Project 2: Hot Spot Analysis

CSE 511: Data Processing at Scale

Lucy Zhang

Ira A. Fulton Schools of Engineering, Arizona State University
lzhan370@asu.edu

Reflection

The hot spot analysis project provided a platform to comprehend the utilization of Scala and Apache Spark in managing geospatial data. Given the complexity of spatial queries on a large database, the primary approach was to familiarize myself with Apache Spark's dataframes to handle big data effectively. The project required proficiency in Scala programming and understanding the intricacies of spatial data manipulation.

In the Hot Zone Analysis task, the first approach was to perform a range join operation on rectangle datasets and a point dataset. To accomplish this, I used the ST_Contains function to ascertain whether a point resides within a rectangle. Once established, I proceeded to calculate the hotness of the rectangles, which is defined by the number of points located within each rectangle.

On the other hand, the Hot Cell Analysis required a deeper understanding of spatio-temporal big data, specifically in calculating the Getis-Ord statistic. With the help of the coding template provided, I established the cell coordinates and focused on developing the logic for the Getis-Ord statistic computation.

Lessons Learned

This project offered several learning opportunities, specifically in the management of large geospatial datasets. I learned how to use Apache Spark's dataframes to perform complex spatial queries. It also improved my proficiency in Scala, which is essential for developing Spark applications.

Understanding the Getis-Ord statistic and how it applies to spatial data was an essential learning outcome. The application of this spatial statistic provided insights into the identification of hotspots, which can help businesses make strategic decisions. Additionally, this project emphasized the importance of optimizing queries for big data.

Implementation

Hot Zone Analysis

This involves identifying rectangles (zones) with high activity. The process uses the function ST_Contains(queryRectangle: String, pointString: String) where queryRectangle is the rectangle of interest and pointString represents a point. The function checks whether the point lies within the rectangle, which helps identify the rectangles with more points, thereby representing hot zones.

Hot Cell Analysis

The Hot Cell Analysis process begins by loading the data and determining the cell coordinates. Following are the steps required before calculating the z-score:

Data Transformation: The input data is first transformed into a format suitable for analysis. In this case, the pick-up location is extracted and transformed into corresponding cell units.

Square Cell Computation: The square cell computation is done next, each cell unit is calculated with a size of 0.01 in terms of latitude and longitude degrees.

Spatial Weight Calculation: The spatial weight of each cell is computed. This is done by calculating the number of neighboring cells for each cell.

Characteristics Calculation: The attributes of each cell such as sum, square sum, and the count are computed, which would later be used for the z-score computation.

The results from these stages are then used to calculate the z-score which allows us to identify statistically significant spatial hot spots. These steps were implemented within the HotcellUtils.scala and HotcellAnalysis.scala files in the coding template.

In conclusion, this project has enabled a deeper understanding of the application of Scala and Apache Spark in handling geospatial data, with particular emphasis on hotspot analysis. The ability to identify high activity regions using spatial data can be invaluable in decision-making processes for businesses and organizations.