Data Processing at Scale (CSE 511)

Haibin Liang

Arizona State University

Tempe, US

hliang7@asu.edu

I. Introduction

This is a portfolio report for Data Processing at Scale (CSE 511) course taken in Fall 2021 semester. In this course, we have to complete 2 projects. We were required to implement the concepts and skills we have learnt throughout the course.

A. Project 1 (NoSQL)

For project 1, we were asked to perform query on the NoSQL system by using a provided NoSQL database engine, UnQLite, using Jupyter notebook as the environment and python as the programming language. There are two tasks I needed to complete, to find the Business Based On City and to find Business within a certain area Based On Location.

For the first task, I needed to search the database and find out all the business present in the city given and save to a text file in a specific format.

In the second task, I needed to search the database for all the business present within a certain distance from a given location and then save to a text file with the name of the business only.

B. Project 2 (Hotspot analysis)

For project 2, we are asked to perform geospatial data analysis for a very typical application, hotspot analysis[1]. In this project, we are using Apache Spark, which is a distributed in-memory cluster computing framework that compile resources from multiple nodes to fulfill the complex and computationally intensive query[2]. In the project, we were asked to perform spatial hotspot analysis on a small subset of NYC taxi dataset to find out the hottest pickup location in NYC. This data is always significant in such business type, to correctly allocate the drivers/other resources in according to the needs. There are two tasks for this project:

The first task is HotZone analysis, it requires us to find hotter rectangles. A 'hotter' rectangle is defined by having more points count within the defined boundary of the rectangle.

The second task is HotCell Analysis. A cell is defined as a space time cube where the x axis represents the latitude of the pickup location, y axis represents the longitude of the pickup location and z axis represents the pickup date. The 'hotness' of the cell is analyzed by using G statistic Z-score.

II. Description of Solution

Project 1

The data was provided in a NoSQL database format, which are non-relational database. Each items in the database are stored with {'business_id', 'state', 'name', 'latitude', 'full_address', 'categories', 'open', 'stars', 'city', 'neighbourhoods', 'review_count', 'longitude'}.

A. Apply filter to a NoSQL database (find the Business Based On City)

To find out all the business present in the city given, I first fetch all the data from the data base. I then perform an equality function to find out all the business within the given city. After that, I used the built.in of python to write a list of business with the required information and in order of (Name$FullAddress$City$State). The list was saved as a text file named 'saveLocation1'.

B. Perform Geospatial filtering (Find Business Based On Location)

To search the database for all the business present within a certain distance from a given location. First, I have to define a function to compute the distance between the given 'my location' and the location of the business in given categories.

I used the 'haversine' formula to calculate the great-circle distance between two points.

Haversine formula[3]:

$a = \sin^2(\Delta\varphi/2) + \cos\varphi 1 \cdot \cos\varphi 2 \cdot \sin^2(\Delta\lambda/2)$

$c = 2 \cdot a \cdot \tan2(\sqrt{a}, \sqrt{(1-a)})$

$d = R \cdot c$

In this way, I will get the distance between each business in the chosen categories and my location. Then, I find business of chosen categories within a maximum distance from my location, by:

First, fetching all the data from the data base and filter out the business of the given category by an equality function.

Second, find out the distance between the location of the business and a given my_location.

Third, compare the distance of each business to my_location with the given maximum distance.

Forth, filter out those businesses which are within a given maximum distance from my location.

The names of those qualifying businesses are saved to a text file called 'saveLocation2'.

C. HotZone Analysis

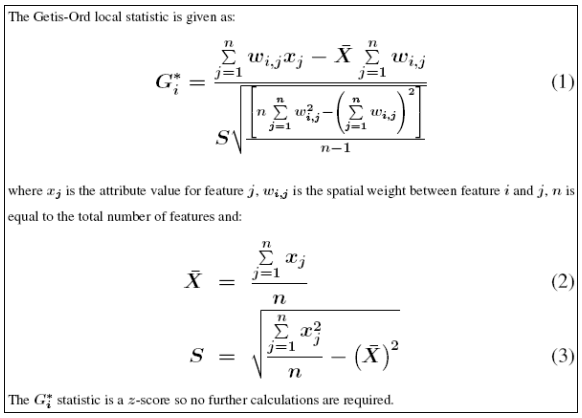
In this task, I will need to do a range join operation on a rectangle datasets and a point dataset. In each rectangle, the number of points located within the rectangle will be obtained. The hotter rectangle includes more pick-up points. In this task, I have to calculate the hotness of all the rectangles and do a ranking.

First, I have to perform a ST_Contains algorithm to check which rectangle contain the point. For ST_Contains, I began with fetching all the x and y coordinates of the point (x,y). Then, I fetch the x and y coordinates of the diagonal points of the rectangle (x1, y1) and (x2, y2). Then, I find the the minimum and maximum x value from x1 and x2 (minX and maxX), and also the minimum and maximum y value from y1 and y2 (minY and maxY). If minx <= x <= maxX and minY<= y <= maxY, then return TRUE else return FALSE.

After that, I perform the Range Join Query to find all (Point, Rectangle) pairs such that the point is within the rectangle by the ST_Contains algorithm. This would return the data containing each rectangle and the points count in each rectangle.

D. HotCell Analysis

In this task, I first need to create a space-time cube of latitude, longitude and time (day) as the three axes and perform Getis Ord Gi* statistics[4].


The Getis-Ord local statistic is given as:

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j}x_j - \bar{X}\sum_{j=1}^{n} w_{i,j}}{S\sqrt{\frac{\left[n\sum_{j=1}^{n} w_{i,j}^2 - \left(\sum_{j=1}^{n} w_{i,j}\right)^2\right]}{n-1}}} \quad (1)$$

where x_j is the attribute value for feature j, $w_{i,j}$ is the spatial weight between feature i and j, n is equal to the total number of features and:

$$\bar{X} = \frac{\sum_{j=1}^{n} x_j}{n} \quad (2)$$

$$S = \sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - \left(\bar{X}\right)^2} \quad (3)$$

The G_i^* statistic is a z-score so no further calculations are required.

Figure 1. Steps for calculating the z-score.

This cube is the HotCell in this task. It is a three-dimensional space with x val corresponding to the latitude of pickup location, y val corresponding to the longitude of the pickup location and z val corresponding to the pickup date. The step between each cells in x and y is 0.01 and z is 1 day. Thus, the value of the cell will return the number of trips on a particular day.

Here is the logic of the algorithm, First, I fetch the point from the given input data and stored in a temporary pointPath. Then, from the pointPath, I fetched the latitude, longitude and day on which the taxi pickup occurred. Third, I created the space time cubes as described above by fetching the latitude, longitude and day of each point. Each cell in the space time cube will contain the count of taxi pickups that occurs in that latitude and longitude range (0.1*0.1) on a given day. Forth, I calculated the the Getis Ord Gi* statistic to get the Z-score for every cell in the space time cube, followed by ranking the values of fifty cells in the descending order according to the Z-score. This Getis Ord Gi* statistic is used to find out the most significant hot spot cells in the space-time cube. The Z-scrore represents the statistical significance of clustering for a specified distance. The larger the Z-score the more intense clustering it is (the hotter).

III. RESULTS

A. Project 1

I successfully performed query on a NoSQL database. In task one, my algorithm successfully fetches out all the business within a given city. In the test case, the output of my result, when the input is city == Tempe, matches the output data. I found three business in the city of Tempe. I also successfully stored the data in the format required, (Name$FullAddress$City$State), and the output would be:
[VinciTorio's Restaurant$1835 E Elliot Rd, Ste C109, Tempe, AZ 85284$Tempe$AZ
Salt Creek Home$1725 W Ruby Dr, Tempe, AZ 85284$Tempe$AZ
P.croissants$7520 S Rural Rd, Tempe, AZ 85283$Tempe$AZ]
In task two, I successfully calculated the distance between two locations, and set a constraint to filter out the business of a certain category that is within a given distance to the given my_location. In the test case, FindBusinessBasedOnLocation(['Buffets'], [33.3482589, -111.9088346], 10, saveLocation.txt, data), the output gives the correct output: ["VinciTorio's Restaurant"].

The algorithm passed the first 7 cases in the auto-grader.

B. Project 2

In task 1, I successfully performed the ST_contrains algorithm and performed a range join query for the (point, rectangle pair) to get the count of the number of points within each defined rectangle. I run the program with the test cases and the output results matched the answer-key results provided.

In task 2, my algorithm was able to create the space-time cube and performed the Getis Ord Gi* statistics to find the z-score of each of the cube. It was also able to rank the 50 cells in descending order according to the Z-score, in according to x desc, z, desc and then y desc.

The algorithm passed the remaining 8 cases in the auto -grader.

IV. LESSON LEARNED

In the first project, I have learned to fetch, search and save data from a NoSQL database, by performing query using a NoSQL database engine. I tried doing the project using both UnQLite and MongoDB (not shown here) to get familiar and practice with these common NoSQL database engines. I also tried some different ways to perform query, (ie. Using filter instead of equality), e.g. b_docs = collection.filter(lambda O: O['city'].decode()==(cityToSearch)). This trained my skills of python coding. Also, I have gained the idea of how a key-stored NoSQL database looks like. I also have an impression of how a NoSQL system can be so well-adapted to the heavy demands of big data**.**

In the second project, I have learnt to work on Apache Spark and using scala. They were new to me but I managed to master it. However, by learning Apache Spark coupled with Hadoop really helped me understand the need and experience the advantage for a Distributed and Parallel Database System[5]. I also learnt to deal with the geospatial big data, to use the Getis Ord Gi* statistics as a parameter to calculate the hotness/popularity of a geospatial location. Such algorithm has a wide range of applications in transportation, business and environmental[4], [6].

Overall, although this project is tough, it is a very good learning experience. It pushed me to explore different resources on the internet as well as push me to speed up my learning curve towards a new programming language. This is beneficial not only to the future MCS Courses but also to my lifelong learning in computer science as it is an area which requires constant continuous learning.

V. Reference

[1] P. Gatalsky, N. Andrienko, and G. Andrienko, "Interactive analysis of event data using space-time cube," in *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, 2004, pp. 145–152.

[2] A. G. Shoro and T. R. Soomro, "Big data analysis: Apache spark perspective," *Glob. J. Comput. Sci. Technol.*, 2015.

[3] C. C. Robusto, "The cosine-haversine formula," *Am. Math. Mon.*, vol. 64, no. 1, pp. 38–40, 1957.

[4] P. Songchitruksa and X. Zeng, "Getis–Ord spatial statistics to identify hot spots by using incident management data," *Transp. Res. Rec.*, vol. 2165, no. 1, pp. 42–51, 2010.

[5] J. G. Shanahan and L. Dai, "Large scale distributed data science using apache spark," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 2323–2324.

[6] P. Songchitruksa, X. Zeng, F. Rossi, and G. Becker, "Creating forest management units with Hot Spot Analysis (Getis-Ord Gi*) over a forest affected by mixed-severity fires," *Aust. For.*, vol. 2165, no. 4, pp. 42–51, 2010.