Active Learning for Air Quality Station Deployment

S. Deepak Narayanan deepak.narayanan@iitgn.ac.in
Apoorv Agnihotri apoorv.agnihotri@iitgn.ac.in
Nipun Batra nipun.batra@iitgn.ac.in
IIT Gandhinagar
Gujarat, India

Abstract

Recent years have seen a decline in air quality across the planet, with studies suggesting that air pollution is a significant cause of death. Governments have set up large scale air quality monitoring stations to aid them in formulating policies for air quality. However, these air quality stations are expensive to install, and have thus been often sparsely deployed. Motivated by sparse air quality monitoring and the expensive cost of air quality monitoring stations, we propose an active learning based solution to recommend locations to install air quality monitoring stations. We use a Gaussian Processes based approach for this purpose, motivated by their ability to encode prior knowledge using custom kernels. We demonstrate via extensive experimentation that our proposed approach outperforms several baselines on a publicly available dataset.

1. Introduction

Recent years have seen a decline in air quality across the planet, with studies suggesting that a significant proportion of the global population has reduced life expectancy by up to 4 years (Chen et al., 2013; Balakrishnan et al., 2019). A recent WHO report suggests that 9 out of 10 people breathe polluted air and air pollution is responsible for more than 7 million deaths in a year [1]. To tackle this increasing growth in air pollution and its adverse effects, governments have set up air quality monitoring stations to measure concentrations of various pollutants like NO_2, SO_2 and $PM_{2.5}$. $PM_{2.5}$ refers to the concentration of particles of diameter less than $2.5\mu m$. $PM_{2.5}$ has been shown to have a significant impact on health (Xing et al., 2016) and is used to measure air quality. One major issue with the deployment of these stations is the massive cost involved - installing each one of these stations costs around a million dollars. Owing to the high installation and maintenance costs, the spatial resolution of air quality monitoring is poor. As an example, in India, a developing country, the current number of air quality stations is around 150, whereas the government pollution agency estimates the requirement to be 4000 stations.

Air quality is affected by various meteorological factors such as humidity, temperature, wind direction and wind speed and thus a location could be affected by distant sources. As a consequence of the above factors, it could be the case that two locations that are spatially far are closer in their air quality and vice versa. Given this complexity of air quality along with the high cost involved in the installation and maintenance of air quality monitoring stations, it is important to be able to recommend station locations in an informative manner.

1. https://www.who.int/airpollution/en/

One natural strategy would be to install stations uniformly to maximize spatial coverage. The factors mentioned above and prior work (Hsieh et al., 2015; Zheng et al., 2013) suggest that uniformly installing stations may not be optimal. This motivates the problem that we try to answer in this paper: *Given a set of air quality monitoring stations, where do we install the next set of air quality monitoring stations/sensors so that we can best infer the air quality at unknown locations?*

We propose an active learning (Settles, 2009) based method for optimizing sensor placement[2]. The optimality of sensor placement is non-trivial to define as it can be based on various objectives, some of which are confounding. The objectives include, but are not limited to 1) minimizing sensor cost; 2) maximizing prediction accuracy for unmonitored regions/times; 3) minimizing labor cost; 4) minimizing maintenance cost. Our objective in this paper is to minimize prediction error at unmonitored locations.

To this end, we first propose a Gaussian Process Regressor (GPR) (Rasmussen, 2005) model to predict air quality ($PM_{2.5}$) values at unknown locations. We choose GPR since it can help encode domain knowledge easily by supporting custom kernels. GPs are Bayesian nonparametric models and thus the model complexity can be tuned based on data availability. We can obtain the mean and variance of the predictive distribution owing to its Bayesian nature. For our model, we propose uncertaintly sampling for active learning. In the case of GPs, since entropy is a monotonic function of the variance, our strategy to use uncertainty sampling is equivalent to decreasing the entropy the most (Settles, 2009) as a measure of uncertainty. Uncertainty sampling helps reduce the overall entropy of our model. We install stations by choosing the station with the maximum posterior variance and install them in an online manner: we install a station every month to the set of monitored stations and show that our model has a very low predictive error at unmonitored locations compared to various baselines. To the best of our knowledge, this is the first work that addresses the problem of online air quality station deployment, where stations are installed one at a time. Our work is completely reproducible and can be found at `https://github.com/sdeepaknarayanan/activepm/`.

2. Related Work

Our related work can be classified into two categories: i) techniques for sensor network deployment; and ii) active learning. We now discuss each of these categories.

Sensor Placement: Sensor deployment, in general, has been a well-studied problem. Krause et al. (2009) propose an algorithm to simultaneously optimize the placement and scheduling of sensors under constraints on the amount of power that is being consumed. Krause et al. (2008) propose a sensor deployment model for the early detection of water contamination. The common aspect in both Krause et al. (2008) and Krause et al. (2009) is that they both deploy sensors from scratch, without having any previous sensors installed. Guestrin et al. (2005), propose using mutual information for GPs, an optimization criterion to find the most information about the unsensed locations. In their case, they demonstrate that with increasing number of sensors, mutual information outperforms entropy in better predicting the phenomena under consideration. Though sensor deployment in general has been a well studied problem, a specific focus for air quality sensor deployment has been largely limited. Hsieh et al. (2015), propose an incremental station deployment strategy for

2. We use sensor and station interchangeably in this paper

air quality station deployment. By incremental, we refer to a strategy where there are locations that already have air quality monitors. They propose a semi-supervised approach to infer air quality and then subsequently recommend a fixed number of locations for installing air quality monitoring stations. Their scheme proposes installing all the recommended air quality stations at once.

Active Learning: Active Learning (AL) is a sub area in machine learning where the learning algorithm intelligently queries minimal data points to learn a good model (Settles, 2009). AL has been used widely in various applications including object categorization (Kapoor et al., 2007) and named-entity recognition (Shen et al., 2004). There are different scenarios and ways in which the learners query data points. Two widely used methods are pool-based and stream-based selective sampling (Settles, 2009). Learners typically have a notion of informativeness of the data points to make a decision. Common ones include uncertainty sampling where model uncertainty is used to choose data points (Lewis and Gale, 1994) and query by committee (Seung et al., 1992) where a committee of learners helps choose data points based on their level of agreement.

Our work mainly differs from related literature in that we do not deploy all the air quality monitoring stations at once; we rather install a station, use air quality data from this installed station, and then install the next station. Such a choice was motivated by the fact that our deployment scheme is highly appropriate and useful for a realistic deployment where there may not be sufficient funds to deploy all the stations at once.

3. Problem Statement

We formalize our problem statement here: Given a set of air quality monitoring stations S, along with information about their $PM_{2.5}$ values and meteorological conditions over some time $\{t_0, t_1, \ldots, t_k\}$, where each t_i denotes a timestamp, deploy air quality monitoring stations at a few candidate locations, every f timestamps, beginning on day t_{k+1}, such that estimation of air quality at unmonitored locations improves the most across timestamps beginning t_{k+1}. In this problem formulation, once an air quality monitoring station is deployed, its $PM_{2.5}$ data is readily available from the day after the deployment onwards.

4. Approach

Gaussian Processes (GPs) is a model that induces a distribution over functions. In any Gaussian Process model, we have a prior mean function $\mu : \mathbb{R}^d \to \mathbb{R}$ and a prior covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. These covariance functions (or kernels) quantify the similarity among different data points. The covariance matrix K has entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where k is the covariance function and $\mathbf{x}_i$ and $\mathbf{x}_j$ are two data points in $\mathbb{R}^d$. We use the following notation in the section: Let $X \in \mathbb{R}^{n \times d}$ be the input data points and $y \in \mathbb{R}^n$ be the labels. K_{XX} refers to the covariance matrix. $\hat{K}_{XX} = K_{XX} + \sigma_n^2 I_n$ is the covariance matrix added with zero-mean Gaussian noise of variance σ_n^2, I_n is the identity matrix of order n. $K_{X\mathbf{x^*}}$ refers to the vector that is formed by calculating the covariance function between any test point $\mathbf{x^*}$ and all the train points. Once the model is trained to fit the data, we obtain the predictive posterior distribution. For a test point $\mathbf{x^*}$ and its corresponding predictive distribution $y*$, we have

$$\mathbb{E}[y^*|X, \mathbf{y}] = \mu(\mathbf{x^*}) + K_{X\mathbf{x^*}}^T \hat{K}_{XX}^{-1} \mathbf{y} \tag{1}$$

$$Var[y^*|X, \mathbf{y}] = k(\mathbf{x^*}, \mathbf{x^*}) - K_{X\mathbf{x^*}}^T \hat{K}_{XX}^{-1} K_{X\mathbf{x^*}} \tag{2}$$

In GPs, there are usually several standard kernels that are used for a variety of problems. The addition or multiplication of kernels still results in a valid kernel and hence they are

combined in a variety of ways to create custom kernels that are typically used to encode domain knowledge and capture complex dependencies between features. In this paper, we use a few standard kernels and combine them to create a custom kernel described below. We use a combination of three standard kernels in our work - the Matérn kernel, the radial basis function (RBF) kernel and the periodic kernel. We use the matern kernel with $\nu = 3/2$ (Matérn32). We choose this particular value of ν to account for less smoothness in the approximation function. In the dataset that we used in this work, we have the following as features: latitude, longitude, weather and meteorological factors humidity, pressure, wind speed and direction. Our final kernel k_{GPR} uses the following kernels.

$$k_{longitude,latitude} = k_{Matern32} + k_{Matern32} \tag{3}$$

$$k_t = k_{Matern32} + \sum_{i=1}^{5} k_{Matern32} \times k_{Periodic} \tag{4}$$

$$k_{Temp,Hum} = k_{Matern32} + k_{Matern32} \tag{5}$$

$$k_{Windspeed} = k_{RBF} \qquad k_{Weather} = k_{RBF} \qquad k_{Pressure} = k_{RBF} \tag{6}$$

$$k_{GPR} = \prod_{f \in Features} k_f \tag{7}$$

where Features := {(Longitude, Latitude), Time, (Temperature, Humidity), Windspeed, Weather, Pressure }. In our kernel, we use the same spatiotemporal kernel ($k_{longitude,\ latitude}$ and k_t) that was proposed by Guizilini and Ramos (2015). Our rationale for using their kernel is as follows: The two Matérn32 kernels naturally can account for short term and long term spatial trends that we wish to capture (Eqn. 3), and the temporal kernel can for periodic decay while also capturing long term or short term trends in the temporal domain. In Eqn. 5, we use the fact that temperature and humidity are related and appropriately capture their variations together. The choice of two Matérn kernels allows the regressor to learn nonsmooth relationships as well as differing trends. For modeling wind speed, pressure, and weather, we used RBF kernels to model smoother variations in their values. We refer to our GP with the proposed kernel as GPR in this paper.

5. Evaluation

Datasets: To evaluate our proposed approach we use the dataset released by Zheng et al. (2015, 2014, 2013). The dataset contains hourly $PM_{2.5}$ measurements for a total of 36 air quality monitoring stations in Beijing from 1st May 2014 to 30th April 2015. In addition to the $PM_{2.5}$ data, the dataset also consists of weather and meteorological data. Meteorological data includes humidity, wind speed, wind direction, pressure, and temperature. We downsampled the data to a single measurement per day per station because of the following reasons: (1) Missing data: In our dataset, we had around 13.3 % of $PM_{2.5}$ data missing and up to 30% and 40% data for wind speed and humidity respectively; and (2) The fact that city authorities often look at 24-hour exposures before deciding to take actions.

Pool based active learning setup: For the purpose our experiments, we maintain three sets of stations - the train set S_{train}, the test set S_{test} and the pool set S_{Pool}. S_{train} contains the air quality stations that are currently monitored, S_{test} contains air quality stations where we wish to estimate the air quality, and S_{Pool} contains the set of candidate air quality monitoring stations (locations) to be installed every month. We query a station from the

pool set to be added to the train set once every month. More formally, let s_q be the queried station. Then $S_{train} = S_{train} \cup \{s_q\}$ and $S_{pool} = S_{pool} \setminus \{s_q\}$. From the day of querying onwards, the $PM_{2.5}$ values are available for the pool stations as they are part of the train set. In our setup, motivated by sparse air quality monitoring stations, we use 6 stations for S_{train}, 24 stations for S_{pool} and the remaining 6 stations for S_{test}. This results in a total of 6 different test sets, each with 5 different training and pool sets. We evaluate our model across all the 30 different station splits.

Models: Our key intuition is that a model that can estimate air quality well will be more selective. Therefore to choose regressor(s) for active learning we initially experimented with predicting $PM_{2.5}$ values on our entire dataset using multiple regressors. We choose the top 3 regressors that performed the best in this task from among a total of 8 regressors including our GPR. While we do not go into the details of this experiment due to space constraints, we release the entire code base, analysis and results for this experiment as part of our repository linked earlier. We found that XgBoost (Chen and Guestrin, 2016), k-Nearest Neighbors (k-NN) and our GPR were the three best regressors in terms of estimating air quality and hence, we proceeded to perform active learning with these regressors.[3]

Active Learning Strategies:
Query By Committee (Seung et al., 1992): In Query by Committee (QBC) we maintain a committee consisting of multiple learners, all of which are trained on the same train dataset. In our setting, we create a committee by using the same learners set to different hyperparameters. We look at the pool station (location) with which this committee disagrees the most. We use the standard deviations in the predictions of these learners to quantify the disagreement and choose the station in the pool set having maximum disagreement as the station to be added to the train set. In all our experiments we use a committee size of 5. Note that we use QBC only for k-NN and XgBoost.
Random Sampling : Random sampling is a sampling method in which we choose a station uniformly at random from the pool set and add it to the train set. It is widely used in Active Learning as a baseline (Settles, 2009). We use 5 different random seeds and report the mean and standard deviation in our predictions of $PM_{2.5}$.
Uncertainty Sampling (Settles, 2009): Gaussian Process Regressor provides us with the posterior predictive mean and variance. This variance gives us the confidence that the GPR has in its predictions. We, therefore, choose the pool station that the model is least confident about and add it to train set from the pool set.

Evaluation Metrics: We denote the prediction of $PM_{2.5}$ at station i at time t by $\hat{y_t^i}$. We consider three evaluation metrics: Root Mean Squared Error of a station i, $RMSE^{Station}(i)$ and Mean Root Mean Squared Error (MeanRMSE) for computing the mean errors in the predictions of $PM_{2.5}$ across stations.

$$RMSE^{Station}(i) = \sqrt{\frac{\sum_{i=0}^{T}(\hat{y_t^i} - y_t^i)^2}{T}} \quad (8) \qquad MeanRMSE = \frac{\sum_{i=0}^{|S|} RMSE^{Station}(i)}{|S|} \quad (9)$$

3. In this paper, we refer to XGBoost as XGB and k-Nearest Neighbors as k-NN.

In Eqn. 8, T is the total number of timestamps for station i. $\hat{y_t^i}$ and y_t^i refer to the predicted value and the ground truth value for the t^{th} timestamp of the i^{th} station. In Eqn. 9, i refers to a station and $|S|$ denotes the total number of stations.

6. Experimental Results

We report the MeanRMSE in prediction across all the different sets of data in Table 1. From Table 1, we can see that our GPR has the least error across all the timestamps and splits of the data when compared to the other baselines. From Table 2, we can see that our method provides up to **40 %** improvement in prediction over the best random method baseline and also an average improvement of up to **14%** across both the random baselines.

Table 1: Mean RMSE with all < Regressor, Active Learning Strategy > pairs

Regressor	Active Learning Strategy	Mean RMSE
GPR	Uncertainty Sampling	**20.67**
GPR	Random Sampling	24.38 ± 5.08
XGB	QBC Sampling	22.67
XGB	Random Sampling	24.13 ± 1.71
k-NN	QBC Sampling	30.68
k-NN	Random Sampling	30.54 ± 1.23

Table 2: Relative improvement in predictions compared to different random strategies

	Max. %		Mean %	
	GPR (US)	XGB (QBC)	GPR (US)	XGB (QBC)
GPR (Rd)	33.57	33.79	14.27	4.96
XGB (Rd)	40.77	23.89	13.73	5.54

We report the best % improvement and the mean % improvement in predictions of our GPR and XGB when compared with a random method. Note that our GPR provides the maximum improvement in predictions on average, clearly outperforming the other regressor. Note: US - Uncertainty Sampling and Rd - Random Sampling

7. Conclusions and Discussions

In this paper, we address the problem of air quality station deployment in an online setting. To the best of our knowledge, this is the first work addressing this problem of online station deployment. In our work, we propose a Gaussian Process Regressor that can encode domain knowledge by supporting custom kernels to choose locations. We demonstrate empirically that our proposed method outperforms several baselines.

In our setting, we install a single station at the end of every month. An extension to this would be allowing for installation of many stations every month so that the predictive error decreases the most. This is a non-trivial extension if we use predictive variance as our query scheme, as selecting multiple stations with the highest uncertainty would not take into account potentially underlying correlations between stations. We avoid this problem by installing one station every month and by leveraging sufficient data from this station to choose the next station. We also implicitly assume the cost of installing stations is the same at any location. This assumption might not hold true. Installations of station at different locations could entail different costs. This imposes additional constraints on top of our current formulation and the key to solving this problem will be balancing the cost of station installation and uncertainty reduction. Currently we use stationary kernels to model air quality. An extension could be to explore non-stationary kernels that can model air quality better.

References

Kalpana Balakrishnan, Sagnik Dey, Tarun Gupta, RS Dhaliwal, Michael Brauer, Aaron J Cohen, Jeffrey D Stanaway, Gufran Beig, Tushar K Joshi, Ashutosh N Aggarwal, et al. The impact of air pollution on deaths, disease burden, and life expectancy across the states of india: the global burden of disease study 2017. *The Lancet Planetary Health*, 3 (1):e26–e39, 2019.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL `http://arxiv.org/abs/1603.02754`.

Yuyu Chen, Avraham Ebenstein, Michael Greenstone, and Hongbin Li. Evidence on the impact of sustained exposure to air pollution on life expectancy from china's huai river policy. *Proceedings of the National Academy of Sciences*, 110(32):12936–12941, 2013.

Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 265–272, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102385. URL `http://doi.acm.org/10.1145/1102351.1102385`.

Vitor Guizilini and Fabio Ramos. A nonparametric online model for air quality prediction. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 651–657. AAAI Press, 2015. ISBN 0-262-51129-0. URL `http://dl.acm.org/citation.cfm?id=2887007.2887098`.

Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. Inferring air quality for station location recommendation based on urban big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 437–446. ACM, 2015.

A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408844.

A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin. Simultaneous placement and scheduling of sensors. In *2009 International Conference on Information Processing in Sensor Networks*, pages 181–192, April 2009.

Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, November 2008. doi: 10.1061/(asce)0733-9496(2008)134:6(516). URL `https://doi.org/10.1061/(asce)0733-9496(2008)134:6(516)`.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, 1994. ISBN 0-387-19889-X.

Carl Edward Rasmussen. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, nov 2005. ISBN 9780262182539.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL `http://burrsettles.com/pub/settles.activelearning.pdf`.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130417. URL `http://doi.acm.org/10.1145/130385.130417`.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1219030. URL `https://doi.org/10.3115/1218955.1219030`.

Yu-Fei Xing, Yue-Hua Xu, Min-Hua Shi, and Yi-Xin Lian. The impact of pm2.5 on the human respiratory system. *Journal of Thoracic Disease*, 8(1), 2016. ISSN 2077-6624. URL `http://jtd.amegroups.com/article/view/6353`.

Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444. ACM, 2013.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55, September 2014. ISSN 2157-6904. doi: 10.1145/2629592. URL `http://doi.acm.org/10.1145/2629592`.

Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 2267–2276, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2788573. URL `http://doi.acm.org/10.1145/2783258.2788573`.