Predicting Air Quality

Royce Gee
CSE190
rgee@ucsd.edu
A09504010

ABSTRACT

In this paper, I'll be looking at air quality index data provided by CitiSense, a provider of small, portable sensors that allow for real-time pollution monitoring through mobile devices. I'll be reporting the results of my exploratory analysis on the data and discussing the predictive task I have chosen. I'll then explain the success metrics for my generated model and go over related literature. Finally, I'll discuss the results gathered and how well they compared to established baselines.

1. EXPLORATORY ANALYSIS

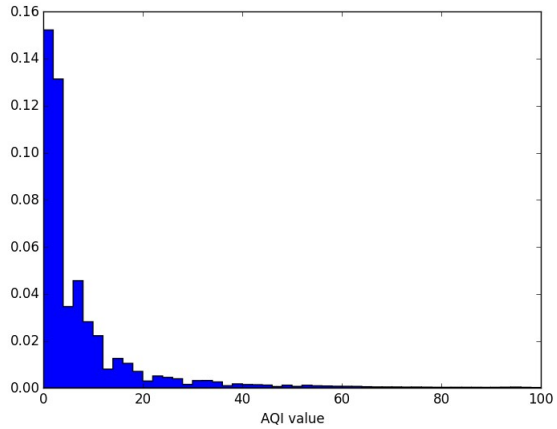
The data-set I'll be using is pollution data gathered by sensors attached to smart-phones. The file initially had 16 million data points with a variety of data types-- temperature, humidity, CO, AQI, etc. For the scope of this assignment, I reduced the data set to 673,161 data points measuring the calculated air quality index value at a given latitude, longitude, and time-stamp.

I used the geopy Nominatum functionality in order to get a sense of where most of the data is. The vast majority of the data was gathered in the San Diego region, but some of the latitude/longitude coordinates corresponded to areas like San Francisco, Seattle, and France.

The data set has a :

mean AQI value of 9.83, median of 3.0, and a standard deviation of 24.22.

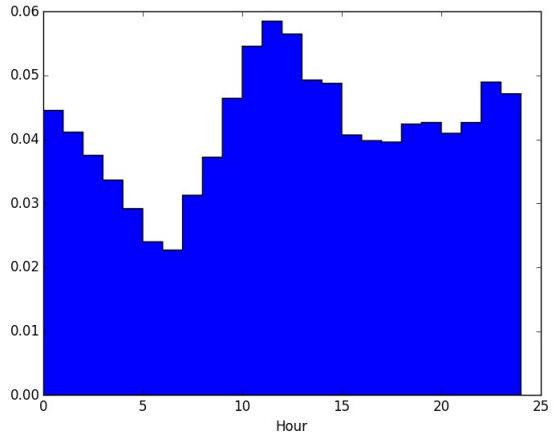
Figure 1. Histogram of AQI data in San Diego



AQI Values

From Figure 1, we get a good look at the spread of our data. For the most part, air quality is pretty good around the San Diego region because there are very few values approaching 50. From here, it is valuable to know when and where those high values are recorded.

Figure 2. Histogram of time-stamp hours

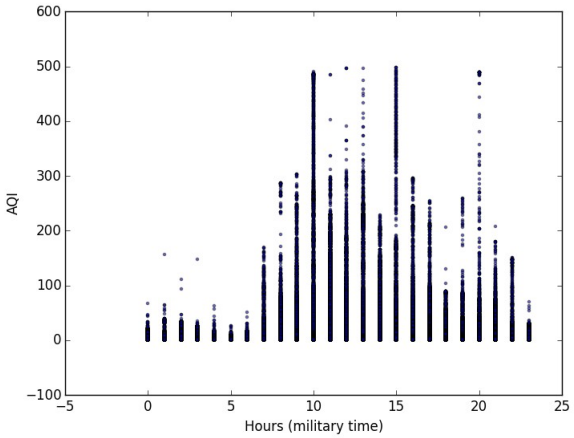


Timestamp hour in Military Time

For some reason most of the calculated AQI times occur during lunch hours, dip, and stay constant until 1AM in the morning when the recordings dip down a

lot, only to rise again at 6AM. My best guess is that these times correspond to how the software handles AQI calculations-- as users fall asleep and don't move around with their phone at around midnight the application becomes inactive. As users began to wake up around 6-7 in the morning activity picks up and becomes most active when users are commuting or walking around campus during the day.

Figure 3. Plot of Time against AQI



From the scatterplot, it seems that air quality is pretty good throughout the night and gets progressively worse starting around 7AM in the morning. There are two peaks at 10AM and 4PM, and some high values around 12 and 11 as well. A lot of this could be explained rather intuitively. When users are asleep at night, their phones are most likely indoors where the calculated AQI would be better and there is probably very little carbon monoxide emitted through vehicles able to travel to the bedroom. As users wake up and go outside to commute to work, carbon monoxide levels begin to rise, peaking at around 10AM which is pretty close to rush hour. From there, levels drop but there are high values recorded around lunch time as well from the lunch-time rush. Afterward, AQI levels peak again as employees leave work. What I didn't expect to see was high values recorded at 11PM. After looking into it, I discovered that this was because the data points were recorded in Europe, which had peak traffic hours at 9AM in their time while the sensor maintained a consistent timestamping methodology.

2. PREDICTIVE TASK

Motivation

My initiation motivation with this data was to create an accurate predictive model for when and where air quality worsens and gets very severe. The histogram shows that for the most part, the reading at any given time and place in San Diego will be less than 10 on the AQI scale, which signifies very good air. Knowing that there wasn't too much data that had very high AQI values, I pivoted my focus towards being able to predict when and where the air quality is *moderately* poor. My hope is that with the data given, I can create a model that with reliably predict which areas have poorer air quality and when that happens so that users with sensitivities can avoid these areas and the symptoms associated with being in them. I haven't talked to CitiSense but I think they would deem the predictive model a very useful application for people with air-related health issues.

Model selection based on exploratory analysis

When selecting the model, I had to first consider what I was trying to predict. Knowing that ideally the prediction would be a value, a binary classifier would be unsuitable. Least-squares and logistic regression would be able to provide a numeric output, but would not work given the nature of the features. Given that the histogram was bimodal and that I was dealing with geolocation data, linear regression wouldn't have churned out the best model because there wasn't a very strong linear correlation between time of day and AQI. Also, geolocation data is clearly not something meant for a linear model-- points on or near the highway may vary vastly in lat/lon combinations but the most important part is that the coordinate lies on the highway. I thought that maybe SVM would be an appropriate model, but was worried that a good hyperplane didn't exist for the dataset.

In the end, I decided with a k-nearest neighbor regression model in the hopes that the readings at a given time and place will be relatively constant.

Feature Selection

For my features, I'll be testing three k-nearest neighbor models. One will use only the timestamp. Another will use strictly the spatial data. Another will use the spatial data along with the time-stamp.

Training/Testing

I will do a 70/30 split between my training set and my test set. From my dataset, I'll randomly generate a number between 0 and the length of my dataset to remove a data point until I get a test set that is 30% of the total data.

Baseline Comparison

The predictions from my test set will be compared to three baseline predictors by looking at the sum-of-squared error obtained from each model. The three baselines are as follows:

1. A model that always predicts the mean of the training set
2. A model that always predicts the median of the training set
3. A model that randomly selects an AQI value between the minimum value seen in the training set and the maximum value seen in the training set.

Pre-processing

My data arrived in a flat file format, so some basic data processing was needed. The processing I did is as follows:

1. Parse through the dataset and filter out those that weren't pertinent to our data-- specifically, I removed data that didn't measure a calculated AQI reading.
2. Convert latitudes and longitudes string values to floats
3. Convert the datetime strings to minutes past midnight integer value.
4. Split these values up into two different 2D arrays. The first will have one dimensional vectors corresponding to the minutes past midnight a recording was made. The second will have the minutes past midnight, latitude, and longitude coordinates.

3. RELATED LITERATURE

The dataset that I'm using wasn't gathered very recently, but not much literature has been done on it. Instead, I looked at other similar datasets trying to tackle the same problem of predicting worsening air quality. The literature re-iterated and expanded upon my woes with a linear classifier. According to *Machine Learning Algorithms for GeoSpatial Data. Applications and Software Tools*, there are some "typical characteristics of geospatial phenomena and environmental data" that prevent them from being used in a lot of machine learning models. First is non-linearity and the inability to fit a line on the data. In addition, the paper argues against simple models like the one I'm using. Instead, it argues that good data needed for an accurate algorithm include slope and curvature.

The paper asserts that variograms, which describe "the degree of spatial dependence of a spatial random field", aren't useful for describing and predicting the spatial and temporal relationships between the inputs and measured AQI when the dimensionality is too high. In such instances, machine learning methods that are able to handle high-dimensional data are more capable of providing an accurate prediction model. The two methods described in the 2008 paper include artificial neural networks and support vector machines, but more importantly they discuss the methodology required prior to plugging data into a machine learning algorithm.

The first step is to quantitatively analyze the dataset to remove biases in modelling distributions and decluster the data. These methods shed light on the biases created due to the nature of how the data was sampled. For instance, in my acquired dataset a lot of the data taken at night time were indoors at a stationary position, creating a strong bias towards low AQI levels at that space and time. Such a space may be next to a highway and all datapoints outside the house may indicate higher levels of pollution. Still, the fact that there's a higher concentration of recordings inside the house at night (eight hours of sleep?) bias the recordings heavily.

The paper ultimately recommends using a hybrid model, combining machine learning with more traditional geostatistical methods.

4. RESULTS

For my first baseline, I took the mean of the training set and got a value of 9.86. The median of the training set was 3.0, which was used as my second baseline. Next, I took the minimum and maximum of 1.0 and 498.0 to be used as parameters for my randomly selected value baseline. The results are as follows:

Method	SSE
Location/Time kNN	15671447
Time kNN	184271427
Location kNN	76906956
Mean baseline	114313675.84
Median baseline	123493291
Random baseline	15962179369

Based on the results from the data, we can conclude that kNN algorithm that utilized both location and time proved to be the best method out of all five methods. The time proved to be a much worse indicator than I had previously thought, performing worse than the mean baseline. Location kNN actually performed better than the mean, but performed worse than time which was a little bit against initial expectations. I believe the performance of the models can be explained after knowing more about the dataset. The Citisense dataset provided only followed six different users as they carried their smartphones around throughout their typical day. With this in mind, it wouldn't be too unreasonable to assume that a lot of datapoints on any given day are going to be similar-- an employee will commute to work at roughly the same time each day and commute back home at the same time each day. Patterns get established and they form clusters of datapoints that are used during kNN training. Time may have a very weak correlation with the data simply because of the way the data is skewed towards having very low AQI levels-- although they may spend 10-15 minutes on the highway each day for the most part they'll be inside a home where AQI levels are generally lower. This is clearly shown in the histogram-- although there are points that indicate worse air quality at certain peak hours, there is always a high density of points clustered around the 0 mark. Location may also have a weaker correlation-- although highways may have a worse air quality, such levels only exist when slower traffic creates the levels of carbon monoxide needed to create the pollution. This is why I believe the best model combined both features-- during training the knn would take into account both the time and location. So even though being on the highway isn't the most accurate predictor and the time being 5pm isn't necessarily the most accurate predictor, it may be the case that being on the highway at 5pm provides meaningful information needed to predict the AQI level at the same and place the next day. This train of thought leads to more features that I would like to use if given the time to further structure the feature vectors. The first idea is rather intensive-- for each of the data points I would like to calculate the distance between the coordinate of that data point and the coordinate of the data point recorded immediately prior. Using this, along with the timestamp, one can approximate how fast the person is traveling. Using a reverse geolocation calculator, I would label each data point indicating whether the person is on the highway or not. Taking these both into account, I would hope to create a better regression model. Another small change I would do is to determine, based on the timestamp, whether the date was a weekday. My hunch is that AQI levels are much higher on the weekdays during peak traffic hours than they would be at the same hour during the weekend.

In conclusion, the kNN regression model trained was much more accurate than any of the baselines established. It was especially important that both parameters were utilized, as performance of the model dropped significantly whenever one of the features was removed. Moving forward, it would be interesting to see more users and maybe get data from an area more polluted-- maybe some industrial areas would have insightful information because factory emissions vary throughout the day. The kNN model performed well with three features and it would be interesting to look further into how implementing other features within each vector would decrease the prediction error. Also, further analysis with regards to variograms would be useful to better understand and interpret the relationship between time of day, location, and overall air quality index values. Thank you!