Ira A. Fulton Schools of Engineering
Arizona State University

CSE 511: Data Processing at Scale

Hot Spot Analysis Project Report

Arda KOCAMAN
Master of Computer Science, Fall 2024
akocaman@asu.edu

Reflection

This project required analyzing taxi pickup location data to analyze high density areas in NYC. The analysis is done on spatial temporal data using Apache Spark and Scala. This assignment is taken from ACM GISCUP 2016 competition. The primary goal was to perform two tasks: hot **zone** analysis and hot **cell** analysis. For Hot Zone Analysis, the task involved identifying the density of points (pickups) within specific geographic zones (rectangles). For Hot Cell Analysis, the focus was on calculating relative density of geographic areas. I started by setting up the development environment using the provided template. Dependencies and configurations like versions were set up for compatibility with the Autotrader's requirements (like Spark 2.4.8). For build/compile and implementation, IntelliJ IDE was configured with Spark and sbt. My approach to this problem started with understanding data. In a separate test file, I ingest the data to dataframe just to understand the structure of it. It was also my first time using scala so it was beneficial to first implement a simple file to build and run the file to get output. After understanding the format, I started to work on the two tasks. For Hot Zone calculation, it was important to understand calculating how many points included given a geometric rectangle, namely *ST_Contains* function. For Hot Cell, it was crucial to understand and figure out the implementation of G-score (Getis Ord) to calculate autocorrelation of a rectangle regarding to its neighboring zones, in this case rectangular cells. After understanding at theoretical level, I try to figure out how to implement these on software engineering perspective. There are utils files to write the required functions and there were given structure to run on analysis files. Finally, I understand how to test and create a .jar file to submit.

Lessons Learned

First principal thing I've learned was, how to work with dataframes in Spark. In my professional career, I also work with data but I've never used Spark before. My to-go processing library is pandas. In many cases I do understand using pandas could be inefficient because it's a single node library and cannot process very large collection of data. According to the GISCUP website[1], it will be processed with 25 computers each 24GB RAM. Spark is distributed processing library that can take care of distributed scheme and for this case with over 1B data points, it would be smarter to use. SQL-like interface and built-in fault

tolerance mechanism makes it easy to use and adapt for the user.

Another thing I learnt during this project was how to work with spatial data. I realized spatial data is not very different from the standard data that we are dealing with daily, it's just some functions that would make computation easier. For this case, learning the point and the area representations were important. Also the functions, like *ST_Contains* were new to me. I assume this function's implementation in pandas would require nested loop structure, which may be very costly to compute due to memory restrictions. Spatial metric for autocorrelation, G-score, was also good to know for spatial statistics.

Finally, coding in scala and for implementing on distributed systems paradigms were one of the essentials I learned during the project. For example,

```
val pickupCounts = pickupInfo.groupBy("x", "y", "z").count()
```

this line calculates number of pickups aggregated for each coordinate. In distributed development, each coordinate can be computed independently and aggregate results in final step. This speeds up the process. Also, as someone who's use to code in Python, coding in scala fresh up my skills.

Implementation

Hot Zone Analysis calculate the density of areas given according to the number of pickup point within them. The areas are defined with two points and the points are defined as longitude and latitude. Aforementioned function gets an area and a point that outputs a boolean value, true or false value determining whether this point lays on this area. Analysis include extracting the points and the rectangles, then filtering the rectangles and points within from where clause in Spark SQL, finally aggregate and count the number of points each rectangle and sort them. The result is kept in a dataframe that will output the desired result.

Hot Cell Analysis calculate statistically significant areas. The significance is determined by how dense the area is compared to its neighboring areas, to be more precise, rate of point change compared to the adjacent cells. The G-score calculates this feature. It is defined by how much the value of a cell and its neighboring cells' total (pickup counts) deviate from the overall average of all cells, considering the number of neighbors and the data's variability. To calculate G-score, first average pickups calculated, then spread of pickups (standard deviation), and adjacent cell finder function. This function is done by self-join in SQL, comparing the edge location of each cell with other cells in three dimensions. If one of the edge location equals to the compared cell's edge location, then they are adjacent. This could also be implemented with nested queries but I realized self-joins might require less intermediate steps and avoid excessive memory usage. Finally, G-score is calculated according to the adjacent cells of each cell and sorted in descending order. The result is the top 50 cells of this dataframe. Those are the 'hot' zones that we are looking for.

References

https://sigspatial2016.sigspatial.org/giscup2016/home
https://postgis.net/docs/ST_Contains.html
https://carto.com/blog/spatial-hotspot-tools