Max Sutton

CS229 Autumn 2019 Final Project Report

## Introduction

Tropospheric ozone is an important pollutant and potent greenhouse gas ("Tropospheric Ozone"). Because of this, the EPA has regulations of how much tropospheric ozone can be in the air ("Ozone Designation"). If an area is designated nonattainment, it warrants expensive and intensive oversight to bring the area back into attainment levels. Predicting nonattainment would allow areas to take action before they are forced to, allowing them greater flexibility and time in how they meet attainment. This approach can easily be extended to other pollutants that are monitored. This is an application of Machine Learning. The input to my algorithm is {temperature, pressure, wind, month, location, NO2 levels, VOC levels}. I then use a neural network to output a predicted ozone level for that day.

## Literature Review

Prediction of tropospheric ozone has been a topic of interest for researchers for a significant time. For example, researchers used a neural network to predict tropospheric ozone based on meteorological conditions as early as 2002, in the paper "Assessment and prediction of tropospheric ozone concentration levels using artificial neural networks" (Abdul-Wahleb and Al-Alawi 2002). Abdul-Wahleb and Al-Alawi were quite thorough, using 3 different neural networks and analyzing the relative significance of different features. Limitations of this paper include that that the authors focused only on summertime, urban areas, rather than more broadly predicting ozone, and that they only considered neural networks, not any other kinds of models. "Prediction of tropospheric ozone concentrations by using the design system approach"

(Abdul-Wahleb and Abdo 2007) experiments with other features and establishes that a non-linear model is required, although they do not consider many meteorological factors such as pressure and wind. The paper “Developing a predictive tropospheric ozone model for Tabriz” (Khatbi et. al 2013) is interesting as it looks at many different models, including not just linear regression and neural networks but also gene expression programming and nonlinear local projection, although it is limited in scope. One can also see the trend of ozone prediction models becoming more specific in “Tropospheric Ozone Formation Estimation in Urban City, Bangi, Using Artificial Neural Network (ANN)” (Aziz et. al 2019) and in “Forecasting of Surface Ozone Concentration by Using Artificial Neural Networks in Rural and Urban Areas in Central Poland” (Pawlak and Jarosławski 2019).

### Dataset and Features

For my project, I used Google BigQuery’ public data interface. For my training and testing I used the “EPA Historical Air Quality” dataset. This dataset contains daily data on many pollutants and meteorological features from 1980 through 2017. Data is collected from 1025 unique sites throughout the United States. Each site is labeled with a unique numeric code. For my data, I aggregated daily information about temperature, pressure, wind, month, location, NO2 levels, VOC (volatile organic compounds) levels by day and location, so that each row represented measurements of each attribute for one site (location) for one day. I chose to look at NO2 and VOCs because NOx, VOCs, and sunlight  are precursors to ozone. I chose to look at meteorological factures such as pressure and temperature because they indicate the presence of sunlight. The labels for my data are the daily ozone measurements from EPA air quality monitors throughout the country.

My data went through some preprocessing before I used it to train my model. First I split my train/eval/test sets. I trained on 200,652 points from 1980-2016, I evaluated on 4,539 points from Jan-June 2017, and I tested on 4,328 points from July-Dec 2017. In a final step, I normalized my data so that all features fell between 0-1, so that my model would not be skewed by the scales of different attributes. I normalized based on my training data set, so as to not bias my evaluation or test data sets.

### Methods

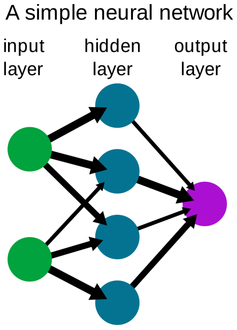
For my project, I focused on three kinds of algorithms: linear regression, support vector regression, and neural networks.

Linear regression is the simplest of the three models. It assumes that the labels are linear to the data attributes, that is, for some **Θ**, y=**Θ**T**x**. Starting with initial guesses for **Θ**, you iteratively change **Θ** in the direction that minimizes the loss, in my case ordinary least squares, defined as: $J(\theta) = (1/2)\sum(y_i - \hat{y}_i)^2$ , where $\hat{y}_i = \theta^T x_i$ .

Support vector regression is an extension of the classification model support vector machines. Support vector machines are used to classify nonlinear data into one of two categories. It does this by mapping data into a higher dimensional plane, where it can then be separated linearly just like linear regression. To make things easier, instead of explicitly mapping to a higher dimension, you can use a kernel function, or any function that can be written as the inner product of a higher dimensional feature vector, and minimize the loss with respect to the kernel. Support vector regression is an application of support vector machines. Its tactic for achieving regression based on a classification model is through the parameter ε. The two categories in support vector regression are: numbers within ε of the true label y, and numbers outside of it.

The kernel function used in my model is the polynomial function

$$K(x,x') = \gamma(< x,x' > +r)^d$$ .



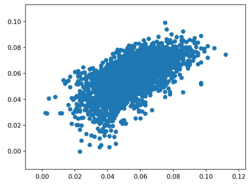
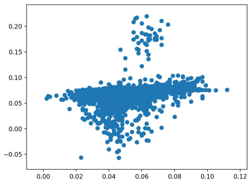
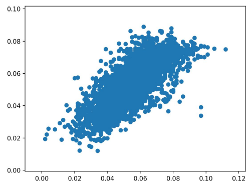
My final model that I used was a neural network. At a high level, neural networks are layers of simpler models that feed into each other. A simple schema is shown here to the right. The input to each new node is the result of feeding a linear transformation of the layer before (ie **ϴ**Tx) through an activation function. My neural network used the rectified linear unit function as the activation function, given as f(**ϴ**Tx) = max(0, **ϴ**Tx).

**Experiments/Results/Discussion**

To optimize my model, I iteratively changed the hyperparameters, similar to a manual gradient ascent. I optimized my models with respect to the R2 value, defined as:

$$r^2 = 1 - (\sum(y_i - \hat{y}_i)^2 / \sum y_i^2)$$

Ultimately, using this method of minimizing the above equation, I settled on a model with an adaptive learning rate and three hidden layers of 200 nodes each. My results are shown below.

| | Linear Regression | Support Vector Regression | Neural Network |
|---|---|---|---|
| R2 Value (Training set) | 0.4893 | -0.3193 | 0.7017 |
| R2 Value (Test set) | 0.4674 | -2.3626 | 0.6537 |
| Ground Truth Labels vs Predicted Labels on Evaluation Set |  |  |  |

It is interesting to see that although the graphs of predictions made by the linear model and neural network model look similar, they perform very differently. In addition, as I discuss in the next section, one of my SVR hyperparameters was not correctly tuned, and you can see this in the graph as it has started to form the same shape as the other two models, but not fully so.

## Conclusion/Future Work

Ultimately, the neural network that I trained performed the best out of all models, likely because it was more complex than the linear model, and because I realized post mortem that my ε hyperparameter for my SVR was much too high, thus leading to the SVR having little sensitivity to error. Even the neural network, however, was only able to predict with an R2 value of 0.7, even on the training set. This is to be expected, however, as there are features that I believe significantly influence ozone levels that I was unable to integrate into my data due to time limitations, such as presence of nearby factories or major highways.

If I had 6 more months to work on this project, there are a few extensions I would love to implement. I would love to extend the approach I used here to other pollutants that are monitored, such as NONOxNOy. I would like to investigate feature contribution, because it is much easier to reduce tropospheric ozone if you know what changes will have the largest effect. I would also like to add features that are preprocessing - intensive but valuable, such as US Census data that would indicate presence of major pollutant sources. Something I would also change is I would try making my model a classifier instead of a regressor, where each category would be a level of federal ozone classification, from marginal to extreme (“Ozone Designation”).

## References

Cite bigquery

Abdul-Wahab, Sabah A., and Jamil Abdo. "Prediction of tropospheric ozone concentrations by using the design system approach." Journal of Environmental Science and Health Part A 42.1 (2007): 19-26.

Abdul-Wahab, Sabah A., and Saleh M. Al-Alawi. "Assessment and prediction of tropospheric ozone concentration levels using artificial neural networks." Environmental Modelling & Software 17.3 (2002): 219-228.

Aziz, Abdul, Fatin Aqilah Binti, and Jarinah Mohd Ali. "Tropospheric Ozone Formation Estimation in Urban City, Bangi, Using Artificial Neural Network (ANN)." Computational Intelligence and Neuroscience 2019 (2019).

Gomez-Sanchis, Juan, et al. "Neural networks for analysing the relevance of input variables in the prediction of tropospheric ozone concentration." Atmospheric Environment 40.32 (2006): 6173-6180.

“Ground-Level Ozone Basics.” EPA, Environmental Protection Agency, 31 Oct. 2018, www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics.

“Introduction to BigQuery.” Google, Google, 4 Dec. 2019, cloud.google.com/bigquery/what-is-bigquery.

Khatibi, Rahman, et al. "Developing a predictive tropospheric ozone model for Tabriz." Atmospheric environment 68 (2013): 286-294.

“Ozone Designation and Classification Information.” EPA, Environmental

Protection Agency, 17 Sept. 2018, www.epa.gov/green-book/ozone-designation-and-classification-information.

Pawlak, Izabela, and Janusz Jarosławski. "Forecasting of Surface Ozone Concentration by Using Artificial Neural Networks in Rural and Urban Areas in Central Poland." Atmosphere 10.2 (2019): 52.

“Scikit-learn: Machine Learning in Python”, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

“Tropospheric Ozone.” Climate & Clean Air Coalition, UN Environment, www.ccacoalition.org/en/slcps/tropospheric-ozone.

Link to code:

https://drive.google.com/open?id=1EAsXtXXvS9CJSOXhPuP2i-vkbojz-Gll