TaxiViz - Geovisualisation of Taxi Cars

Jonathan Bosson

Abstract— TaxiViz is a web application that with a heatmap visualises where taxi cars were hired around the Stockholm area over a set time during March 2013. The dataset contains recordings of 1787 unique cabs and stores location, time and status data every minute for the month. The user can cluster the shown data with the data mining algorithm OPTICS in order to gain improved insight of where a taxi could park to increase the chance of getting hired.

Index Terms—cluster, OPTICS.

1 Introduction

It can be difficult to know where to go in a big city to quickly find a taxi. At the same time is it interesting for a taxi company to know where most people pick up a cab in order to streamline the drivers work shifts. TaxiViz visualises what streets in Stockholm are most popular during a chosen time period in March 2013. Through further exploration can trends during a specific time period be extracted, for example during the weekend. This information enables the user see the streets of which the most taxi cars are hired and make a wiser decision on where to go from there.

2 Background and Related Work

Numerous different application that visualise taxi data have been done before. Most common is to visualise the traffic and path different cabs take in big cities. In such applications will only a minute of data be visible at the same time which was not what has been done in TaxiViz. TaxiViz can show where taxi cars were hired over multiple days which in turn gives the user a larger overview of the data and better understanding of trends.

3 Data

The dataset that TaxiViz uses was recorded during March 2013 with 1787 unique taxi cars around Stockholm. Each minute the driver was working was the GPS location, time and hired status stored. This leads to a 5-dimensional dataset with id, time, latitude, longitude and hired status. The taxi drivers worked at different times but had an average work shift of six hours, which lead to the data containing roughly 13 million data points or 860 MB.

Since TaxiViz only visualises GPS locations where a taxi car were hired, were only the data points where a unique car's hired status changed meaningful. Furthermore could the hired status-dimension be taken away as it had no effect. The result was a 41.6 MB big dataset with four dimensions and roughly 670 000 data points spanning over all of March 2013.

4 Method

To sort out only the interesting, and uncorrupt, data in the original dataset was a python script written. The python script sorted the dataset by first ID and then time, saved only the first data point where a unique ID changed status to being hired.

The map of the area was taken from Google Maps API. A possible route to take was to create a topojson map and use D3.js to draw the data. However, since street names were relevant and the Google Maps API also provides efficient functions for drawing heatmaps, was the decision set to use Google Maps. Using a heatmap to show the most popular streets was a natural decision since no singular data points were important in the visualisation. A difficulty with a heatmap is deciding the maximum intensity. Using a dynamic maximum intensity set to the highest number of closeby data points led to a poor representation of less popular streets. Instead was a manual maximum intensity set, depending on the length of the chosen visualised time period.

By setting a manual maximum intensity will the most popular locations be trunctated and show up just like a location with high popularity. To still be able to visualise hotspots was a clustering function implemented with the data mining algorithm OPTICS [1, 2]. The algorithm takes two inputs; radius of the cluster as well as the minimum number of points inside the radius to create a cluster. The algorithm will process each data point and create a cluster around the point if it has enough neighbours. During run time could a better cluster appear which will then reassign the data points to a higher quality cluster. The result is that all data points that do not fulfill the criteria of minimum points inside the radius will be deleted.

To be able to only visualise a set of few days was a slider implemented. This allowed some further interaction and let the user see how popular streets vary depending on the chosen days. The slider was developed with D3.js.

5 Implementation

TaxiViz was developed as a web application using mostly Javascript and Google Maps API to make it accessible to any type of device. Low-end laptops were used in development with simple text editors for coding since it was what was most available to the developers. Since the data mining is computationally heavy did the used laptops take a considerably longer time to use the clustering function. To limit the waiting time was a maximum threshold of input data points set to 10 000.

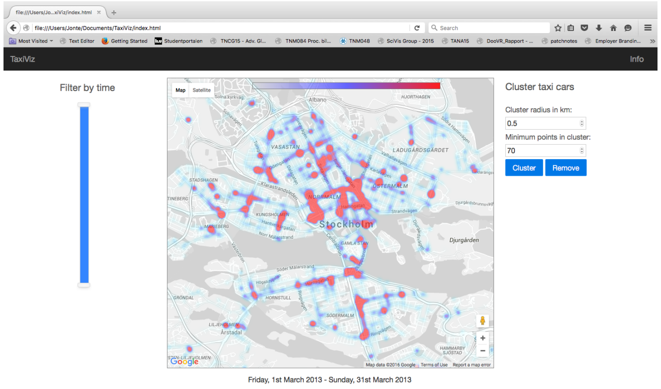
The application only works with the Firefox web browser since TaxiViz was developed during only three weeks by three part-time developers.

6 Results

The end product is a web visualisation tool to see where taxi cars are hired. The user can filter the time and move or zoom the map to desired view. The initial loading time of the application is long since it is opening a large data file directly on the web browser. Once loaded is the interaction fluid and responsive. The filter by time works smoothly on small and large ranges since it updates first when the user releases the slider. However, with a large range can the application become unresponsive for a few seconds since the heatmap has to reload with the new selected data.

Figure 3 displays the clustering functions with the red circles on the map. The circle size represent the number of cabs in the cluster. A problem with the circle size appears once the user zooms in or out on the map. The circles will then stay in the same size resulting in covering a larger area.

- *Jonathan Bosson is a student at Linköping University, Sweden, e-mail: jonbo665@student.liu.se.*



Filter by time

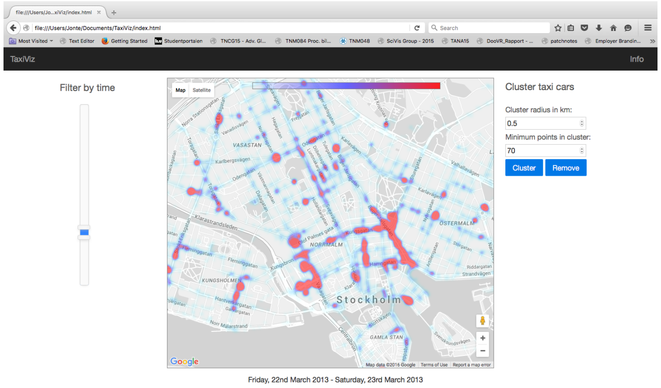
Cluster taxi cars

Cluster radius in km: 0.5

Minimum points in cluster: 70

Friday, 1st March 2013 - Sunday, 31st March 2013

Fig. 1. Heatmap visualising hotspots for where taxi cars are hired



Filter by time

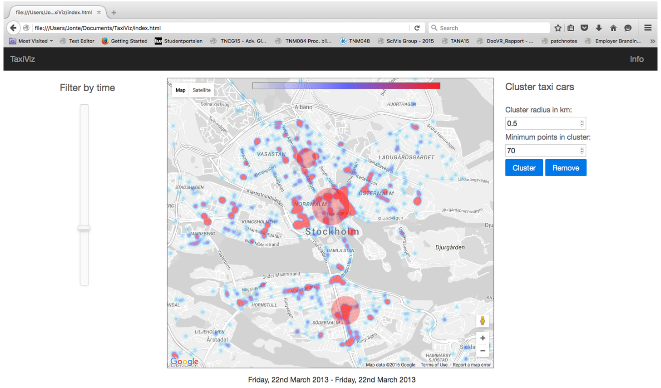
Cluster taxi cars

Cluster radius in km: 0.5

Minimum points in cluster: 70

Friday, 22nd March 2013 - Saturday, 23rd March 2013

Fig. 2. Small time preiod visualised to see where cars are hired during the weekend



Filter by time

Cluster taxi cars

Cluster radius in km: 0.5

Minimum points in cluster: 70

Friday, 22nd March 2013 - Friday, 22nd March 2013

Fig. 3. Clusters show where the highest density of taxi cars are hired

7 Evaluation

A user-centered evaluation was done with the focus of intuitivity. The test was done with three different people by giving the user a set of tasks to complete while the tester spoke out loud of what he was thinking or expecting. After the task had been completed was the tester asked a few additional questions about suggestions of how to improve the application.

The first task was to filter the data to only show half of March. All testers completed this task without any issue but one commented that it would be preferable to display exact time rather than just the selected dates. The second task was to cluster a selected time period. This caused some confusion since the application requested a very small selected time period, up to the point of where the users felt it was difficult to choose a small enough range. The error message that comes up when the user tries to cluster a too large data set was also ambiguous since it did not specifically say it needed a smaller time input. The clustering function was took a long time to complete. The testers understood the size of the circles represented the number of taxi cars in that area, but else than that did they not grasp what the result described. It was said it was difficult to see small clusters as they disappeared in the heatmap that was under it. Another request was to implement more information about the cluster once the user hovered over the circles.

The testers saw it as favourable to have the map in focus and appreciated the minimalistic graphical look. One made a suggestion to move the slider to be horizontal in order to give more space to enlarge the map even more.

8 Conclusions and Future Work

The graphical interface is well made and there is little to no confusion of how to interact with the application. The filter time could show what hours are selected to more accurately filter the desired time. The application runs smooth on low-end computers apart from a few seconds when a large time period is selected and when clustering. This could have been done more efficiently by adding the data into a database and letting the filter time request the required data. Further data mining of similar points could have compressed the dataset furthermore making the load time on large time periods much faster.

The clustering function could have been written in a more effective way, lowering the loading time and allowing for more datapoints to be considered. However, the OPTICS algorithm does not scale well with large datasets so the overall limitation would still exist. The drawn circles from the clustering function does not represent the area they cover well, specially if the user zooms in or out, which could be solved by determining the spatial radius of the cluster rather than the pixel radius.

References

[1] M. Lin and W.-J. Hsu. Mining gps data for mobility patterns: A survey. *Pervasive and Mobile Computing*, pages 1–16, 2013.

[2] B. Thierry, B. Chaix, and Y. Kestens. Detecting activity locations from raw gps data: a novel kernel-based algorithm. *International Journal of Health Geographics*, pages 1–10, 2013.