Hotspot Analysis With Spark

Kevin Anthony Grifo
kgrifo@asu.edu

Introduction

The goal for the group project is to analyze and process NYC Yellow Taxicab spatial data. The initial phase of the project is to set up and create the initial queries. The second phase of the project is to identify hotspots in the spatial data.

For the initial phase of the project we created basic spatial queries. We created range, range join, distance, and distance join queries in Scala.

- **Range query**: Given a query rectangle R and a set of points P, find all the points within R.
- **Range join query**: Given a set of rectangles R and a set of points P, find all pairs such that the point is within the rectangle.
- **Distance query**: Given a fixed point location P and distance D, find all points that lie within a distance D from P.
- **Distance join query**: Given two sets of points P1 and P2, and a distance D, find all pairs such that p1 is within a distance D from p2.

To do this, user defined functions were created. These user defined functions are ST_Contains and ST_Within.

- **ST_Contains**: Given a point P and a rectangle R, determine if the rectangle contains the point [3].
- **ST_Within**: Given two points P1, P2 and a max distance D, determine if P1 and P2 are within D distance from each other [3].

The second phase of the project uses the queries that were written in Scala during the initial phase. There are two parts to this phase: hot zone analysis and hot cell analysis.

Hot zone analysis finds the 'hotness' of each zone. The 'hotness' of a zone is the number of points that are found within a zone. This is done using the Scala functions written before.

Hot cell analysis focuses on applying spatial statistics to spatio-temporal data. The purpose of this is to find the fifty most significant cells based on the Getis-Ord calculation or Z-Score. The higher the result of the Getis-Ord calculation, the more significant the clustering of hot spots [2].

Description of Solution

There were two different milestones that needed to be completed to be able to successfully analyze and process New York City Yellow Taxicab spatial data. The first milestone of the project required us to set up and create the initial queries and UDFs (User Defined Functions). The second milestone then used the UDFs defined in the first milestone to identify hotspots in the spatial data.

For the first milestone of the project we created basic spatial queries. The queries we created included range, range join, distance and distance join queries using Scala, a programming language based off of Java. For the range and range join queries we utilized the ST_Contains UDF. The ST_Contains UDF checked to see if a given point fell within a rectangle given two of the rectangle's corner points (bottom left and top right). Please view figure 1 for a graphic representation of how ST_Contains was implemented.

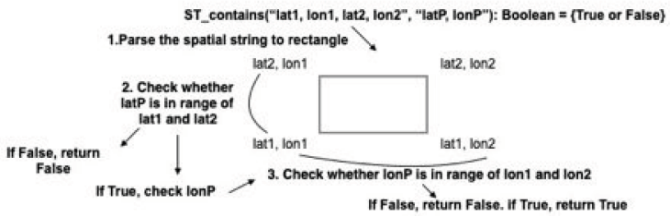

Figure 1 (ST_Contains Graphic Representation)

For the first project milestone the team also had to design the ST_Within user defined function for the distance and distance join queries. This UDF examined if two points were within a max distance of each other. This UDF was implemented by calculating the cartesian product between two points. Graphic representation of this function can be seen in figure 2 below.

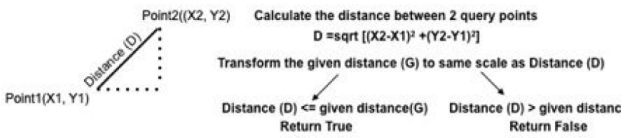

Figure 2 (ST_Within Graphic Representation)

For the 2nd milestone of the project the team was required to perform Hot zone and Hot Cell analysis. For hot zone analysis we had to implement a utility object called HotzoneUtils with a function called ST_Contains. The implementation of the ST_Contains was similar to the implementation of ST_Contains UDF defined for the first milestone. Here we had to check whether or not a point was contained within a given rectangle. We then had to return the count of points within each rectangle, this in turn showed the hotness of each zone/rectangle. See figure 1 for a graphical representation of how ST_Contains works. The 2nd milestone also required us to perform a hot-cell analysis task, which brought a third dimension into the mix, that third dimension being time. For hot-cell analysis, we needed to apply spatial statistics to spatio-temporal big data to identify statistically significant spatial hot spots. Eventually we would return to the user, a sorted spatio-temporal list according to the G value (Getis-Ord), given the taxi trip pickup dataset as input. Figure 3 displays the formula used to calculate the G-value.

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

Figure 3 (G-value formula) [1]

Results

The results of running our project were typically outputted into a .csv file or printed out, as a table, to the terminal/command line. For simplicity the results will be shown in table format. Table 1 shows the top ten hottest zones in New York, given the taxi trip pickup dataset as input. The first column in the table provides two points of each rectangle, the bottom left and top right corners. The second column displays the count of points, using ST_Contains, that are contained within the rectangle in the first column.

rectangle	counts
-73.789411,40.666459,-73.756364,40.680494	1
-73.793638,40.710719,-73.752336,40.730202	1
-73.795658,40.743334,-73.753772,40.779114	1
-73.796512,40.722355,-73.756699,40.745784	1
-73.797297,40.738291,-73.775740,40.770411	1
-73.802033,40.652546,-73.738566,40.668036	8
-73.805770,40.666526,-73.772204,40.690003	3
-73.815233,40.715862,-73.790295,40.738951	2
-73.816380,40.690882,-73.768447,40.715693	1
-73.819131,40.582343,-73.761289,40.609861	1

Table 1 (Top Ten Hotzones)

Table 2 shows the top ten hottest cells in New York, given the taxi trip pickup dataset as input.

x	y	z	G-value
-7399	4075	15	79.39845
-7399	4075	22	77.06822
-7399	4075	14	76.25263
-7399	4075	29	76.05845
-7398	4075	15	75.61182
-7399	4075	16	75.2817
-7399	4075	21	75.24286
-7399	4075	28	75.0681
-7399	4075	23	74.19426
-7399	4075	30	74.03891

Table 2 (Top Ten HotCells)

In Table 2 the x column represents the x-coordinate, the y column represents the y coordinate, the z column represents time and the G-value represents the Getis-Ord value (also known as z-score). The G-value is used to determine the hotness of each cell in our grid. The higher the G-value the hotter the cell.

Your Contribution

The group project provided students an opportunity to collaborate and work together to achieve a common goal. It allowed us to learn extremely beneficial teamwork, communication and collaboration skills. Each team member worked on project milestones 4 and 5 separately and helped each other out along the way. If team members were having difficulties we, all as a team, would provide hints and help, whether it be helping out with the actual code itself or providing a better explanation of the requirements for the project. I also set up a shared Google Drive folder to allow for effortless team collaboration. I made sure to send an email out to the team once the teams were announced with information on how to connect to the shared Google Drive folder. This gave us a head start on the project and allowed us to collaborate early and make plans to meet, via zoom to figure out how to divide the work. The team as a whole decided to meet at least once a week, to provide a status update on where each team member was on the project, and to discuss any issues they were having. This was a great strategy implemented, which allowed team members to help each other out, similar to how a scrum team does following the agile methodology.

We also decided to split up the team project report as a team. This helped us better divide the work for our report. We created a project template that contained the following sections; overview, business requirements, assumptions, high level architecture, process flows, environment setup and appendix. The section I was responsible for was the process flows section. For this section I created process flow diagrams for milestones 4 and 5. These process flow diagrams provided a visual representation of how the ST_Contains and ST_Within UDF (user defined functions) were implemented for project milestones 4 and 5. The Diagrams started with the submitted input files and terminated with the csv file that was generated at program termination. An example of the ST_Contains process flow is shown in figure 4.

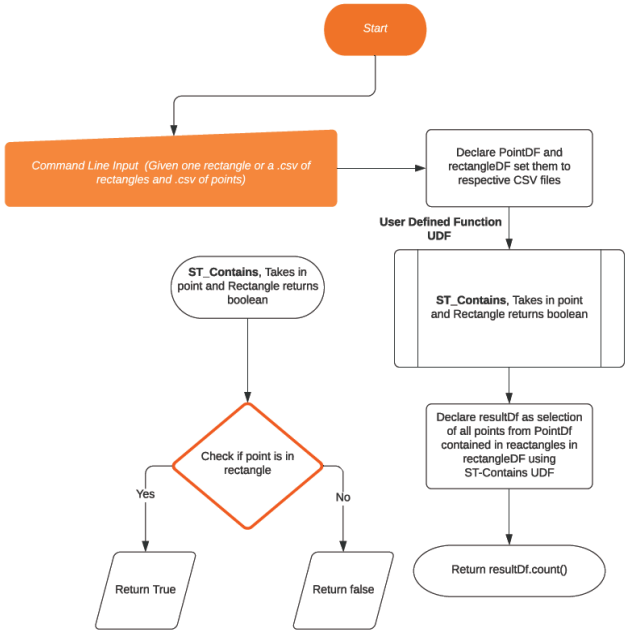

Figure 4 (ST_Contains Process Flow)

As I mentioned in the first paragraph, each team member individually worked on both milestones (the code). Doing so allowed me to learn about how to utilize Spark, SQL and Hadoop with Scala, a java based language. For the first project milestone I learned how to create a user defined function (UDF) and register it, allowing it to be used in SQL spark queries. The first UDF function, ST_Contains checked to see if a given point fell within a rectangle given two of the rectangle's corners (bottom left and top right). We also had to define the ST_Within UDF which checked if two points were within a max distance of each other. This UDF was implemented by calculating the cartesian product between two points. The second project milestone required us to perform Hot zone and Hot Cell analysis. For hot zone analysis we had to implement a utility object called HotzoneUtils with a function called ST_Contains. The implementation of the ST_Contains function was similar to the implementation of ST_Contains in milestone 4. Here we had to check whether or not a point was contained within a given rectangle. We then had to return the count of points within each rectangle, this showed the hotness of each zone/rectangle. For the hot cell analysis the task was focused on applying spatial statistics to spatio-temporal big data to identify statistically significant spatial hot spots using Apache spark. We

were provided min and max values for x, y and z; x and y being spatial coordinates and z being a temporal coordinate. With these coordinate and temporal values we were then required to perform queries on the input data to return the z-score, which we treated as the hotness of each cell based on time. The input data is a dataset of monthly taxi trips from 2009 to 2012. The output was once again a csv file sorted by z-score, with the hottest cells residing at the top.

Lessons Learned

I learned a lot about how to use spark and hadoop with SQL while completing the Hotspot Analysis with Spark project. Some tips I would suggest include setting up the project to work with IntelliJ. The main reason for that is that testing out code changes with intelliJ is much easier, as intelliJ conveniently provides a Scala plugin that works with SBT to compile, build and run your code. Doing this helped me save countless hours, as I didn't need to continuously run sbt assembly to build my jar file that I would eventually run.

Another lesson learned while completing this project was to make sure to communicate with team members early on in the project, to ensure that all the project deadlines were met. This group project enabled the team to learn how to work in a team, together, to meet a common goal. Teamwork and collaboration is a great skill to have in the real world as a majority of employers work in a team setting. To be successful in completing the project I encourage others to be proactive and communicative.

One last beneficial resource to use, that all students have access to, is the discussion forums and the slack channel. The two channels allow for class wide collaboration and help. The live events were also helpful in getting questions answered and acceptance criteria ironed out. They were particularly helpful for the project report requirements, such as the number of pages needed and the main sections that should be included in the report. Utilizing all resources available is extremely important if students plan on being successful in any school setting.

References

[1] A. SIGSPATIAL, "ACM SIGSPATIAL GIS Cup 2016", *Sigspatial2016.sigspatial.org*, 2020. [Online]. Available: http://sigspatial2016.sigspatial.org/giscup2016/problem. [Accessed: 06- Dec- 2020].

[2] A. SIGSPATIAL, "ACM SIGSPATIAL GIS Cup 2016", *Sigspatial2016.sigspatial.org*, 2020. [Online]. Available: http://sigspatial2016.sigspatial.org/giscup2016/submit. [Accessed: 06- Dec- 2020].

[3] C. ASU, "Coursera | Online Courses & Credentials From Top Educators. Join for Free | Coursera", *Coursera*, 2020. [Online]. Available: https://www.coursera.org/learn/cse511/programming/0KL37/project-milestone-5-hot-spot-analysis. [Accessed: 06- Dec- 2020].