Starbucks Locator in United States

Jiao Ma
George Mason University
Dept of Geography and Geosciences
4400 University Drive
2400 Exploratory Hall
Fairfax, VA22030
Jma10@masonlive.gmu.edu

ABSTRACT

In this paper, we describe how to use Leaflet to create a web mapping application which could find all the Starbucks locations and information in United States. Several functionalities like cluster markers, popup information, choropleth map has been executed here. Some JavaScript and CSS library is contained. This small application is used as a test to manage geospatial data. Several technologies are included in this web application. Postgres Database is used to store geospatial, then GeoServer is used to publish initial geospatial data obtained from database to web browser. Last we use Leaflet JavaScript library to modify the output layer. The result is much more friendly and interactive with users.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Starbucks, database, geospatial data, JavaScript, Leaflet, web application, OpenData

1. INTRODUCTION

With the fast development of Internet, GeoSpatial data occupies increasingly important status nowadays. Users are not satisfied with traditional display method to show the data. Instead, they want to have a directly visualization to show result for them. Additionally, intersection between users and output data is attractive. As a result, web GIS is becoming an essential part of all GIS application.

In this paper, geospatial data of Starbucks and United States state shapefile is put into use. Several web maps of this datasets are generated to allow users to intersect with these data and detect their interesting location.

Outline of this paper is: Section 2 describes the data used in Starbucks Locator. Section 3 describe the workflow to output geospatial data. Section 4 describes the user interface of the Starbucks Locator. Section 5 introduces some more functionalities will be added to Starbucks Locator application.

2. GEOSPATIAL DATA

Geospatial data, also known as spatial data, represent the location, size and shape of an object on planet earth such as a building, a lake or a county. Geospatial data could also contain some attributes which could provide more information about the object. Usually, Geographic Information System software could be used to access, manipulate, visualize and analyze geospatial data.

Geospatial data of Starbucks collecting in 2013 is provided by opendata.socrata.com, which is type of open source data. There is a total of 10844 Starbucks records. Besides the geometry attributes, Starbucks Locations data also contains very useful information, such as store name, ownership type, products, service, phone number and location. These detail information could be added later in output layers. Another data used in Starbucks Locator application is GeoJSON data us-states.js. GeoJSON is a format for encoding geographic data structures. It supports multiple kinds of geometry type. Both geometry features and properties could be contained.

3. WORKFLOW

For publishing a web map application, several components need to work together. In the Starbucks Locator application, a combination of Open source software is used. PostgresSQL is applied to store geospatial data, while GeoServer is used to connect the web and publish data from PostgresSQL database to the web.

Original Starbucks data downloads as CSV file, which could be import to PostgresSQL database as a simple table directly. Then we created a second table to transform latitude and longitude number from decimal to geometry type. This table filters the original database to store only interesting attributes in table. Considering the data will be used to present, we keep attributes brand (varchar type), store_number (integer type), store_name (varchar type), ownership_type (varchar type), facility_id (integer type), products (varchar type), service (varchar type), stations (varchar type), phone_number(varchar type), location (varchar type), city (varchar type), state (varchar type), zip (varchar type), geom (geometry point type), inserttime (varchar type), country (varchar type). Among all these attributes, facility_id is the primary

Conference'10, Month 1–2, 2010, City, State, Country.

key for this starbucks4 table. To fast geometry search, a spatial index GIST is added to this table. To achieve full-text search, a full text index is added to this table. And to achieve different query requirement in Geosever, we create file starbucks4 by copying starbucks3 table. Figure1 shows the SQL query [1].

```
CREATE TABLE starbucks4 AS
    SELECT * FROM starbucks3;
```

Figure1: SQL query of copy an existing table

Also, we not only concern about the Starbucks distribution in United States, but want to review other attributes in user interface. For example, we want to know which Starbucks offers lunch for customers, or which Starbucks has free WiFi hotpot. To get result from different kind of query, several columns like service, products and stations will be involved together. However, it is difficult to query from different column considering of the full-text search. To solve this problem, we generate a new column combination to combine different interesting columns information into one column. In this way, we just need to query column combination to get Starbucks with free WiFi hotpot or Starbucks having lunch plan. Column service, products, location are combined to the combination column in Starbucks 3 table. Figure2 shows the SQL query for adding a new column. Figure3 shows the SQL query for merge two columns. This query result will be null only if both values are null.

```
ALTER TABLE Starbucks3
    ADD COLUMN combination character varying;
```

Figure2: SQL query of adding a new column

```
UPDATE Starbucks3 SET
combination = CONCAT(service, ' ', products);

UPDATE Starbucks3 SET
combination = CONCAT(combination, ' ', location);
```

Figure3: SQL query of merge two columns

Geoserver functions like the bridge between database and web application. In the Starbucks Locator application, we use Geoserver to publish data form PostgresSQL database. At the beginning, we create a workspace named 'test', then connect with the PostgresSQL database GGS692_project. When connect successfully, different tables could be chosen as layers to publish to the web. GeoServer supply both web mapping service (WMS) and web feature service (WFS). Web mapping service is usually to publish image to the web, which has little interaction with users. While web feature service publish vector data to the map, which allow users to explore the attributes of layers.

In the Geoserver, we publish two layers, starbucks3 and starbucks4. Both of two layers are published in WFS type. We plan to use starbucks3 to achieve all the queries from users, use starbucks4 to show all the Starbucks locations in United States. By obtaining a research word from user, starbucks3 table combination column will be searched to find the text matching the query word and return the result. Figure4 shows the query language for searching word parameter.

```
    SELECT    store_name,    products,    service,
phone_number, location, geom, combinaton
    FROM starbucks
    WHERE
     to_tsvector('english', combinaton) @@
  to_tsquery('english', regexp_replace(trim('%word%'),
E'\\s+', '&', 'g'))
```

Figure4: SQL query of searching word parameter

A tsvector is used to parse and generate a numerical analogue from a text string. It could detect same word in different tense or plurals. A tsquery is to descibe both the words to be searched for, and the logic to be applied in returning the results.

In order to show all the Starbucks locations in United States, we just use the column country, and all values in this column is 'US'. In this way, all 10844 Starbucks records could be showed on the map.

4. USER INTERFACE

To achieve the interface, many JavaScript or CSS API is used. Leaflet JavaScript API, Google Maps JavaScript API, bootstrap API, mapbox API and jquery are all used in this application. Leaflet API, mapbox API and jquery are used a lot to render and map user interface [5]. Google Maps JavaScript API is used to invoke Google base map. Bootstrap API is applied to set the framework of HTML file and generate navigation bar for the user interface [3]. Figure5 shows the user interface screenshot.

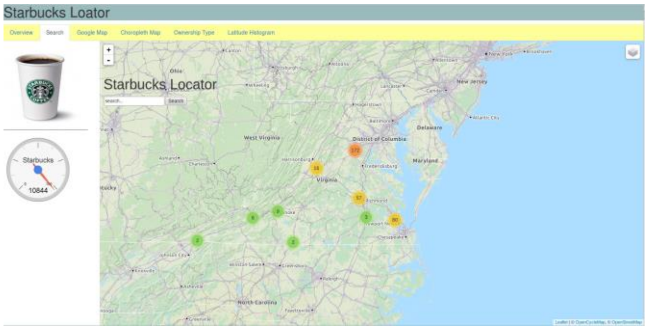

Figure5: user interface screenshot

4.1 Overview Map

The Overview map is basically to provide an entire impression to users. From this layer, users could see all the Starbucks location in cluster maps. As they zoom in or zoom out, the size of the clusters will change too. Multiple layers could be chosen form the layer icon in the upright of the map. A Thunderforest_Landscape map and an OpenStreet black and white Map is inserted here. Also, users could select to make the icons visible or not from the layer icon. Figure6 shows the overview map interface.

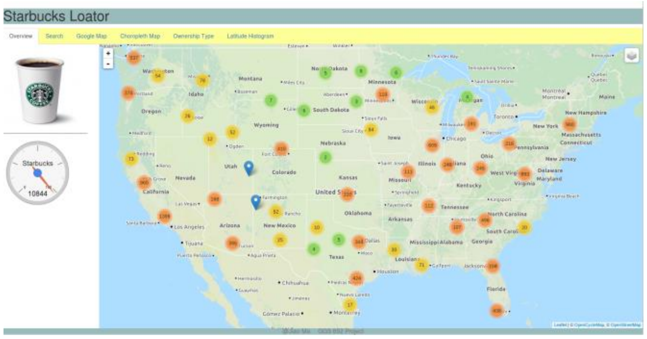

Figure6: overview layer

4.2 Search Map

This layer allow users to search specific contents from the database. Users could not only search location according to the address of each Starbucks, but also search service they want from this query bar. Result is also present in cluster method. Default icon is replaced by customer made icon, which is the small Starbucks icon. When clicking the icon, information about this Starbucks will pop out. For experiment, we type WiFi in the query bar to find all Starbucks have this WiFi service. Figure7 shows the result for searching WiFi.

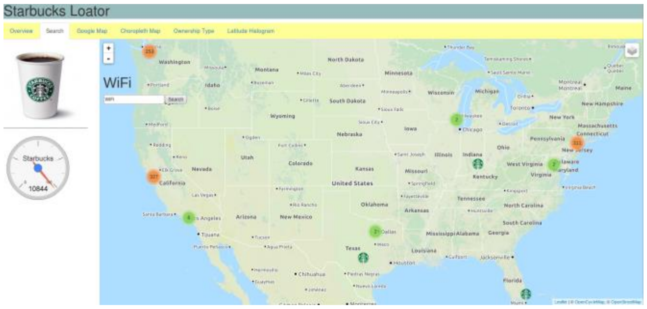

Figure7: result for searching WiFi

4.3 Google Map

Google Map function is not complete yet. Since Leaflet layer kind of on the bottom of the Google Map layer. But considering the power tool in Google Maps, street view methods, this error will be debug in future work [2]. Figure8 shows the Google Map interface.

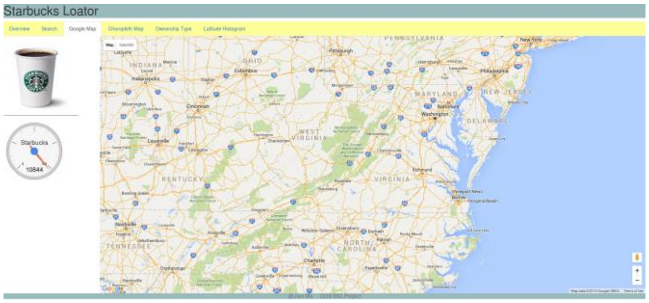

Figure8: Google Map layer

4.4 Choropleth Map

The choropleth map used here is borrowed from Leaflet, which is the choropleth map of population density. Here at first I plan to compare the cluster map and the choropleth map in the same file. However, things seem complex and more efforts need to be made to achieve this layer. By now the cluster map showed here is just a screenshot. As we see that region with higher population density may also have more Starbucks. Region with lower population density may also have less Starbucks. More statistical methods need to be applied here to detect the uncover facts.

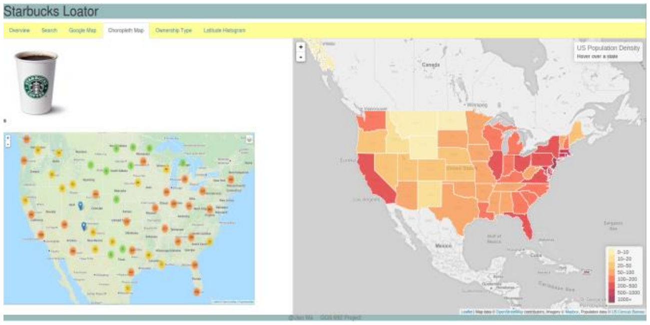

Figure9: Comparison cluster map with choropleth map

4.5 Statistics

Some basic statistic are received from Google Sheet. Below is the company owned Starbucks compared with licensed Starbucks. I personally think the company owned Starbucks may have more good taste. If someone is insistent on company owned Starbucks, user could use the query bar mentioned before to filter out the company owned Starbucks. Figure10 shows the percentage of company owned Starbucks and licensed Starbuck.

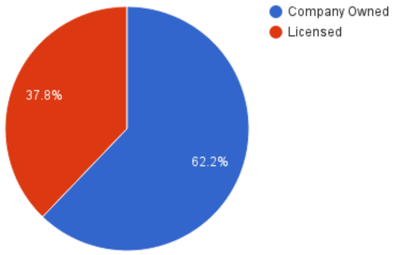

Figure10: the percentage of company owned Starbucks and licensed Starbuck

Another interesting thing I find is the histogram for latitude. As we see that the tendency of the number of Starbucks increases from latitude 19N to 43N, then the tendency goes down. Is there some relation between latitude and the Starbucks quantity? More further statistic research need to be applied. Figure11 shows the variety of Starbucks quantity according to the latitude.

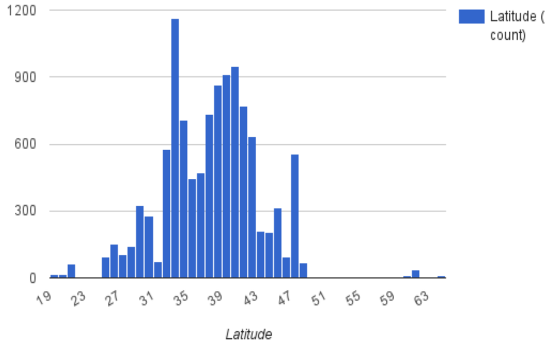

Figure11: the variety of Starbucks quantity according to the latitude

5. Future Work

Geospatial web application is widely popular nowadays. For future work, the errors existed still need to be eliminated. The Leaflet layers load correctly up Google Maps, as well as the drop down navigation bar to be added to contain all statistic graphs. Another problem is to solve the conflict and show two web maps in one HTML file. Besides, there are many functionalities could be added to the Starbucks Locator application.

6. REFERENCES

[1] Boundless,. "Education Center - Boundless". N.p., 2015. Web. 16 Dec. 2015.

[2] "Https://Developers.Google.Com/Maps/Documentation/Javascript/". N.p., 2015. Print.

[3] Mark Otto, and Bootstrap contributors. "Components · Bootstrap". *Getbootstrap.com*. N.p., 2015. Web. 16 Dec. 2015.

[4] OpenData by Socrata,. "Starbucks | Socrata". N.p., 2015. Web. 16 Dec. 2015.

[5] "Https://Www.Mapbox.Com/Mapbox.Js/Example/V1.0.0/Overzoom/". N.p., 2015. Print.

[6] W3schools.com,. "HTML Paragraphs". N.p., 2015. Web. 16 Dec. 2015.