Project Phase 3: GISCUP 2016 Challenge

CSE 512, Spring 2017

Namrata Nayak (1212408767) Varun Chandra Jammula (1207940275)

Shrinivas Bhat (1212415813) Nimshi Venkat (1209396432)

Problem Definition:

Apply spatial statistics to spatio-temporal big data in order to identify statistically significant spatial hot spots using Apache Spark.

Input:

A collection of New York City Yellow Cab taxi trip records spanning January 2009 to June 2015. The source data may be clipped to an envelope encompassing the five New York City boroughs in order to remove some of the noisy error data (e.g., latitude 40.5N – 40.9N, longitude 73.7W – 74.25W).

Output:

A list of the fifty most significant hot spot cells in time and space as identified using the Getis-Ord statistic.

Algorithm:

We have three java classes for this task. The class MathUtility.java deals with all the mathematical calculations for calculating Getis-Ord statistic, the class Cell.java provides all the cell details and the class HotSpots.java handles the steps required to calculate Getis-Ord statistics and saves top fifty hot spot cells in the output file.

We have taken a few constants like the minimum and maximum latitude and longitude as per the problem statement. Each cell unit size is 0.01 * 0.01. The total number of cells is calculated as the product of (number of_days * latitude_range * longitude_range). Also, the program takes an input file and an output file as the parameters.

After creating the space time cube, we calculated the z-score for each cell as per the formula for calculating the Getis-Ord statistics (G_i^*).

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

where,

x_j : attribute value for cell j

$w_{i,j}$: spatial weight between cell i and j

n : total number of cells.

The standard deviation (S) is calculated as below:

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

where,
Summation of x_j^2: sum of squares of all the points in the space-time cube
n : total number of cells

The mean ($\bar{X}$) is calculated as below:

$$\bar{X} = \frac{\sum_{j=1}^{n} x_j}{n}$$

where,
x_j : attribute value for cell j

n : total number of cells

Functions Used:

- collectData()

  In this function, we use mapreduce on each record in the input file to return a list of tuples where, each tuple is in the format (latitude, longitude, day).

- createSpaceTImeCube()

  In this function, we create 3-D array of the space time cube using the list of tuples that was returned by the collectData() function.

- getOrdisNumerator()

  In this function, we calculate the sum of spatial weight of neighbors for each cell in the space time cube and also the total attribute cost of all neighbors.

- getOrdisDenominator()

  In this function, we calculate the sum of square of spatial weight of every neighbor for each cell in the space time cube and also the square of the spatial weight of all neighbors.

- computeOrdisValue()

  In this function, we calculate the z-score for each cell and save it in a priority queue.

- getTopkHotSpots()

  In this function, we call the above mentioned methods to obtain the priority queue which contains the z-score for each cell in the space time cube.

- saveResults()

  Once the z-score is calculated for each point, we create a priority queue and compare the z-score for each cell and store the latitude, longitude, day and z-score of the top 50 most significant cells in the output file provided by the user.