Introduction:

Spatial Hotspot Analysis is a technique to identify significant hotspots using the Getis-Ord Statistics[1]. The Getis-Ord Statistic is a measure that uses the values in the current cell and those of the neighbouring cells that depicts how hot that cell is. In this phase of the project, we find the hotspots in the New York Taxi, Yellow cab trip data for the month of January 2015 using the Getis-Ord Statistic. The resultant Z-scores from the Getis Ord statistic helps us understand which latitude and longitudes in the dataset are hot for New York City, Yellow Cab trips for January 2015. Getis Ord statistics is given by,

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

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

Algorithm:

1. Define the following constant values, to be used in algorithm:

minimumLatitude = 40.50

maximumLatitude = 40.90

minimumLongitude = -74.25

maximumLongitude = -73.70

2. Calculate derived fields based on those values, to create a dimension matrix as:

latitudeCount = (maximumLatitude - minimumLatitude + 0.01) / 0.01

longitudeCount = (maximumLongitude - minimumLongitude + 0.01) / 0.01

cellCount = latitudeCount * longitudeCount * 31

3. Create a 3-dimensional matrix:

dimensionMatrix[latitudeCount][longitudeCount][31]

4. Read the data from csv line by line, and get the values of latitude, longitude, and date. We will only use the rows of data where values of longitude, latitude falls between the range given in problem statement.

5. Then normalize the values of longitude, latitude, and date so that we can use the three values as indices of a 3-d matrix (from 0 to some value).

6. After normalization, we store the values in a map. The keys of map will be of the form: latitude,longitude,date.

And values will be of the form: Number of entries having the same key.

7. Then we create a dimension matrix, iterate through the map and store the values as:

dimensionMatrix[dim1][dim2][dim3] = mapEntry.getValue()

where dim1, dim2 and dim3 will represent the latitude, longitude and date corresponding to the mapEntry respectively.

8. Then we calculate the Z-score for each cell of dimensionMatrix by using the formula given in problem description and store the values in a priority queue. The priority queue contains Node objects. Node is a user defined class, which contain 4 fields: latitude, longitude, date, and score. We have provided an implementation of comparator to sort the priority queue values in descending order.

9. Once we are done with calculation for all the cells, we write the first 50 values to a file, which will be our desired output. These first 50 cells will correspond to the hottest cells.

Assumption(s): The weight of neighbouring cells: 1

Weight of non-neighbouring cells: 0

Result:

The program was tested on Spark Framework and the output of our program is a set of the top 50 hottest cells each defined by a set of 3 attributes latitude, longitude, and date. The results allow us to analyse which cells are the hottest for the Yellow cab trip in New York City for the month of January in year 2015. The output is in the following format,

cell_x, cell_y, time_step, zscore

References:

1. http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-statistics-toolbox/hot-spot-analysis.htm