Michael Gerakis
Patrick Gomes
Raphael Baysa

CS 591 L1 - Data Mechanics

Recommended Locations for New Hospitals

Abstract

As overall population increases, space efficiency in highly dense areas becomes a more prevalent issue. As a result, buildings that may have been located in a good spot before are now in poor locations. Our focus is on hospital locations, but the general ideas used here could apply to any building by changing the factors involved. We define two sets of points: a set of points the optimal hospital should be close to, and a set of points it should be far from. Using these two sets we determine how good a specific hospital location is and using our algorithm, determine where is the next optimal hospital placing or if Boston even needs another hospital.

Introduction

Major cities in the United States are suffering from a space deficit. As the population in the world increases, our city planners need to become conscious of this and maximize efficiency in these cities. Boston is on the the cities suffering from this problem. Land spaces in boston are increasing as the city runs out of the space it has to offer. While we can't remove the essential buildings, such as police stations, fire departments, or schools, we can determine how to better place them to minimize the number needed. The goal of our research was to determine how well a generic optimization problem could solve this issue for hospitals specifically.

The optimization problem we designed was simple, minimize distance to a set of close points but maximize distance to a set of far points, where the points pairs of latitudes and longitudes. The hospital should be easily accessible to people without means of transportation, and close to crime but not surrounded by it. The hospital should be far away from police stations to avoid two emergency vehicles in one area, and high traffic areas to avoid delays in returning to the hospital.

Given the two sets of points it is easy to come up with a raw score for a hospital to compare with other locations. To analyze how accurate our results were, we compared the ranking of our raw scores versus the ranking determined by the hospital's star rating. Points can also be temporarily added and a raw score calculated to determine how that location would do.

Data Sets

Hospitals: https://data.cityofboston.gov/Public-Health/Hospital-Locations/46f7-2snz

MBTA Stops: http://datamechanics.io/data/pgomes94_raph737/stops.csv (Given by developer@mbta.com)

Traffic: https://data.cityofboston.gov/dataset/Waze-Point-Data/b38s-xmkq and https://data.cityofboston.gov/Transportation/Waze-Jam-Data/yqgx-2ktq

Crime: https://data.cityofboston.gov/Public-Safety/Crime-Incident-Reports-August-2015-To-Date-Source-/fqn4-4qap

Police Stations: https://data.cityofboston.gov/resource/pyxn-r3i2.json

Hospital Rankings: Individual requests to maps.googleapis.com

Data Transformations

The number of data points were on the order of thousands to tens of thousands, rather than compare to all the individual points we decided to create clusters to compare to. We combined the 4 datasets into a single dataset with proximity labels, close or far, as described before. Then we used k-means on each label to create these cluster centers. After trial and error on the number of clusters, we ended up with 22 close clusters and 23 far clusters.

We used the k-means algorithm to cluster the 4 sets of data. We created two groups of clusters, one for locations that we wanted to keep far away from the hospital, and the other for locations that we wanted to keep near. Each cluster point was labeled ‘F’ ‘C’ to determine if the point should be considered in the far cluster or close cluster. Specifically, we wanted to have a hospital near police stations and transit stops for safeness and accessibility. We also wanted the hospital far from crime spots, for safeness once again, and far from traffic spots, so that ambulances have an easier time moving towards the hospital. After some trial and error, we found that for the close data we should use 22 clusters and for the far data we should use 23 clusters.

Once we had these clusters, we used PySpark to run a map-reduce job to determine a score of all current hospitals in Boston. Our score was made by taking the closest point to each hospital from both clusters, based on euclidean distance, and subtracting those two distances to come up with the raw score.

Finally, we created recommendations on possible locations for a new hospital. To do this, we solve for the top 5 coordinates that minimizes the Euclidean distance to all

the close proximity clusters calculated in the previous transformations. Ideally Manhattan distance would be used, but it would require remapping the data points and roads into a grid like shape, which could be a future improvement. Instead we used the standard euclidean distance which is the straight line shortest distance between two points. From the returned points, we choose the one that maximizes the distance to all the far proximity clusters. This is an optimization problem defined as:

$$arg\ max\ x\ =\ \forall x \in top\ 5\ coordinates \sum_{y \in clusters\ centers\ with\ proximity\ 'F'} d(x, y)$$

where *d* is the Euclidean distance.

Top Hospital Ratings based on Google (ratings out of 5)

Massachusetts General Hospital	4.3
Beth Israel Deaconess Medical Center West Campus	4.1
Boston Medical Center	4
Franciscan Children's Hospital	3.9
New England Medical Center	3.7
Beth Israel Deaconess Medical Center East Campus	3.6
Kindred Hospital	3.4

Top Hospital Raw Scores generated by our algorithm:

Carney Hospital	0.004737505267096494
Kindred Hospital	0.002084855056278357
Boston Medical Center	0.0005538733174830423
Franciscan Children's Hospital	-0.004174204470076006
New England Medical Center	-0.006835661400495226
St Elizabeth's Hospital	-0.009744925356388302
Massachusetts Eye & Ear Infirmary	-0.012101165898738787

Data Analysis

From the data, most existing hospitals had a negative raw score; they were located closer to a cluster we expected them to want to be far from rather than a cluster we expected them to be close to. The abnormally high score of Carney Hospital can be attributed to it being on the edge of Boston. More data analysis would need to be put into determining if the issue lies in the number of clusters, the factors in each cluster, or how the actual values are calculated. For example, adding in more factors to the cluster sets might or by giving different weights to the different factors may fix this issue.

Despite this, when we compared how our scores did against real world results the results were not as far off as we expected. We extracted star ratings based on Google results and then ranked the hospitals based on the Google results, and created a separate ranking based off of our scores. We found that on average, our rankings were off by ~7 order placements, over a list of 26 hospitals.

Our hospital recommendation suffered because the latitude and longitude was not bounded to the abnormal Boston polygon shape. The algorithm suggests points that lie outside of the Boston area since it is farther from the far clusters, which slightly outnumber the close clusters. Bounding the points into the Boston polygon and then looking more into the calculations to combine the two steps in the recommendation into one would work better.

Extensions

Moving forward with this project, more research should be done into picking more factors. Two other factors we would have liked to include were the hospital locations and population dense areas. Ideally hospitals won't be close to one another but nearby areas with dense groups of people. We weren't able to find data on the population density of Boston that we could easily use, and we wanted to keep the clusters with the same number of factors in each so we chose to exclude the hospital locations.

The inspiration behind the recommendation for locations came from the k-means algorithm, with the change that you want to be far from some points. Once the bounding of possible points is established, rather than breaking the problem into 2 parts a better solution would be to write an algorithm similar to k-means that takes into account our label system. It would be more complex but should provide more accurate results.

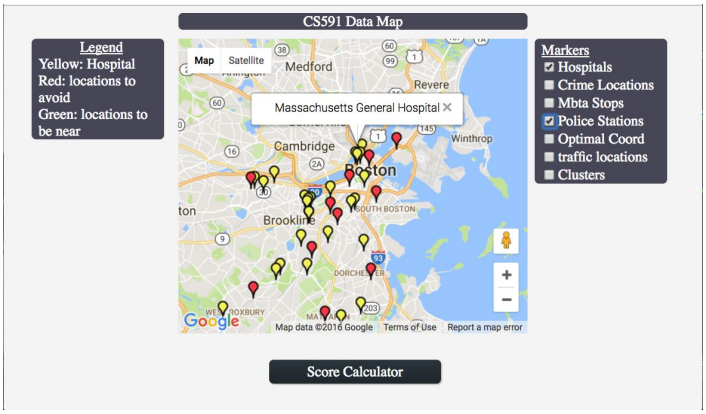
Lastly, a solver could be implemented that removes the worst located hospitals from the list and attempts to replace them with hospitals in better areas. A different approach would be given both sets, instead of replacing hospitals having the algorithm select the locations for the top hospitals. Either way, this would require establishing a minimum number of hospitals needed in the Boston area. The first approach is more reasonable practically since it keeps some of the hospitals and doesn't imply a full layout change.

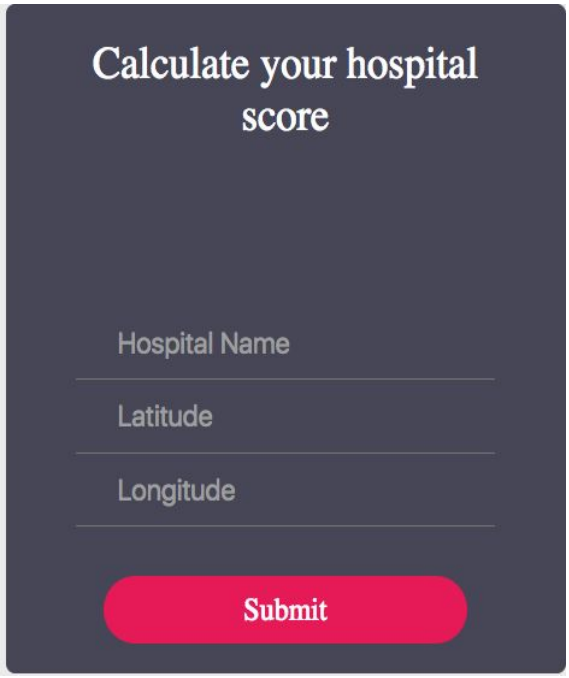
Conclusions

For a basic implementation we feel that this produced results that are worth looking more into. With the improvements mentioned in the extensions section, this algorithm could be used to determine whether or not hospitals are placed well throughout Boston and extended to other essential city buildings as well.

Visualization(1)

We took our data and made a web application that made a visual representation of the current data in our mongo db. Information to see this application live can be found in the README. This application allows users to click on specific data points they want to see and have it display on the map. (Warning: Mbta stops will lag since there are so many data points)





Visualization(2)

For our second application/visualization we decided to allow users to input in their own hospital information and see how their location compares to the other hospitals. Once users submit their data and press submit, the application takes their inputs and runs pysparkMapReduceJob.py with it and calculates a raw score similar to the scores of hospitals above.