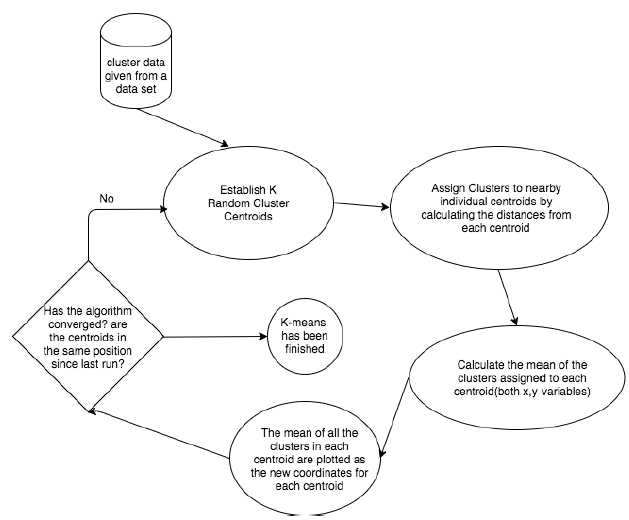
A fairly large data set was used throughout the study, it has over 100,000 instances and 55 attributes. The analysis of this large data set with 10 years worth of data will not only help with results, but provide a blueprint for better patient care in the near future. There is also data about laboratory tests as well as medications which were administered to the patient during the encounter. Information about the specifics of patients is apart of this dataset as well: Number, race, gender, age etc.

Our goal was to make a prediction on whether there was any correlation between the number of lab procedures and number of medications they have taken at the hospital. We believe that the more lab procedures that a patient has been administered, the lower the amount of medications they would be consuming during their stay.

**6. Approach and Implementation**

We started out by finding the proper tools to develop our algorithm. We initially agreed on using python as our programming language for this project. Python is becoming a well respected, and highly supported language in Data Science. Companies have used Python for it’s minimal code usage to quickly extract data. With the addition of Python, we have also decided on using a reputable framework that does MapReduce easily.

Mrjob is a Python MapReduce framework that gives the ability for users to write their code and test it using Hadoop, Amazon EMR, or locally. We found mrjob to be the quintessential tool for anyone who desires to perform MapReduce jobs with little to no code.

As we mentioned before, mrjob is capable of effortlessly parallelizing MapReduce with little to code no at all! In the python command line prompt, one can easily add this snippet of code (-r emr) to the end of the line; mrjob should automatically parallelize your job to your EMR cluster (assuming that you have pre configured mrjob to access your aws account).

*Figure 1.- A Diagram demonstrating how the k-means algorithm is performed.*

Our implementation of k-means was developed using the ideas that were demonstrated by a professor from the University Of Chicago (Sotomayor). The program will perform two MapReduce jobs in two separate files. The folder of the project contains three python files: kmeans.py, kmeans-selector.py, and kmeans-updater.py.



*Figure 2. - Our k-means implementation through MapReduce (mrjob)*

The objective k-means selector file is to perform a mapreduce job that selects the centroids. It starts off by doing the a mapper job that retrieves all of the coordinates, and finishes by pushing to the combiner a list of coordinates that were extracted from the dataset. Next, the combiner iterates through the list of coordinates to find the maximum, and minimum coordinate within the dataset. It then proceeds to push the two variables to the reducer as a list file. The reducer uses those coordinates and computes the initial centroids based on the number of cluster that we use.

The goal of the k-means updater file is the update the centroid position. The updater starts off with the mapper which assigns coordinates to centroids nearest from their location. The cluster acts as the key, while the coordinate acts as the value. The mapper then pushes all of the values as a list to the combiner. The combiner then retrieves the clusters, and coordinates list to combine the values into separate lists for each individual cluster. Each list is associated with their specific cluster, and are pushed with the key cluster, and list of coordinate list values to the reducer. Lastly, the reducer retrieves the clusters, and the list of coordinates for each cluster. Given the coordinates, the reducer computes the average of all of the coordinates for each individual cluster, and returns the key(cluster), and the value(average of the coordinates for that cluster).

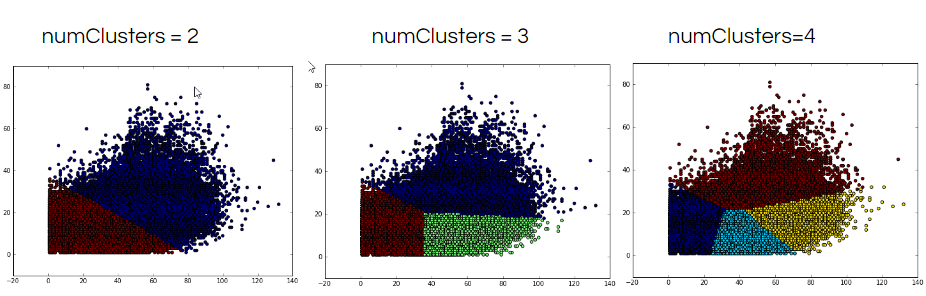
The original k-means file is used to perform the actual implementation of k-means. We have used MapReduce in the selector and updater to assist in processing our large dataset. The program begins by creating an instance of the selector to retrieve the initial list of centroids for the k clusters we are looking for. The centroids are written to a file to be used later on. We then create a while loop where we will perform the update centroids. The centroids are inputted into the running instance of the centroid updater, which returns the updated list of centroid values. We then compare the new values to the old coordinates to check if the algorithm has converged. We do this by comparing the difference between the new centroids and the old centroids. The while loop will break if the program has converged.

We have decided to go against the use of parallelization of our algorithm. This is because our dataset would not be suitable for a Elastic MapReduce, as the file would take forever to process our job. We discovered for our dataset, that it would be more appropriate to perform the MapReduce job locally, then diverting to Hadoop or EMR.

**7. Experiment Results and Discussion**

After applying the our diabetes dataset to the k-means program, the program should print out the number of iterations that have occurred for each cluster. The final position of the centroids is given after the k-means algorithm has converged. The results can be seen inside the project folder as ‘centroid\_results.txt’.

Unfortunately there was no decent way to plot the coordinates on matplotlib, However we did have an alternative solution to this issue. We figured that we could compare centroid values to the centroids plotted by the scikit learn plugin’s k-means function. Our centroids could not relate to the centroids given by the k-means function. However, the data we received from the k-means function was easier to display, as it worked in conjunction with the python’s matplotlib library. Matplotlib is a python library that has the capability to create distinct charts to visualize data.



*Figure 3. - Diabetes dataset plotted on matplotlib using the k-means function from scikit learn. Adjusted clusters range from 2 to 4. X axis represents the total number of procedures, while Y axis represents the number of medications provided*

Initially our prediction was that a patient would need less medications if they underwent more tests to help to classify their situation. Based on these charts, high numbers of tests have diminishing returns on the number of medications needed for treatment, generally dropping off at about 60 tests. We’ve realized that the number of lab procedures is most likely not the sole determining factor for the number of medications they need to have administered.

**8. Conclusion**

From this project we discovered that k-means is extremely efficient at clustering a specific type of dataset and is remarkably useless when applied to any other type of dataset.

Our intended goal was to find numerical values that we could cluster effectively, Since we have decided on choosing the k-means as our MapReduce algorithm. Most of the datasets that we looked at came out to be too complicated for us to analyze. We finally settled on a dataset we thought would fit k-means well, but discovered much later that running k-means on this dataset would not give us very much new information. With that being said, if someone wanted to develop insights with a unsupervised machine learning algorithm such as k-means, they must find a dataset that is consistent, unbiased, and make accurate predictions to the data.

**9. References**

*Sotomayor, Borja . "Lab 3." Department of Computer Science. Borja Sotomayor, Spring 2013. Web. Dec. 2016.*

*"Mrjob v0.5.6 documentation." Pythonhosted. Mrjob, n.d. Web. <<https://pythonhosted.org/mrjob/>>.*

*Caraciolo, Marcel . "Introduction to Recommendations with Map-Reduce and mrjob." Artificial Intelligence in Motion. N.p., 12 Aug. 2012. Web. 2 Dec. 2016.*

*Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.*