Air Pollution Mapping and Prediction

Gaurav Mahamuni
Mechanical Engineering Department
University of Washington Seattle
Washington, WA 98105
gauravsm@uw.edu

Mingyu Wang
Mechanical Engineering Department
University of Washington Seattle
Washington, WA 98105
edward.n.518@gmail.com

Su Ye
Department of Computer Science
University of Washington Seattle
Washington, WA 98105
yes23@uw.edu

Abstract

Air pollution monitoring is done by analyzing particulate matter (PM) in air. PM analysis is important in assessing an individual's exposure to potentially harmful particles, such as aeroallergens, toxins, and emissions from combustion sources. Currently, PM is recorded at sparse locations in a geographical area, however, the PM level can vary dramatically over small distances. The sparsity of air quality measurement sensors makes assessing PM at specific locations quite difficult. In this report, we evaluated the accuracy of assessing PM levels at specific locations in the city of Krakow in Poland from spatio-temporal data of PM levels by applying different models. We apply three approaches for mapping and prediction: 1. Using Bellkor recommendation system, and achieved an overall $R^2 = 0.928$ for measuring the PM level trend 2. PM levels are mapped and predicted onto 1km X 1km grid points over the city using semi-supervised classification. L_1-regularized logistic regression along with expectation maximization results in 69.5% accuracy for mapping and 61.5% - 52% accuracy for prediction from 1 hr - 4 hrs respectively. 3. A measurement of the PM level trend using radial basis function interpolation achieved an overall $R^2 = 0.873$.

1 Introduction

The air quality has a particularly close connection with human health. Exposure to indoor and outdoor air pollution was estimated by World Health Organization (WHO) to cause about 7 million deaths in 2012. Therefore, there exists a need for providing information on air quality at a specific location. However, solving this problem can be difficult because the air quality can change dramatically in urban areas and the number of air quality monitoring stations is limited. Since there are insufficient air-quality-monitoring stations in a city it is expensive to obtain labeled data. The labeled data is often incomplete and there is not a universally accepted mechanism to suggest the main causes of the occurrence and dissipation of air pollution.

One solution for this is to build a model based on the data collected from current existing monitor stations and using it to model and to predict air quality for a specific location with a certain spatial resolution. In this paper, we are focusing on mapping as well as predicting the concentration of PM2.5. PM2.5 (particles less than 2.5 micrometers in diameter) analysis is used in air quality measurement since exposure to PM2.5 is associated with respiration and cardiovascular illness. Our goal is to map and predict concentration of PM2.5 at any location over considerable time in the future the so that

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

people can have sufficient time to prepare for imminent pollution exposure risks such as exposure due to forest fires.

2 Review of the relevant prior work

In the first approach, we setup and develop the model for mapping and prediction based on the following papers: 1. High-Resolution Air Pollution Mapping with Google Street View Cars: Exploiting Big Data, 2. High-resolution mapping of traffic-related air pollution with Google street view cars and incidence of cardiovascular events within neighborhoods in Oakland, CA.[2] In both the papers, the authors used the data collected by the street view car around the same region. The first paper demonstrates the approach for data collection and local pollution variation mapping. In the second paper, the authors emphasize the importance of having such a map by indicating a positive relationship between the variation of pollution in a local region and the incidence rate of cardiovascular events within the neighborhood. Even though it has been proved that the local air pollution level map is useful, the map constructed in the first paper was made by high-resolution data, which is hard to obtain. The papers provide us some ideas on how to investigate the local pollution mapping and prediction problem.

In the second approach, the geographical ares is split into 1km X 1km grids, and we map as well as predict the level of PM concentration for each grid point on the map. This idea is inspired by the strategy mentioned in Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality.[3] The authors of this paper developed an algorithm named Deep Air learning model which uses semi-supervised learning. Considering the topics of interpolation and prediction both as the classification problems with different outputs, they develop a neural network model with a general multiple-output classifier to combine and solve the two topics. The input layer uses feature analysis to select the main relevant features and the output layer uses spatio temporal semi-supervised learning. In this report, we implement a logistic regression model with L_1-regularization for feature selection and expectation maximization for semi-supervised learning to map and predict PM2.5 concentrations.

Model development for the third approach is based on the paper, Data-driven discovery of partial differential equations[4]. In this paper, the researcher proposed a sparse regression method that would be capable of discovering the governing partial differential equations of a given system by time series measurements in spatial domain. In the implementation of this model, we try to measure the spread of PM in spatial domain over a time range.

3 Data collection, processing and basic analysis

Our group plans to use the following existing dataset: `https://www.kaggle.com/datascienceairly/air-quality-data-from-extensive-network-of-sensors`.

This is the air quality data (the concentrations of particulate matter PM1, PM2.5 and PM10, temperature, air pressure and humidity) from 2017 generated by a network of 56 low-cost sensors located in Krakow, Poland. The data can be used for mapping and predictive modelling of air pollution. The data is recorded every hour for 12 months. The number following PM e.g. "10" in PM10, "2.5" in PM2.5, etc, are the particle sizes in microns of pollutants in air. The lesser the size, the more harmful the particles can be to the human health. Although this data has a lower spatial resolution, we develop models to map and predict PM concentrations in areas smaller than the spatial resolution of the sensors.

We have two kinds of data in the dataset for each sensor: 1) Meteorological data: Consisting of temperature, humidity and barometric pressure, 2) Air quality data: We collect real valued concentration of three kinds of air pollutants, consisting of PM2.5, PM10 and PM1. The PM1 data was not calibrated hence we do not use it in this study. PM2.5 are smaller particles which can go deeper into the lungs and can cause cardiovascular/respiratory diseases. PM10 are bigger particles which get filtered out in the nasal and throat region and may not be as harmful to human health as PM2.5. Hence, we mainly map and predict PM2.5 concentrations. The data in the dataset had to be preprocessed in order to apply machine learning algorithms. Only a few of the sensors recorded data over the entire 10 months in 2017.

For approach 2 and 3, we filtered out sensors that did not record data for over a month and also which did not have data for all the meteorological features - temperature, pressure and humidity and PM2.5 concentrations to obtain a network of 29 sensors. In approach 2, Some senors have missing data for a few hours which was fitted through interpolation. For approach 3, the time series data points with missing information were dropped and the remaining data was used for model development. Since only time series data was dropped from all the sensors, dropping entire 10 months of data from particular sensors was not necessary. Hence, this approach uses different subsets of data from all 56 sensors for each month. Sensors are split randomly to have 6 test sensors and remaining sensors are assigned as training sensors. The position of the 6 randomly chosen test sensors are shown in Figure 3. The time series data for training sensors is used for model development and the time series data for test sensors is used to determine accuracy of the model. All approaches use the same test sensors. Fig. 1 shows the average normalized PM2.5 concentration over 10 months in 2017 and the location of the 29 sensors. Fig. 2 shows time series data for 6 sensors. The PM2.5 data varies from 0 - 300 $\mu g/m^3$. The PM2.5 levels are higher in fall and winter months and lower in spring and summer months.

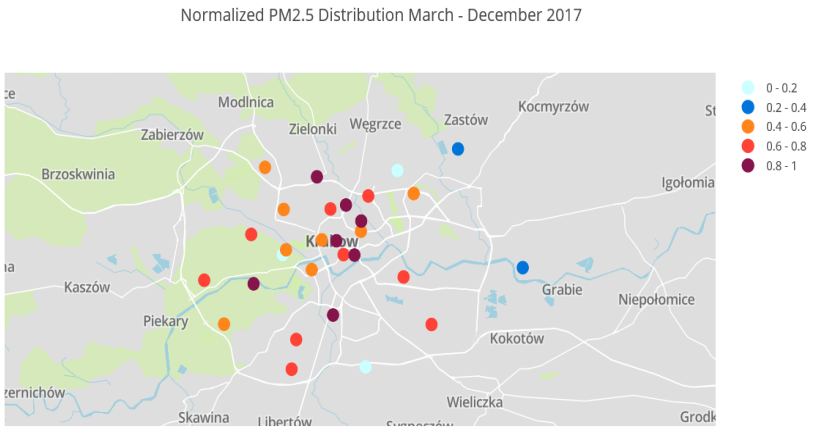

Figure 1: Overall distribution of the sensors and average normalized pollution at sensor location for 10 months in 2017.

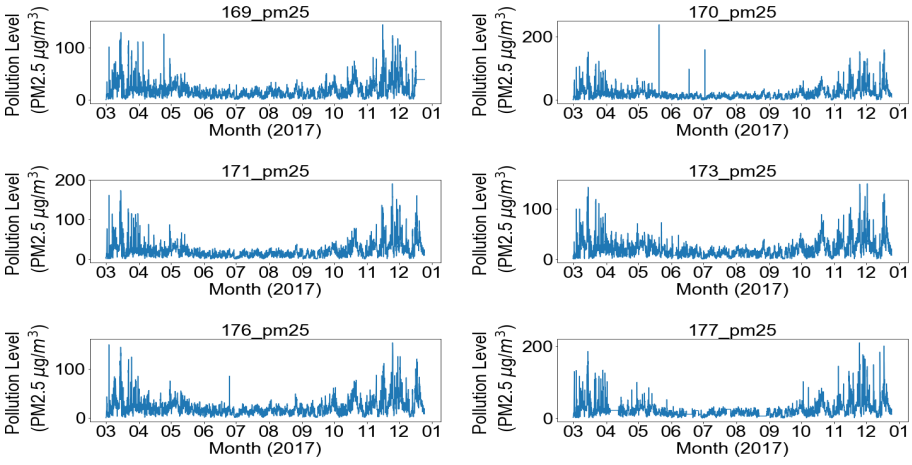

Figure 2: Pollution data over 10 months for 6 of the 29 sensors. The pollution levels are higher in the fall and winter months.

4 Methods/Algorithms/Models

Codes for all method implementations can be found here: `https://github.com/gauravsm31/CSE547-Air-Pollution-Mapping`

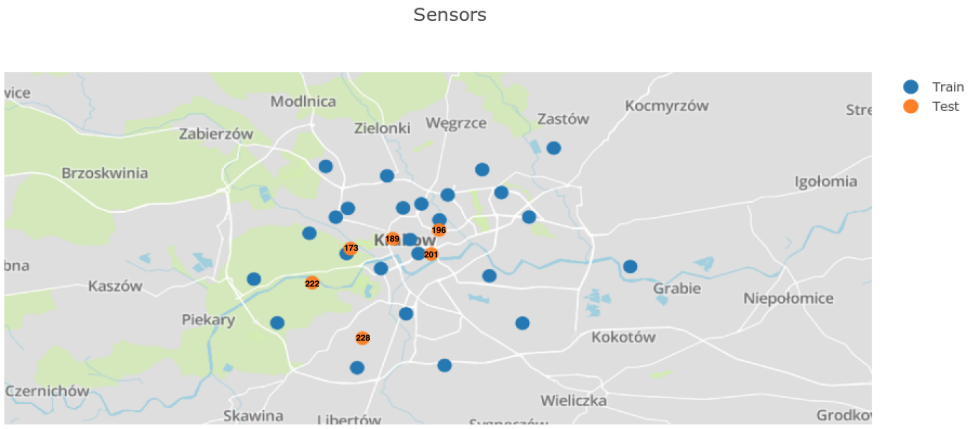

Figure 3: The relative position for test data with respective to all other training sensors

4.1 Bellkor recommendation system

In Bellkor recommendation system, it has users and ratings to some movies for each user. The algorithm would make a prediction for the missing movie ratings. In our problem, we have location data and pollution data for each location, and we want to mainly predict for the missing PM2.5 value. Therefore, we see the similarity between these two problems, and we plan to apply the Latent factor model. In this method, we will try to find some P, Q matrices for reconstructing R, the PM2.5 value chart for each location. The dimensions for Q should be $m \times k$ and the dimensions of P should be $n \times k$. R is a $m \times n$ matrix, where m is the number of features, and n is the number of locations. The columns in R would be composed of geographical data(temperature, humidity, pressure) and pollution data(PM2.5) at each time instance, and the rows are each sensors. There would never be a cold start problem in our setting since the geographical data is fixed and pre-computed. We made a strong assumption for this model, which is that we assumed that the temperature, humidity and pressure data values are given in each location. This assumption makes sense based on the fact that the geographical data we have does not vary too much within a region. Even if these geographical data is missing it would be easy to obtain. Then, when we would try to predict the PM2.5 value for a new location with certain geographical data using our constructed P, Q matrices.

We put geographical data together with pollution data and treat each of them as a "movie" in Bellkor recommendation system. Then we would run the Stochastic Gradient Descent algorithm to minimize the reconstruction error function:

$$E[R, P, Q] = \sum_{(i,l) \in records} (R_{il} - q_i * p_l)^2 + \lambda(\sum_{l=1}^{n} \|p_l\|_2^2 + \sum_{i=1}^{m} \|q_i\|_2^2) \quad (1)$$

Algorithm 1: Stochastic Gradient Descent Latent Factor Model

Inputs: Training dataset $D = D_l \cup D_{ul}$, where D_l consists sensors with geographical data and given value for PM2.5, D_{ul} contains sensors with only geographical data.
Initialization: Initialize P, Q matrix with initial value $\sqrt{100/k}$
for <*i = 1...number of iterations*> **do**
  for *each data point* $v_s t$ **do**
    $\epsilon_s t \leftarrow 2(v_{st} - q_t \cdot p_s)$
    $q_t \leftarrow q_t + \mu(\epsilon_{st} p_s - 2\lambda q_t)$
    $p_s \leftarrow p_s + \mu(\epsilon_{st} q_t - 2\lambda p_s)$
  end
  end

After we ran the algorithm we would expect to get two matrices, P, Q, where $p_s \cdot q_t$ should be a good estimation for the value of r_{st}. For obtain the PM2.5 value that is measured for each test sensors, we just need a dot product of the corresponding row in P and column in Q.

In order to better apply this algorithm, we would need to normalize the value for each feature, because the value for different features varies a lot. For example, pressure data would take range from 99596 to 103171, but PM2.5 data only takes a value between 0 and 351. We normalized the original data to a range between 0 and 1. However, in practice, we realized that this normalization would have minimized the difference between two data point a lot so that the differences between the data of two sensors would not be that obvious. After several trials, we decided to scale up this normalized feature values by 1000 so that the differences between values could be obvious. Therefore, the feature values that are been fed to the model would all having value between 0 and 1000. Since the PM2.5 data would generally be low(the maximum is still 351, and the majority data points would be around 100), we choose to initialize the P, Q matrix with the value of $\sqrt{100/k}$.

4.2 Semi-supervised classification using L_1-regularized Logistic Regression

Consider a binary classification problem. Assume that $Pr(Y = 1|X = x) = p(x; \theta)$, for some function p parameterized by θ. Assume that the observations are independent of each other. For classification, the model parameters can be estimated by maximizing the log likelihood function given by (2).

$$\hat{\theta} = \underset{\theta}{argmax}(log(\prod_{i=1}^{n} p(x_i; \theta)^{y_i}(1 - p(x_i; \theta)^{y_i}))) \tag{2}$$

Logistic regression is used in classification problems to provide a linear boundary between values in feature space. The class probabilities for logistic regression are given by (3).

$$p(x; b, w) = e^{(b+wx)}/(1 + e^{(b+wx)}) \tag{3}$$

Here θ is (b, w). θ can be estimated by maximizing the log likelihood function given by (2).

We developed a general and effective approach to unify the interpolation, prediction, feature selection and analysis of fine-grained air quality into one model. We divided the Krakow city geographical area into disjointed grids (1km X 1km in the experiment) assuming that each grid is an instance and the air quality in a grid is uniform. The instance is labeled if there is an air quality monitor station located in the grid; otherwise, the instance is unlabeled. In the model development experiments, 23 grids are labeled, and 217 grids are unlabeled. Assuming that the current time is t, the target is to map and predict the PM2.5 concentrations for time from $t - T_{\mathrm{f}}$ hours for both the labeled and unlabeled data. Hence, the number of the output nodes for the model is $T_{\mathrm{f}} + 1$. The PM2.5 concentrations are divided into six classes.

For a given instance i, the input vector $\mathrm{x_i}$ contains the following features for each gridpoint: 1) The meteorological data (temperature and humidity) for the grid of instance i, from time $t - T_{\mathrm{p1}}$ to time $t + T_{\mathrm{f}}$; 2) The air quality data of the N nearest air quality monitoring stations for the grid of instance i, from time tT_{p2} to time t. For unlabelled grid points, the meteorological data from the nearest training sensor is assigned as data for that grid point.

To determine accuracy, we use real data of the first 30 days out of the 10 months of data available as the model development becomes computationally expensive with increase in training samples. We set $T_{\mathrm{p1}} = 11$, $T_{\mathrm{p2}} = 23$ and $T_{\mathrm{f}} = 5$, thus the dimensionality of the input vector is $2(T_{\mathrm{f}} + T_{\mathrm{p1}} + 1) + N(T_{\mathrm{p2}} + 1)$, and the dimensionality of the output vector is $T_{\mathrm{f}} + 1$. This is the dimensionality for each grid and we have 240 grids.

We use sklearn from scikit library in Python to perform L_1-regularized logistic regression. The following algorithm is used to determine the logistic regression model parameters. Let D_l be the

labelled data and D_{ul} be the unlabelled data. We perform semi-supervised learning through maximum likelihood estimation (MLE).

Algorithm 2: Semi-supervised Logistic Regression

Inputs: Training dataset $D = D_l \cup D_{ul}$, where D_l consists of labeled samples and D_{ul} contains unlabeled samples

Initial Estimates: Build initial classifier (L_1-regularized Logistic Regression + MLE) from the labeled training samples, D_l. Estimate initial parameter θ using MLE.

while *log likelihood increases* **do**

E-step: Use current classifier to estimate the class membership of each unlabeled sample, that is, the class with maximum probability that the sample belongs to that particular class (see (3)).

M-step: Re-estimate the parameter, $\hat{\theta}$, given the estimated label of each unlabeled sample (see (2))

end

Output: An MLE classifier that takes the given sample (feature vector) and predicts a label.

4.3 PDE functional identification of nonlinear dynamics algorithm

In general, a dynamic system is described by a governing partial differential equation. Considering the variance of pm2.5 as a dynamic system, we look for generating a governing equation for it.The method is called Sequential Threshold Ridge regression. Using this method,We assume that the solution PDE takes a forms presented below,picture U_t is the time partial derivative. N is the summation of terms in the parentheses.

The first step, we need to numerically calculate the time derivative with respect to the concentration of pm2.5 for each point. calculating for a time point, we extract the five data before and after that time point, do the polynomial interpolation with degrees of 5 and use the central finite difference derivative method. We build a single column vector for these derivative values. We calculate all training sensors' derivative at the same time points and add into column vector.

$$\Theta(\mathrm{U},\mathrm{Q}) = \begin{bmatrix} 1 & \mathrm{U} & \mathrm{U}^2 & \dots & \mathrm{Q} & \dots & \mathrm{U}_x & \mathrm{U}\mathrm{U}_x & \dots & \mathrm{Q}^2\mathrm{U}^3\mathrm{U}_{xxx} \end{bmatrix}.$$

$$u_t = N(u, u_x, u_{xx}, \dots, x, t, \mu)$$

Secondly, we generate the possible candidate terms for N. To begin with, we need to build column vectors of the spatial partial derivative. To find this derivative, we need fix rest of the variables and left only one variable change. In the project,for example, we need to fix time and y-axis and collect the pm2.5 value vary along the x-axis around that target point. However, the sensors position are scatter, which no sensor has the same x or y value. Therefore, we use radial basis function interpolation from scipy library in python. The type of the interpolation function is inverse function. By using interpolation, we obtain the desire value and calculated the derivative by center finite difference method. The next step is is building the vector of other possible data,for example, the value of pm2.5, temperature and humidity.

The final step is building the candidate terms library,θ, by making the combination of the terms obtained above and calculating the corresponding value. The library form presents below, The general function form is below,

$$U_t = \theta\xi \tag{4}$$

The form of writing this function in matrix is below,picture1 ξ is the coefficient of each candidate terms in the library. It is calculated by the algorithm.

The algorithm derives from sequentially threshold least squares method. In STLS, a least squares predictor is obtained, and a hard threshold is performed on the regression coefficients. This process is repeated recursively on the remaining nonzero coefficients. The term to regularize the least squares

$$\begin{bmatrix} u_t(x_0,t_0) \\ u_t(x_1,t_0) \\ u_t(x_2,t_0) \\ \vdots \\ u_t(x_{n-1},t_m) \\ u_t(x_n,t_m) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & u(x_0,t_0) & u_x(x_0,t_0) & \dots & u^5u_{xxx}(x_0,t_0) \\ 1 & u(x_1,t_0) & u_x(x_1,t_0) & \dots & u^5u_{xxx}(x_1,t_0) \\ 1 & u(x_2,t_0) & u_x(x_2,t_0) & \dots & u^5u_{xxx}(x_2,t_0) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u(x_{n-1},t_m) & u_x(x_{n-1},t_m) & \dots & u^5u_{xxx}(x_{n-1},t_m) \\ 1 & u(x_n,t_m) & u_x(x_n,t_m) & \dots & u^5u_{xxx}(x_n,t_m) \end{bmatrix}}_{\text{Example } \Theta \text{ for real valued function in one spatial dimension}} \begin{bmatrix} \\ \\ \\ \\ \\ \\ \xi \\ \\ \\ \\ \\ \\ \end{bmatrix}.$$

Algorithm 3: STRidge($\boldsymbol{\Theta}, \mathbf{U}_t, \lambda, tol, \text{iters}$)

$\hat{\xi} = arg\,min_\xi \|\boldsymbol{\Theta}\xi - \mathbf{U}_t\|_2^2 + \lambda\|\xi\|_2^2$ # ridge regression
bigcoeffs $= \{j : |\hat{\xi}_j| \geq tol\}$ # select large coefficients
$\hat{\xi}[\sim \text{bigcoeffs}] = 0$ # apply hard threshold
$\hat{\xi}[\text{bigcoeffs}] = \text{STRidge}(\boldsymbol{\Theta}[:, \text{bigcoeffs}], \mathbf{U}_t, tol, \text{iters} - 1)$
recursive call with fewer coefficients
return $\hat{\xi}$

problem is used to help avoid problems due to correlations. The pseudo-code for the algorithm is in algorithm 3. The algorithms, matrix formulation and equations are taken from [4].

5 Results and findings

5.1 Bellkor recommendation system

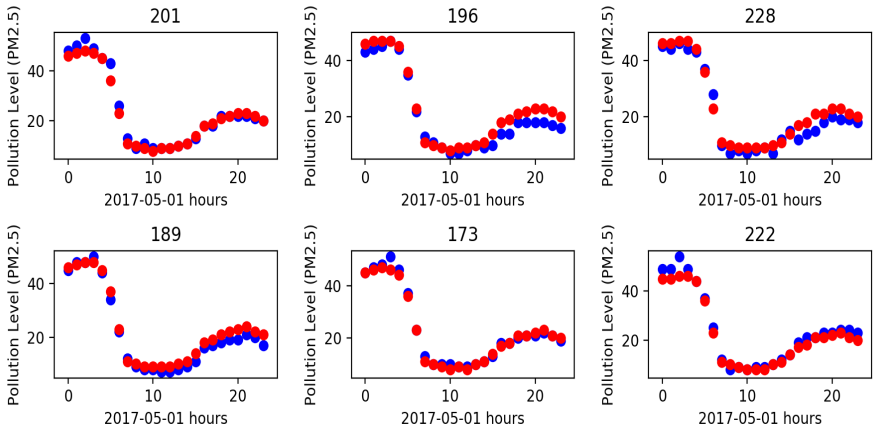

Figure 4: Pollution data of test sensors for 24 hrs on date 2017-05-01. The red dots are measurements result by latent factor model, the blue dots are the true records.

This model is limited to predict only the PM2.5 level of the current day. It would not be able to predict future data since the evaluation depends on the PM2.5 level measurement in other locations. However, it has done a good job for measuring the past trend by achieving a R^2 score of 0.928 on the test data over the past 12 month, with a tuned regularization parameter $\lambda = 0.1$, learning rate = 0.00001, $k = 20$, for 60 iterations. Figure 4 shows one instance trend measurement in May 1st, 2017. In general, the prediction was not successful in this model since it can only measure the past trend. The best current time(within one hour) prediction have an R^2 score of 0.484. This would potentially be caused by insufficient data features. In our data, we only have a temperature, humidity, pressure

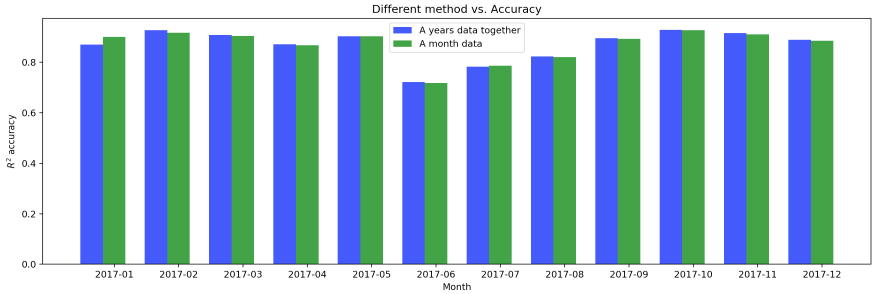

Figure 5: Comparison between including a year's data and a month's data

and coordinate. Among these features, only temperature was measured as the most effective influence on the level of PM2.5. The intuition behind choosing this model is that if there are two locations that have similar features, the PM2.5 level in those two locations should be similar. However, here in our data, we do not have sufficient sensor point and features related to each sensor point. The model would have done a better job if we have wind direction, precipitation, traffic, etc. data. One another reason might be related with the fundamental property of this model. This model was mainly used for recommendations instead of prediction. It would not assign relatively weighted to each potential feature, and it would be affected if there are more columns of data for a certain type of feature. Given these limited data and our model performance on the data we still have the following two findings.

- The R^2 score does not vary a lot if we include more data. We have tried to include all 12-month data or include only 1, 2, 3-month data, the R^2 score does not vary a lot or have no correlation with more data. In Figure 5, the blue bar is our measurement if we input a year's data in, and the green bar is our measurement if we input a month's data, as shown in the figure there is not much difference between the two bars. We have also tested with including 2, 3 months, and there is no major improvement in the R^2 score. This is also an indication that the features in the data are not sufficient for concluding meaningful result by using this model.
- Location with more sensor surrounded would get a better measurement. Table 1 shows the R^2 score for different test sensors, and we will see the highest one is 189, refer back to Figure 3, we would see that 189 is at the center place where there are many sensors around it. This does not imply that if we have a sensor nearby we would have a better prediction since there is a very near sensor around sensor 173, but 173's prediction is not the best.

Table 1: R^2 measurement for all test sensors

	189	201	173	196	222	228
R^2 scores	0.935	0.915	0.912	0.906	0.822	0.778

5.2 Semi-supervised classification using L_1-regularized Logistic Regression

We calculate the accuracy of prediction on the test data using 0/1 loss function i.e accuracy(%) is the percentage of labels correctly predicted. Table 2 shows the prediction accuracy for each hour from 0-4 hrs for test data. As the future time for prediction increases, the accuracy drops. Hence future time for prediction must be decided based on accuracy requirement. The training accuracy is only slightly higher than test accuracy and is not reported. This indicated that the model is not over-fitting. The regularization parameter in L_1-regularized logistic regression was tuned using only two values as the computational cost of tuning is expensive. Higher computational resources will help in tuning the regularization parameter and improve model accuracy.

Fig. 6 shows PM2.5 concentration labels for 7th March at 6:00 AM at the 29 sensor locations (training + test sensors). These concentration labels are compared to labels from mapping by the semi-supervised L_1-regularized logistic regression model at that instance. The mapping shows small low concentration regions to the left and bottom right which is indicated in Fig. 7. The mapping also shows a small high concentration slightly above the center. The mapping is able to capture some

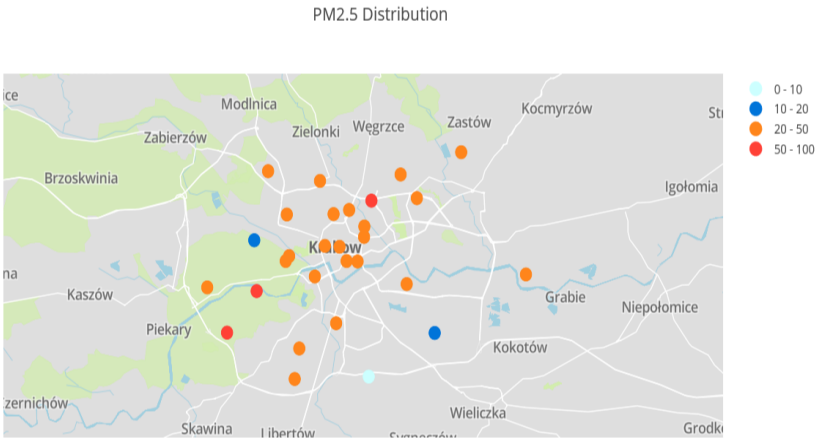

Figure 6: PM2.5 concentration labels for 7th March 6:00 AM at all 29 sensor location

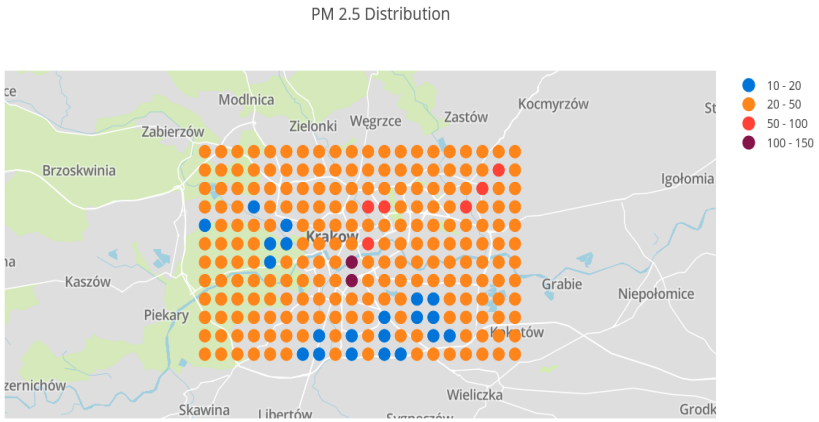

Figure 7: PM2.5 concentration labels for 7th March 6:00 AM mapped by semi-supervised L_1-regularized logistic regression model.

low and high concentration regions, however, it produces some high concentration labels in the class having PM levels 100-150 which does not exist in sensor locations and also does not show the very low concentration label 0-10 that one sensor has.

Other than L_1-regularized logistic regression, a model using the semi-supervised classification algorithm was developed using gaussian naive bayes approach. The model resulted accuracies around 30% for 0 Hour hence was not reported.

Table 2: Prediction accuracy using 0/1 loss for semi-supervised classification

	0th Hour	1st Hour	2nd Hour	3rd Hour	4th Hour
L_1-regularized Logistic Regression	69.4%	61.4%	57.5%	54.7%	51.7%

5.3 PDE functional identification of nonlinear dynamics algorithm

The prediction of testing data based on the interpolation of training data is feasible. We test this interpolation by inputting the training sensors data for each time and checking the testing sensor' position pm2.5 and compare it with the real number. The average R2 score of the prediction is 0.87. The figures below are plotting of prediction and exact value of four sensors .
(173)(196)(201)(228)

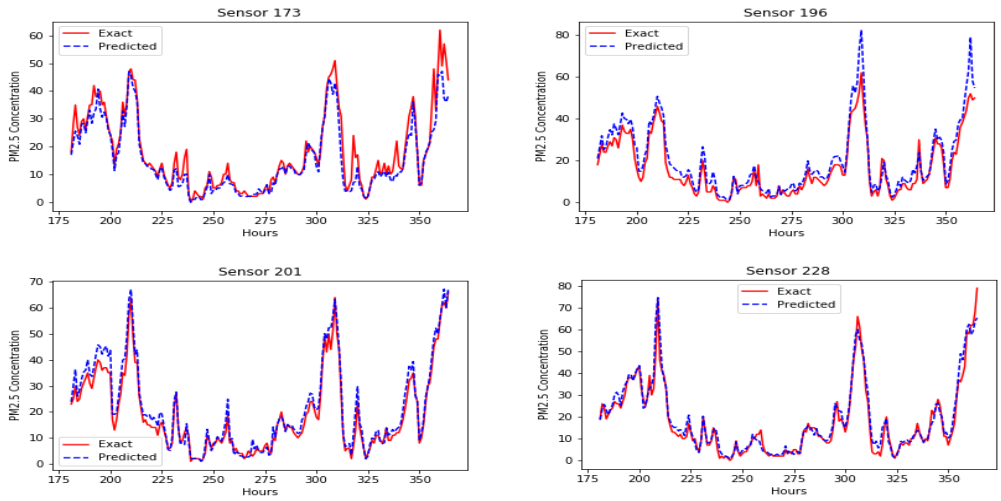

Figure 8: Comparison between exact and predicted PM2.5 concentrations for four sensors

We generated partial differential equations based on different month's data and entire data set that combines all month's data. All partial differential equations are different. The equation generated based on December's data is examined. We randomly selected a time point that included comprehensive pm2.5 temperature, the humidity of each training sensor and one testing sensor. We interpolated the pm2.5 trend based on training sensor data. We calculated the pm2.5, spatial partial derivative of pm2.5 that appeared in the PDE of the testing point based on the interpolation. In PDE, it needed the temperature and humidity of testing point, which I assumed that we have the system of predicting these two quantities and the prediction is the same as the measurement. With these values, I obtain the time derivative of pm2.5 at the testing point. The prediction of the next hour's pm2.5 of that test point is calculated by the current value of pm2.5 base on the interpolation plus the time derivative times the time interval. To predict the next two hours, we used the same method to calculate the predictive value of each training sensor and based on that prediction do the same process as before to get the next two hours' pm2.5 of the testing point. We calculated the next five hours' prediction pm2.5 of the testing point and solved for the R2 score with respect to the real value. The R2 score of the prediction is negative so the generated equation was in-feasible.

6 Conclusions

We propose three approaches for mapping and prediction of air pollution particulate matter (PM) in the city of Krakow, Poland based on temperature, humidity and PM measurements. We report the accuracies of two approaches: 1. Belkor Recommendation and 2. Semi-supervised classification using L_1-regularized Logistic Regression. In the first approach, a measurement of the PM level trend using Bellkor recommendation system achieved overall $R^2 = 0.928$. In the second approach, we map and predict PM2.5 concentration on 1km X 1km grid points across the geographical area usingL_1-regularization for feature selection and expectation maximization for semi-supervised learning. We classify PM concentrations into 6 classes using logistic regression. The model has 69.4% mapping accuracy and 61.5 % - 51.7% prediction accuracy for 1-4 hrs.

The important features for pollution mapping and prediction listed by Qi, Zhongang, et al[4] are temperature, wind strength and precipitation. Since we are missing two of the important features the model accuracy might not improve significantly using other techniques than the ones reported here. However, the contribution of the features may vary for different geographical locations as the weather is different in different locations. Since 8 months of data amounts to a large sample size, the mapping and prediction can be improved with improved feature selection rather than increasing number of samples.

References

[1] Apte, Joshua S., Kyle P. Messier, Shahzad Gani, Michael Brauer, Thomas W. Kirchstetter, Melissa M. Lunden, Julian D. Marshall, Christopher J. Portier, Roel CH Vermeulen & Steven P. Hamburg. (2017) High-resolution air pollution mapping with Google street view cars: exploiting big data. *Environmental Science Technology* 51(12):6999-7008.

[2] Alexeeff, Stacey E., Ananya Roy, Jun Shan, Xi Liu, Kyle Messier, Joshua S. Apte, Christopher Portier, Stephen Sidney & Stephen K. Van Den Eeden. (2018) High-resolution mapping of traffic related air pollution with Google street view cars and incidence of cardiovascular events within neighborhoods in Oakland, CA. *Environmental Health* 17(1):38.

[3] Qi, Zhongang, et al. "Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality." IEEE Transactions on Knowledge and Data Engineering 30.12 (2018): 2285-2297.

[4]Samuel H. Rudy1, Steven L. Brunton2, Joshua L. Proctor3, and J. Nathan Kutz.(2016) Data-driven discovery of partial differential equations *Science Advances* 10.1126/sciadv.1602614.

Individual Contributions

6.1 Gaurav Mahamuni

Coding up the 4.2 algorithm, running tests, tabulating final results and writing up the report for this algorithm. Data pre-processing.

6.2 Mingyu Wang

Coding up the 4.3 algorithm, running tests, tabulating final results and writing up the report for this algorithm.

6.3 Su Ye

Coding up the 4.1 algorithm, running tests, tabulating final results and writing up the report for this algorithm. Data pre-processing.