Towards Active Air Quality Station Deployment

Zeel B Patel [1] **Nipun Batra** [1]

Abstract

Air pollution is a global problem and has a severe impact on human health. Fine-grained air quality (AQ) monitoring is important in mitigating air pollution. However, existing AQ station deployments are sparse due to installation and operational costs. In this work, we propose an Active Learning-based method for air quality station deployment. We use Gaussian processes and several classical machine learning algorithms (Random Forest, K-Neighbors, and Support Vector Machine) to benchmark several active learning strategies for two real-world air quality datasets (Delhi, India and Beijing, China).

1. Introduction and Related Work

Today, 91% of the global population lives under unsafe levels of air quality[1]. Long-term exposure to PM2.5 increases cardiopulmonary mortality by 6–13% per 10 µg/m3 of PM2.5, which causes yearly 8 million deaths worldwide (Kloog et al., 2013). AQ is affected by multiple factors including but not limited to physicochemical processes, meteorological variables, and the geography of a place. Primary air pollution sources are solid fuels used in domestic cooking, industrial plants, vehicular emissions, roadside dust, and construction activities. Thus, air pollution is a complex spatio-temporal phenomenon, and fine-grained AQ monitoring is essential to make informed decisions towards air pollution mitigation.

The installation and maintenance of a single AQ monitoring station cost roughly 0.2M USD per year. Developing and underdeveloped countries have remarkably fewer air quality stations than required. For example, India has around 200 continuous monitoring stations over the need for 4000 such stations according to AQ experts[2]. A set of work shows that uniform placement of sensors is not optimal (Hsieh et al., 2015). Thus, optimization is crucial for air quality station deployment. Recent work has proposed a chemical process modeling approach for sensor placement (Klise et al., 2017), but it requires extensive domain knowledge and rich information about air pollution sources. Our work closely aligns with (Narayanan et al., 2020; Narayanan & Batra, 2020) in terms of the experimental setup and several techniques. We use an additional query strategy (Mutual information maximization), and fine-grained data compared to (Narayanan & Batra, 2020).

Active learning techniques are used for intelligent data annotation in supervised learning. Similar methods are also proposed and used for sensor placement (Krause et al., 2008). We attempt to demonstrate a majority of the active learning techniques available for regression settings in this work. Additionally, we also analyze various strategies for choosing the test stations so that results generalize for the entire unmonitored space. Note that we do not account for the budget constraints for stations in the current work but plan to use them in future work.

We believe that our work will provide insights on challenges in applying active learning techniques in the real-world dataset and several possible variations that can be used in such applications.

2. Problem Statement

Given a set of air quality monitoring stations S, along with their PM$_{2.5}$ values and spatial locations (Latitude, Longitude) starting from time t_0, deploy an air quality monitoring station at one of the few candidate locations, every k timestamps (on timestamps $t_k, t_{2k}, t_{3k}, ...$), such that estimation of air quality at unmonitored locations improves the most across timestamps beginning from t_k. Once a station is deployed, its PM$_{2.5}$ data is available from the day after the deployment.

3. Methodology

3.1. Gaussian processes (GPs)

Gaussian processes assume a prior distribution (Gaussian) over the data using a covariance function, and then posterior

[1] IIT Gandhinagar, India. Correspondence to: Zeel B Patel <patel_zeel@iitgn.ac.in>, Nipun Batra <nipun.batra@iitgn.ac.in>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] https://www.who.int/health-topics/air-pollution#tab=tab_2

[2] https://urbanemissions.info/

predictions are obtained using the Bayes rule. Given input locations X with d features (latitude and longitude), corresponding observed values $\mathbf{y}$, and a kernel function $K(\cdot,\cdot)$; posterior distribution at new locations X_* can be obtained using the following equations:

$$\mathcal{K} = K(X, X) \tag{1}$$

$$\mathcal{K}_* = K(X_*, X) \tag{2}$$

$$\mathcal{K}_{**} = K(X_*, X_*) \tag{3}$$

$$\boldsymbol{\mu_*} = \mathcal{K}_* \left[\mathcal{K} + \sigma_n^2 I\right]^{-1} \mathbf{y} \tag{4}$$

$$\Sigma_* = \mathcal{K}_{**} - \mathcal{K}_* \left[\mathcal{K} + \sigma_n^2 I\right]^{-1} \mathcal{K}_*^T \tag{5}$$

where $\boldsymbol{\mu_*}, \Sigma_*, \sigma_n^2$ are posterior mean, variance and likelihood noise, respectively. The key element in GP regression is a covariance function (aka kernel). Kernels regulate the characteristics of the resultant fit. For example, a Periodic kernel is helpful to learn the cyclic data. The Squared Exponential kernel is infinitely flexible to learn any data generated from stationary processes. Combinations of kernels via addition/multiplication also remain valid kernels as leveraged by (Guizilini & Ramos, 2015) to model the AQ.

3.2. Active Learning

Active learning (AL) is a subset of supervised learning, where a model intelligently queries the data points for labels to minimize the number of training points (Settles, 2009). AL has been used in various applications including but not limited to object detection (Kapoor et al., 2007), image classification (Gal et al., 2017) and natural language processing (Siddhant & Lipton, 2018). There are two major settings in which active learning can be applied: i) Pool-based AL and ii) Streaming AL (Settles, 2009). Pool based setting is more suitable for our work. Thus, we discuss it in the following section.

3.2.1. Pool based active learning

Pool-based active learning uses a pool of candidate instances in the unlabeled dataset. A model is trained with the initial training points. Then, we can use various query strategies with the help of the trained model to iteratively select instances from the Pool and query for their label. Usually, a human annotator or an oracle provides the label for the queried instances.

In station deployment, we already have several stations installed in the area of interest, which can be treated as the training data. In addition, we would have some candidate locations (the Pool) where the station installation is feasible. Thus, pool-based active learning is directly applicable to station deployment tasks. We discuss various query strategies that can be used in Pool based settings in Section 5.2.

4. Dataset

4.1. Delhi air quality

We download the Delhi data from 33 stations for the month of August-2019 from the OpenAQ[3]. The data contains PM$_{2.5}$ at 15 minutes granularity. We down-sample the dataset to daily granularity to avoid missing data at a finer resolution. We leverage the spatial features as the predictors of air quality in our experiments.

4.2. Beijing air quality

We leverage the Beijing data used by (Zheng et al., 2013). The dataset includes hourly PM$_{2.5}$ data from 36 stations in Beijing and meteorological data (temperature, humidity, pressure, wind speed, and six categorical weather variables) from the stations in the same district. The duration of the dataset is one year (2013-2-8 to 2014-2-8). We consider a continuous segment of 277 time-stamps with no missing PM$_{2.5}$ data in any of the stations (2013-11-08 to 2013-11-20). We only leverage the spatial features (longitude and latitude) in this work. The dataset is publicly available via the official site[4].

5. Experiments

There are two components in our experimental setup: i) spatial inference; ii) active learning. We start an experiment with 6 training stations at time t_0, assuming them as an initial deployment (train stations). 6 random stations are kept as test stations assuming their locations as unmonitored locations. We discuss various strategies to carefully select test stations in Section 5.3. The remaining stations are used as the pool for potential deployments. We infer the PM$_{2.5}$ values at test locations to evaluate the performance of newly installed stations in addition to initial deployment. An active learning strategy is called on the trained model to vote for the next deployment at each time-stamp. For Delhi, we directly deploy the voted station because one time-stamp is 24 hours. For Beijing, we use majority votes to choose a station for deployment after every 24 time-stamps (24 hours). We repeat the same experiment with ten random splits of the train, pool, and test stations for both datasets and report the mean RMSE across them.

Now, we discuss the methods and active learning strategies employed in our experiments.

[3] `https://openaq.org/`

[4] `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Air20Quality20Data.zip`

5.1. Regression methods

We use the following regression methods in our experiments. Due to the paucity of time, we could not fine-tune the hyperparameters in any of the Non-GP methods.

5.1.1. Random Forest

Random Forest is widely used and known to perform efficiently on the non-linear regression tasks (Fawagreh et al., 2014). It uses an ensemble of multiple decision trees for regression. We use Random Forest with default hyperparameters with Scikit-learn (Pedregosa et al., 2011).

5.1.2. K-Neighbors Regression

K-Neighbors regression is a non-parametric method for regression. At any query location, prediction is the average of the nearest K stations. We use K-Neighbors regression with five nearest locations.

5.1.3. Support Vector Regression

Support vector regression (SVR) (Drucker et al., 1997) is a fast method for regression that uses various kernels to transform the input space into a higher dimension and learns the relationship between the transformed space and the output. We use SVR with default hyperparameters with Scikit-learn (Pedregosa et al., 2011).

5.1.4. Gaussian processes

We use Gaussian process regression with Matern32 kernel where ARD (Automatic Relevance Determination) (Rasmussen & Williams, 2005) is enabled. ARD enables learning different lengthscales for different features (latitude and longitude), allowing variable smoothness for each feature.

5.2. Active learning query strategies

5.2.1. Query by committee

In Query by Committee (QBC) (Seung et al., 1992), we build a committee of multiple learners trained on the same dataset. We use Random Forest, Support vector regressor, and K-Neighbors regressor to build the committee of learners. We deploy a station from the pool for which committee learners disagree the most in terms of AQ predictions. We use standard deviation in the predictions as a metric for disagreement. Thus, a location with the highest standard deviation in predictions among all committee regressors is chosen for deployment.

5.2.2. Uncertainty sampling

Uncertainty sampling (Lewis & Gale, 1994; Settles, 2009) is among the most commonly used querying strategies. In this strategy, the learner queries the instances for which it is least confident. Entropy is a widely used measure for uncertainty (Settles, 2009). In Gaussian processes, predictive variance can be taken as a measure of entropy (Krause et al., 2008). Thus, we deploy a station with the highest predictive variance from the pool. Note that this method is used only in the GP approach.

5.2.3. Maximum mutual information (MI)

Reducing uncertainty about the unmonitored locations is equivalent to maximizing mutual information between monitored and unmonitored locations (Krause et al., 2008). We can exploit the submodularity of mutual information to greedily deploy the stations at the benefit of time-complexity (from NP-complete to polynomial). We use a greedy algorithm (Krause et al., 2008) to deploy the next station using mutual information criterion.

5.2.4. Random sampling

Random sampling is a widely used baseline to evaluate the active learning strategies (Settles, 2009). We randomly deploy a station from the pool set at k time-stamps intervals.

5.3. Test station selection strategies

We employ several techniques to choose the test stations selectively. We believe that these techniques will ensure that inference at test locations generalizes for the entire unmonitored space.

5.3.1. Random selection

In this basic strategy, we randomly choose the test stations without employing any specific constraints on them. This strategy is generally used in any train-test splitting in relevant experiments. Figure 1 illustrates one example split of a train, pool, and test stations on the Beijing dataset.

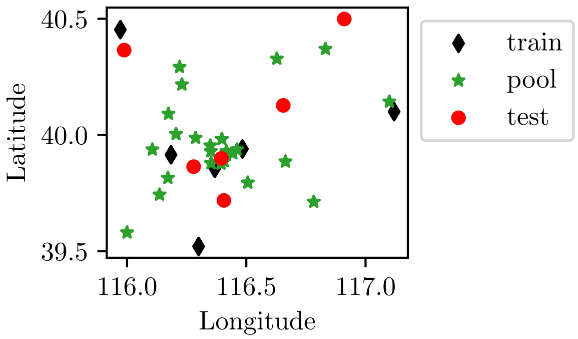

Figure 1. Train, pool and test stations on Beijing dataset. Tests stations are selected with random sampling.

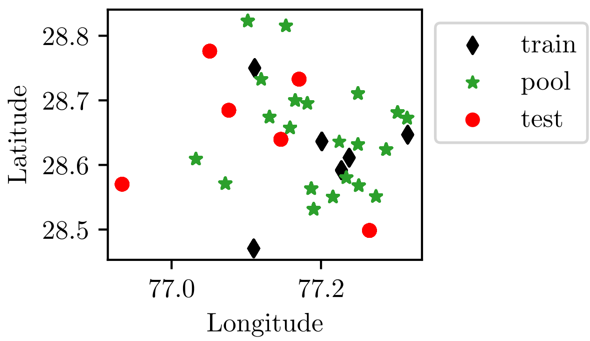

Figure 2. Train, pool and test stations on Delhi dataset. Tests stations are selected with LHS sampling.

5.3.2. D^2 SAMPLING SELECTION

D^2 sampling (Arthur & Vassilvitskii, 2006) iteratively chooses the next point which is farthest from the current points; thus, it avoids choosing test stations too close to each other. Algorithm 1 illustrates the D^2 sampling.

Algorithm 1 D^2*sampling*

```
Data: Station locations $S$, number of samples $N$
Result: Subset of stations $S_{test} \subset$S
Choose $S_1$ randomly and assign to $S_{test} = \{S_1\}$.
$d(S_i, S_j)$ is Euclidean distance between stations $S_i$ and $S_j$.
for $i \leftarrow$ 2,3,...,N do
    Sample $S_i$ with probability $\frac{d^2(S_i,S_{test})}{\sum_{S_j \in S_{test}} d^2(S_j,S_{test})}$ and add to $S_{test}$
end
```

5.3.3. LATIN HYPERCUBE SAMPLING (LHS) SELECTION

LHS (Iman et al.) samples are guaranteed to cover the space homogeneously and thus, we can generalize an approach better if test stations are chosen with this method (trying to provide the best estimate of AQ in each unmonitored location in the space of interest). We create a 2D instance of LHS samples and choose the closest stations to the samples as the test stations (distance measured was euclidean). Figure 2 shows an example choice of train, pool, and test stations on the Delhi dataset.

5.4. Evaluation metric

We use RMSE at the test stations as the primary evaluation metric. After each deployment, we compute the RMSE for each test station. For a single experiment (deploying all stations), we compute mean RMSE for each test station across all the time-stamps. After that, we take the mean RMSE across all test stations as the final RMSE for an experiment. Finally, we report the average RMSE across ten such experiments with different random initialization (shuffling train, pool, and test stations).

6. Results and Analysis

6.1. Delhi air quality dataset

	Test stations selection		
Models + Q strategy	Random	D^2	LHS
GP + MI	**14.80**	**12.96**	**12.42**
GP + Uncertainty	16.41	13.69	13.33
GP + Random	16.61	13.45	12.73
RF, SVR, KNN + QBC	12.30	11.72	11.68
SVR, KNN + QBC	**11.71**	**11.36**	11.63
RF + Random	14.06	12.78	12.27
KNN + Random	11.89	11.78	**11.41**
SVR + Random	11.76	11.93	11.99

Table 1. Delhi: Mean-RMSE for test stations for all time-stamps after 10 random experiments. For any test station selection technique, MI (mutual information) and Query by Committee outperform their respective random baselines. Q strategy: Active learning query strategy.

Table 1 shows the RMSE on the Delhi air quality dataset. We present the analysis of RMSE as following,

- GP-Based methods:
  - **Random selection**: GP with MI yields the best RMSE. GP with uncertainty sampling is better than Random sampling.
  - **D^2 selection**: GP with MI is still the best approach here, but GP with Random sampling is yielding better RMSE than GP with uncertainty sampling. This suggests that uncertainty sampling is not selecting the most informative station locations.
  - **LHS selection**: Results resonate with D^2 sampling, where again, GP with uncertainty sampling does not select information stations.
- Non-GP methods:
  - **Random selection**: QBC with SVR and KNN yield the best RMSE among other baselines. Note that Random Forest yields higher RMSE than other baselines, and as a result, QBC with SVR, KNN, and RF has higher RMSE than QBC with SVR and KNN.
  - **D^2 selection**: Results resonate with Random selection in all the aspects. Overall RMSE has been reduced in this setting.

- **LHS selection**: KNN with random sampling yields the best result in this setting though QBC with SVR, KNN also yields comparable RMSE.

Figure 3 shows the variation in RMSE due to various AL techniques as the new stations are installed for an experiment with LHS selection. After nine placements, stations placed with QBC with SVR, KNN, and GP with MI constantly yield lower RMSE than stations placed with other approaches.

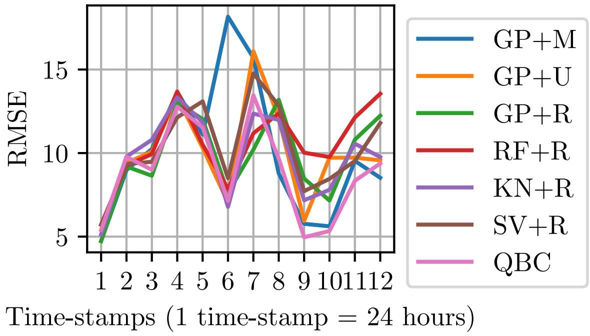

Figure 3. RMSE after installing each sensor with various combinations of models and AL techniques on Delhi dataset with LHS selection for test stations. Active learning strategies start performing better after several placements. After nine placements, QBC with SVR, KNN, and GP with MI constantly yield lower RMSE than other approaches. M: MI, U: Uncertainty, R: Random, RF: Random Forest, SV: SVR, KN: KNN, GP: Gaussian processes, QBC: Query by Committee.

6.2. Beijing air quality dataset

	Test stations selection		
Models + Q strategy	Random	D^2	LHS
GP + MI	**33.80**	38.43	42.50
GP + Uncertainty	33.97	38.73	42.43
GP + Random	34.41	**37.41**	**41.59**
RF, SVR, KNN + QBC	**30.86**	35.32	**36.85**
RF + Random	31.68	**34.54**	37.54
KNN + Random	33.11	37.09	39.95
SVR + Random	31.42	35.11	38.83

Table 2. Beijing: Mean-RMSE for test stations for all time-stamps after 10 random experiments. For random sampling on the test stations, MI (mutual information) and Query by Committee outperform their respective random baselines. For D^2 and LHS sampling, active learning techniques do not yield better performance compared to the random baselines. We try to explain this behavior in Section 6.2.

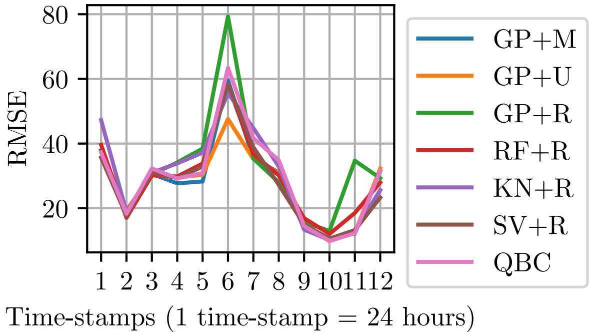

Figure 4. RMSE after installing each sensors with various combinations of models and AL techniques on Beijing dataset. We conclude that improvement in RMSE is not remarkable with active learning techniques while compared with random sampling baselines.

Table 2 shows the RMSE for all the experiments on Beijing dataset (Zheng et al., 2013). We present our analysis as the following,

- GP-based methods:
  - **Random selection:** GP with MI yields better RMSE compared to Uncertainty sampling and random sampling. Difference in RMSE for MI and uncertainty is not significant here.
  - **D^2 selection:** GP with random sampling yields better results compared to GP with MI and uncertainty sampling. RMSE, in general, is high due to D^2 selection, which forces test stations to have maximum distance among them. We confirm with six-fold cross-validation on inference alone with GP that remote stations yield high RMSE than concentrated stations. Overall we do not find active learning strategies work well in this setting.
  - **LHS selection:** The results here resonate with D^2 selection except that RMSE is comparably high in this setting. We believe that similar reasons affect the RMSE in this setting as well.
- Non-GP methods:
  - **Random selection:** QBC yields the best results compared to random baselines with RF, KNN and SVR.
  - **D^2 selection:** Random Forest with random sampling has the least RMSE but QBC also has a comparable RMSE.
  - **LHS selection:** QBC is yielding the best results showing its efficacy against the random sampling in this case.

Figure 4 shows the variation in RMSE due to various AL techniques as the new stations are installed. We have used the random test station selection setting here. We conclude that improvement in RMSE is not remarkable with active learning techniques compared with random sampling baselines.

7. Future work

We will extend our study in the following future directions,

- We will use more sophisticated GP models such as well-designed combinations of kernels (Guizilini & Ramos, 2015) and more flexible GP models such as non-stationary GPs (Plagemann et al., 2008).
- We will benchmark the implemented techniques on other datasets of polluted cities such as Shanghai, China and Mumbai, India.
- We would like to explore other active learning techniques suitable for regression tasks and apply for the air quality station deployment.

References

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. Support vector regression machines. Advances in neural information processing systems, 9:155–161, 1997.

Fawagreh, K., Gaber, M. M., and Elyan, E. Random forests: from early developments to recent advancements. Systems Science & Control Engineering, 2(1):602–609, 2014. doi: 10.1080/21642583.2014.956265. URL https://doi.org/10.1080/21642583.2014.956265.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In International Conference on Machine Learning, pp. 1183–1192. PMLR, 2017.

Guizilini, V. and Ramos, F. A nonparametric online model for air quality prediction. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pp. 651–657. AAAI Press, 2015. ISBN 0262511290.

Hsieh, H.-P., Lin, S.-D., and Zheng, Y. 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, pp. 437–446, 2015.

Iman, R. L., Davenport, J. M., and Zeigler, D. K. Latin hypercube sampling (program user's guide). [lhc, in fortran]. URL https://www.osti.gov/biblio/5571631.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. Active learning with gaussian processes for object categorization. In 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE, 2007.

Klise, K. A., Nicholson, B. L., and Laird, C. D. Sensor placement optimization using chama. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

Kloog, I., Ridgway, B., Koutrakis, P., Coull, B. A., and Schwartz, J. D. Long-and short-term exposure to pm2. 5 and mortality: using novel exposure models. Epidemiology (Cambridge, Mass.), 24(4):555, 2013.

Krause, A., Singh, A., and Guestrin, C. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research, 9(2), 2008.

Lewis, D. D. and Gale, W. A. A sequential algorithm for training text classifiers. In SIGIR'94, pp. 3–12. Springer, 1994.

Narayanan, S. Deepak, A. A. and Batra, N. Active learning for air quality station deployment. In ICML 2020 Workshop on Real World Experiment Design and Active Learning, 2020.

Narayanan, S. D., Agnihotri, A., and Batra, N. Active learning for air quality station location recommendation. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, CoDS COMAD 2020, pp. 326–327, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377386. doi: 10.1145/3371158.3371208. URL https://doi.org/10.1145/3371158.3371208.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

Plagemann, C., Kersting, K., and Burgard, W. Nonstationary gaussian process regression using point estimates of local smoothness. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 204–219. Springer, 2008.

Rasmussen, C. E. and Williams, C. K. I. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005. ISBN 026218253X.

Settles, B. Active learning literature survey. 2009.

Seung, H. S., Opper, M., and Sompolinsky, H. Query by committee. In Proceedings of the fifth annual workshop on Computational learning theory, pp. 287–294, 1992.

Siddhant, A. and Lipton, Z. C. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. arXiv preprint arXiv:1808.05697, 2018.

Zheng, Y., Liu, F., and Hsieh, H.-P. 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, pp. 1436–1444, 2013.