

MATRIX COMPLETION WITH VARIATIONAL GRAPH AUTOENCODERS: APPLICATION IN HYPERLOCAL AIR QUALITY INFERENCE

Tien Huu Do^{**}, Duc Minh Nguyen^{**}, Evaggelia Tsiligianni^{**}, Angel Lopez Aguirre^{†‡},
Valerio Panzica La Manna[◇], Frank Pasveer[◇], Wilfried Philips^{*†}, Nikos Deligiannis^{**}

^{*} Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium

[†] IPI Ghent University, Sint-Pietersnieuwstraat 25, B-9000 Ghent, Belgium

[‡] Escuela Superior Politécnica del Litoral, Km 30.5 Vía Perimetral, Guayaquil, Ecuador

^{*} imec, Kapeldreef 75, B-3001 Leuven, Belgium

[◇] Holst Center, imec, High Tech Campus 31, 5656 AE Eindhoven, The Netherlands

ABSTRACT

Inferring air quality from a limited number of observations is an essential task for monitoring and controlling air pollution. Existing inference methods typically use low spatial resolution data collected by fixed monitoring stations and infer the concentration of air pollutants using additional types of data, e.g., meteorological and traffic information. In this work, we focus on street-level air quality inference by utilizing data collected by mobile stations. We formulate air quality inference in this setting as a graph-based matrix completion problem and propose a novel variational model based on graph convolutional autoencoders. Our model captures effectively the spatio-temporal correlation of the measurements and does not depend on the availability of additional information apart from the street-network topology. Experiments on a real air quality dataset, collected with mobile stations, shows that the proposed model outperforms state-of-the-art approaches.

Index Terms— air quality inference, variational graph autoencoder, graph-based matrix completion, deep learning.

1. INTRODUCTION

Air pollution is one of the most serious threats for the human health and the environment. In order to mitigate air pollution, we need to accurately measure air quality at very high spatial and temporal rates, especially within urban areas. Fixed monitoring stations have been deployed to measure the concentration of air pollutants. Given the high cost of the necessary instruments, the number of such installations is limited. Although fixed stations can collect measurements with high temporal resolution, their spatial resolution is very low; hence, there is a need to spatially infer the concentration of air pollutants. Recent advances in sensors, IoT platforms, and mobile communications enable deploying low-cost mobile monitoring stations, e.g., by mounting sensors on vehicles. Examples include the air quality monitoring system using the public transport network in Zurich [1], the system using Google street-view cars in Oakland, CA [2], and imec's City-of-Things platform that uses postal trucks [3]. Deploying mobile stations increases the spatial density of air quality measurements; however, their temporal resolution per location is low since the vehicles are moving. In addition, there are still locations not covered by the vehicles. This renders computationally inferring missing air quality measurements across the spatial and temporal dimensions a problem of high interest.

A number of methods have been proposed to infer the air pollutant concentration using measurements collected by *fixed monitoring stations*. They are based on either *physical models* or *data-driven solutions* [4]. In the former approach, the complex physical dispersion processes of air pollutants are modeled using observed data and empirical assumptions [5, 6, 7]. Methods in this category, however, often require the availability of additional information, e.g., the distribution of pollution sources and accurate weather models [8]. Furthermore, the assumptions behind them might not hold given the variability of urban landscapes [4]. Data-driven methods do not rely on strong assumptions; instead, they utilize diverse local data, such as meteorological information, points of interest and traffic information, to infer the concentration of air pollutants. By leveraging the recent advances in deep learning, in particular, data-driven methods have achieved good inference performance [4, 9, 10].

Only very limited work has focused on air quality inference using data collected by mobile stations [1]. In this paper, we use the City-of-Things platform from imec [3] to retrieve street-level air quality data measured using mobile stations in Antwerp, Belgium. Given the available data, we infer the air quality in unmeasured locations across time and space. We follow a data-driven approach and formulate the air quality inference problem as a graph-based matrix completion problem. Specifically, we exploit the topology of Antwerp's street network and propose a novel deep learning model based on variational graph autoencoders; we refer to our model as AVGAE. The model captures effectively the spatio-temporal dependencies in the measurements, without using other types of data, such as traffic or weather, apart from the street-network topology. Experiments on real data from the City-of-Things platform show that our method outperforms various reference models.

To summarize, our main contributions in this paper are: (i) we formulate air quality inference as a graph-based matrix completion problem and propose a variational graph autoencoder for accurate inference. To the best of our knowledge, this is the first work to explore graph-based neural network models in the context of air quality inference; (ii) the proposed model effectively incorporates the temporal and spatial correlations via a temporal smoothness constraint and graph convolutional operations; (iii) we carry out comprehensive experiments on real-world datasets to evaluate the proposed model showing its superior performance compared to existing models.

The rest of this paper is as follows: Section 2 reviews the related work, and Section 3 states the problem and presents our model. Section 4 describes the experiments and Section 5 concludes the paper.

2. RELATED WORK

2.1. Air Quality Inference

Unmeasured air pollution in locations or time instances can be estimated using simple interpolation or resampling techniques [10, 11]. However, given the dynamics of air pollutants, these techniques tend to produce high estimation errors. Alternatively, one can use kriging-based variogram models to capture the variance in air pollution data with respect to the geodesic distance [12, 13]. As a purely spatial interpolation method, however, this approach does not capture the temporal correlation in the air quality data.

In recent years, we have witnessed the rise of machine-learning-based methods. In [14], a co-training approach with temporal and spatial classifiers is proposed for classifying discrete air quality indices (AQIs); yet, this model can not be used to infer the real-valued concentration of air pollutants. Deep-neural-network-based models have been proposed for air quality inference in [4, 9]. These models exploit the spatio-temporal correlations in the concentration of air pollutants either by incorporating additional information in the model—from traffic, weather, etc.—or by imposing objective constraints. Unlike these methods, our work utilizes a graph variational autoencoder to estimate the concentration of air pollutants across space and time, and provides higher estimation performance without considering additional information. Alternatively, the authors of [15] proposed a model to infer the air quality using a graph-based semi-supervised approach. The work considers an affinity graph of locations and deploys a label propagation mechanism to predict the air quality. This work is similar to ours in terms of formulating the air quality inference problem on graphs; however, instead of label propagation, we propose an end-to-end graph convolutional model, which is more flexible. It is worth noting that in [9, 14, 15] the considered area is divided in a uniform grid, whereas in the proposed approach we aggregate measurements non-uniformly across the street network (see Section 3.1)

2.2. Matrix Completion on Graphs

Matrix completion is a fundamental problem in machine learning, which focuses on inferring unknown entries of matrices [16]. Applications of matrix completion include recommender systems [17], cellular network planning [18] and air quality inference [19], to name a few. Recently, a number of studies have addressed the problem of matrix completion with tools from graph signal processing [20, 21, 22, 23, 24] with applications in recommender systems. Our method is related to these approaches but it includes specific components tailored to the problem of air quality inference from mobile measurements. In the experimental section, we compare the performance of our method against [17, 20] and demonstrate its superior performance in inferring air quality data.

2.3. Variational Graph Autoencoders

Variational autoencoders (VAEs) [25] are generative models that have lately received considerable attention. The study in [26] proposed a VAE with fully connected neural network layers with application in collaborative filtering, a particular application of matrix completion. Furthermore, variational inference on graphs has been proposed for link prediction [27]. Our model is different from [27, 26] in that we propose a variational graph autoencoder, which can express the spatial and temporal dependencies across air pollution measurements. Furthermore, the data in [26] is assumed to follow a discrete multinomial distribution, whereas in our model, the air pollution measurements follow a continuous distribution.

3. METHOD

3.1. Problem Formulation and Notation

We focus on air quality inference at the street network of urban areas—namely, we consider only locations on streets—using measurements on the concentration of air pollutants collected by sensor-equipped vehicles moving around a specific urban area; the problem statement adheres to the smart cities concept. Each vehicle makes measurements while moving on the city street network, resulting in high spatial measurement density; in contrast, the measurements at a specific location have low temporal resolution.

As the time and location associated to a measurement are continuous, it is convenient to aggregate the measurements at discrete time instances and locations. We uniformly divide the time span of the data into equal slots of duration τ (e.g., one hour). In a given timeslot t , we gather all measurements within a pre-defined geographical distance r from a given spatial location p on the street network and take their median-value as the measurement at location p at timeslot t . The street network information is obtained from OpenMapTiles¹. Hence, the aggregation across space is non-uniform and is adapted to the considered locations on the street network.

The above aggregation process results in a measurement matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, with N the number of considered geographical locations and T the number of timeslots. An entry \mathbf{X}_{ij} , with $i = 1, \dots, N$ and $j = 1, \dots, T$, corresponds to the measurements at the i^{th} location and the j^{th} timeslot. \mathbf{X} is a highly incomplete matrix with the set of known entries denoted by Ω . Our task is to predict the air pollution concentration values in the unknown entries using the measurements (known entries) and the street-network topology.

For notational consistency, in the rest of the paper, we use bold-faced uppercase letters for matrices, bold-faced lowercase letters for vectors and regular lowercase letters for scalar variables. Both regular uppercase and lowercase Greek letters denote constants.

3.2. Variational Autoencoders

VAEs build on the assumption that the data points in a dataset can be drawn from a distribution conditioned by latent variables; furthermore, the latent variables follow a prior distribution, e.g., the Gaussian distribution. VAEs attempt to learn a deterministic function that transforms the Gaussian distribution to the distribution of the observed data.

Let \mathbf{x} denote an example in the dataset and \mathbf{z} the vector containing the latent variables. The inference process is modelled by

$$q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}), \quad (1)$$

where $\boldsymbol{\mu} = f_{\boldsymbol{\mu}}(\mathbf{x}, \boldsymbol{\Theta}_1)$ and $\boldsymbol{\sigma} = f_{\boldsymbol{\sigma}}(\mathbf{x}, \boldsymbol{\Theta}_2)$ are parameters of the Gaussian distribution. The generative process is characterized by

$$p(\mathbf{x}|\mathbf{z}) \propto f_z(\mathbf{z}, \boldsymbol{\Phi}). \quad (2)$$

It should be noted that $f_{\boldsymbol{\mu}}$, $f_{\boldsymbol{\sigma}}$ and f_z are parameterized functions and their parameters $\boldsymbol{\Theta}_1$, $\boldsymbol{\Theta}_2$ and $\boldsymbol{\Phi}$ can be learned from data. These functions are often implemented by neural network layers. To find the parameters, one needs to minimize the following equation:

$$\mathcal{L} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z})] + \mathcal{D}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]. \quad (3)$$

In (3), one can interpret the first term as the reconstruction error and the second term as a regularization constraint. The second term is the Kullback-Leibler (KL) divergence between $q(\mathbf{z}|\mathbf{x})$ and

¹<https://openmaptiles.com/downloads/europe/belgium/antwerp/>

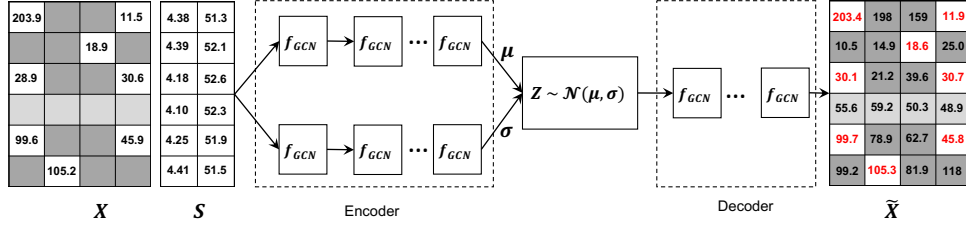


Fig. 1. The proposed variational graph autoencoder architecture for air quality inference (AVGAE). The input of AVGAE consists of the incomplete matrix \mathbf{X} and the matrix of geocoordinates \mathbf{S} . The light gray row in \mathbf{X} indicates a location without measurements across time, dark gray cells represent unmeasured locations at a given time instance, and the entries with a red font are reconstructed known entries on which we evaluate the loss function. The function blocks f_{GCN} represent GCN layers. The encoder outputs the parameters μ, σ of a Gaussian distribution. The output matrix $\tilde{\mathbf{X}}$ approximates the known entries and contains the inferred unknown entries.

the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, which can be computed with a closed form formula [25].

3.3. Variational Graph Autoencoders

Variational graph autoencoders (VGAEs) [27] adhere to the VAE concept and utilize graph convolutional layers (GCN) for the parameterized functions f_μ, f_σ and f_z . Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and a degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$, $N = |\mathcal{V}|$, a graph convolutional layer [28] is expressed as

$$f_{GCN}(\mathbf{X}) = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{W}) \quad (4)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $\tilde{\mathbf{D}}_{ij} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{X} \in \mathbb{R}^{N \times T}$ is the input signal summarized in a matrix, $\mathbf{W} \in \mathbb{R}^{T \times D}$ is the corresponding weight matrix with D being the GCN layer's dimensionality, and σ indicates a nonlinear function. By stacking multiple GCN layers, more complex functions can be constructed. In what follows, we propose a particular architecture tailored to the air quality inference task.

3.4. The Proposed AVGAE Architecture

The architecture of our model, which we refer to as AVGAE, is depicted in Fig. 1. We build a graph of N nodes by considering the geodesic distance among the N corresponding discretized locations on the street network. Two nodes are connected if the geodesic distance between them is smaller than a predefined threshold δ , or if they belong to the same road segment. The weight of a connection is the inverse of the geodesic distance in meters computed by the Haversine formula [29]. Furthermore, we summarize the locations' geocoordinates in a matrix \mathbf{S} . Our model is described by the following set of equations:

$$\mu = \text{GCN}_\mu(\mathbf{X}, \mathbf{S}, \Theta_1) \quad (5)$$

$$\sigma = \text{GCN}_\sigma(\mathbf{X}, \mathbf{S}, \Theta_2) \quad (6)$$

$$\mathbf{Z} \sim \mathcal{N}(\mu, \sigma) \quad (7)$$

$$\tilde{\mathbf{X}} = \text{GCN}_z(\mathbf{Z}, \Phi) \quad (8)$$

In (5), (6), (8), $\text{GCN}_\mu, \text{GCN}_\sigma$ and GCN_z are functions obtained by stacking GCN layers and Θ_1, Θ_2 and Φ are parameters that can be learned from the data. We concatenate the geocoordinate matrix \mathbf{S} with the measurement matrix \mathbf{X} horizontally, making \mathbf{S} a part of the input of our model. By doing so, we aim at exploiting the correlation between the measurements and the geocoordinates. Our

model utilizes two separate branches for training μ and σ , thereby allowing to select proper activation functions for μ and σ (the selected functions are mentioned in Section 4).

It is worth mentioning that our model is capable of inferring values at locations that are not measured by vehicles, which are illustrated by an empty row in matrix \mathbf{X} in Fig. 1. This is because the proposed model captures the spatial correlation between the unobserved and observed locations through their geocoordinates and the street network's topology.

The loss function of our model is defined in (9). We modify (3) by using the mean absolute error (MAE) regularized by a KL divergence term. Even though the MAE is not everywhere differentiable, we find that using its sub-gradient is sufficient for optimization with gradient descent. The temporal dependency between measurements imposes an additional smoothness constraint:

$$\mathcal{L}(\mathbf{X}, \Theta_1, \Theta_2, \Phi) = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} |\tilde{\mathbf{X}}_{ij} - \mathbf{X}_{ij}| + \beta \mathcal{D}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] + \gamma \sum_{(i,j)} \sum_{k \in \mathcal{T}(i,j)} e^{-|j-k|} (\tilde{\mathbf{X}}_{ij} - \tilde{\mathbf{X}}_{i,k})^2 \quad (9)$$

In (9), β and γ are positive tuning parameters and $\mathcal{T}(i, j)$ is the neighborhood of the entry $\mathbf{X}_{i,j}$ with respect to the temporal dimension. The width of the neighborhood $w_{\mathcal{T}}$ is a parameter that is fine-tuned experimentally. We minimize the loss function with respect to training entries using the stochastic gradient descent—where we use the reparameterization technique in [25]—and we deploy the dropout regularization technique to mitigate overfitting. After training, we obtain the re-constructed data matrix $\tilde{\mathbf{X}}$ containing predicted values for the unknown entries.

4. EXPERIMENTS

4.1. The Dataset

We rely on the City-of-Things platform [3] to obtain measurements of the air quality in the Antwerp city in Belgium. The platform makes use of 24 cars equipped with mobile monitoring devices. We retrieve the measurements during May 2018 for two air pollutants, that is, NO_2 and $\text{PM}_{2.5}$.

As described in Section 3.1, we first apply aggregation as a data preprocessing step, where we choose $\tau = 1$ hour and $r = 100$ meters. After processing, we obtain 3630 and 4086 discrete locations

Table 1. Air quality inference results.

	NO ₂		PM _{2.5}	
	MAE	RMSE	MAE	RMSE
Kriging linear [12]	18.19	28.43	3.28	7.98
Kriging exponential [12]	15.86	25.58	2.89	7.43
KNN-based collaborative filtering [30]	20.92	32.67	3.60	7.47
SVD [31]	27.35	38.32	7.41	13.40
NMF [32]	71.67	82.34	6.75	13.09
NMC [17]	22.12	32.83	3.99	8.35
RGCNN [20]	48.6	60.11	6.2	15.4
AVGAE (Our method)	14.92	24.33	2.56	6.42

Table 2. The description of the NO₂ and PM_{2.5} dataset. The units for NO₂ and PM_{2.5} are parts per billion (ppb) and $\mu\text{G}/\text{m}^3$.

	NO ₂	PM _{2.5}
Number of locations	3630	4086
Duration in hours	720	720
Max concentration	633.65	189.03
Min concentration	0.16	0.07
Mean concentration	85.50	9.83
% of known entries versus all	0.60	0.56

for the NO₂ and PM_{2.5} datasets, respectively. Each location is specified by the pair of latitude and longitude geocoordinates. Moreover, for each pollutant, a location is associated with a measurement vector of $T = 30 \times 24 = 720$ dimensions, which is the number of hours during the considered period. The description of the dataset is presented in Table 2.

4.2. Experimental Setting

To evaluate the proposed method, we randomly divide the known entries into training and test sets. That is, 90% of the known entries is used for training and the rest is reserved for testing. We use two common evaluation metrics, namely, the root mean squared error (RMSE) and the mean absolute error (MAE). To obtain robust results, we repeat this procedure with 5 random divisions and report average results.

To create the graph, we set the distance threshold to $\delta = 200$ m. The parameters of the AVGAE are chosen experimentally: we set the learning rate to $\alpha = 0.005$, the KL divergence coefficient to $\beta = 0.1$, the temporal smoothness coefficient to $\gamma = 0.8$, the temporal neighborhood width to $w_T = 3$ and the dropout rate to 0.4. For all GCN layers, we use the same dimensionality, that is, $D = 512$. We use 4 GCN layers for the encoder and 1 GCN layer for the decoder. We employ ReLU to activate the GCN layers of the encoder except for the last GCN layer of the σ branch where the sigmoid function is used because σ should contain strictly positive entries. Because the output is unbounded, it is not necessary to use an activation function for the GCN layer of the decoder.

As reference benchmarks, we have selected two well-established kriging-based models, that is, the linear and exponential models [12]. A kriging model is applied per column of the matrix \mathbf{X} (corresponding to a timeslot) using the geocoordinates information in \mathbf{S} . Furthermore, we consider various state-of-the-art matrix completion methods, including KNN-based collaborative filtering [30], SVD-based matrix completion [31], non-negative matrix factoriza-

tion [32], and extendable neural matrix completion [17]. These models perform completion under an assumption on \mathbf{X} , e.g., a low-rank prior. Furthermore, we compare against the graph-based matrix completion method in [20]; specifically, the RGCNN model, where the graph for the row-factor matrix is the same as in our AVGAE model and the hyper-parameters are kept as in [20]. For the implementation, we rely on PyKridge² for the kriging models and Surprise³ for the reference matrix completion techniques. The implementations of [17, 20] are available online. All models have been trained in our dataset.

4.3. Result and Analysis

The results in air quality inference with the different methods are shown in Table 1. Kriging-based methods provide good estimation accuracy, particularly the exponential model. This is because such models capture properly the spatial correlation in the air quality measurements with respect to the geodesic distance.

On the other hand, matrix completion models assume that there are hidden factors characterizing rows (a.k.a., discrete locations) and columns (a.k.a., timeslots). While this assumption is appropriate for other problems such as recommendation systems, it does not properly capture the spatio-temporal correlation in the concentration of air pollutants.

It is evident that our AVGAE model achieves the best performance for both the RMSE and MAE metrics and for both pollutants (NO₂ and PM_{2.5}). Conversely to kriging models, AVGAE effectively captures *both* the temporal and spatial correlations in the data, and leverages the underlying graph structure of the street network. Furthermore, unlike the reference matrix completion models, either graph-based or not, AVGAE adheres to an autoencoder model, which provides good performance in reconstruction problems.

5. CONCLUSION

Measuring the concentration of air pollutants with mobile stations is a promising approach to achieve hyperlocal air quality monitoring. The measurements collected by such mobile stations, however, have very low temporal resolution per location and there are still unmeasured locations. We formulated the air quality inference problem in this setting as a matrix completion problem on graphs, and proposed a variational graph autoencoder model to solve it. The proposed model was experimentally shown to effectively capture the spatio-temporal correlation in the measurements, resulting in better air quality inference compared to various state-of-the-art kriging and matrix completion methods.

²<https://pykrige.readthedocs.io/en/latest/index.html>

³<https://surprise.readthedocs.io/en/stable/index.html>

6. REFERENCES

- [1] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *IEEE International Conference on Pervasive Computing and Communications*, 2014, pp. 69–77.
- [2] J. S. Apte, K. P. Messier, S. Gani, M. Brauer, T. W. Kirchstetter, M. M. Lunden, J. D. Marshall, C. J. Portier, R. CH. Vermeulen, and S. P. Hamburg, "High-resolution air pollution mapping with google street view cars: exploiting big data," *Environmental Science & Technology*, vol. 51, pp. 6999–7008, 2017.
- [3] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. De-meester, "City of things: An integrated and multi-technology testbed for iot smart city experiments," in *IEEE International Conference on Smart Cities*, 2016, pp. 1–8.
- [4] W. Cheng, Y. Shen, Yanmin Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," in *AAAI Conference on Artificial Intelligence*, 2018.
- [5] M. J. Kim, R. J. Park, and J. J. Kim, "Urban air quality modeling with full o₃-no_x-voc chemistry: Implications for o₃ and pm air quality in a street canyon," *Atmospheric Environment*, vol. 47, pp. 330–340, 2012.
- [6] N. K. Arystanbekova, "Application of gaussian plume models for air pollution simulation at instantaneous emissions," *Mathematics and Computers in Simulation*, vol. 67, pp. 451–458, 2004.
- [7] C. Mensink, K. D. Ridder, N. Lewyckij, F. Lefebvre, L. Janssen, J. Cornelis, S. Adriaensen, and M. Ruts, "Aurora: an air quality model for urban regions using an optimal resolution approach," *WIT Transactions on Ecology and the Environment*, vol. 51, 1970.
- [8] T. Godish, W. T. Davis, and J. S. Fu, *Air Quality*, CRC Press, 5th edition, 2015.
- [9] Z. Qi, T. Wang, G. Song, W. Hu, X. Li, and Z. M. Zhang, "Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
- [10] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, and S. Lin, "A spatiotemporal prediction framework for air pollution based on deep rnn," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 15, 2017.
- [11] L. Li, X. Zhang, J. B. Holt, J. Tian, and R. Piltner, "Spatiotemporal interpolation methods for air pollution exposure," in *SARA*, 2011.
- [12] P. K. Kitanidis, *Introduction to geostatistics: applications in hydrogeology*, Cambridge University Press, 1997.
- [13] X. Xie, I. Semanjski, S. Gautama, E. Tsiligianni, N. Deligiannis, R. T. Rajan, F. Pasveer, and W. Philips, "A review of urban air pollution monitoring and exposure assessment methods," *ISPRS International Journal of Geo-Information*, vol. 6, pp. 389, 2017.
- [14] Y. Zheng, F. Liu, and H. P. Hsieh, "U-air: When urban air quality inference meets big data," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1436–1444.
- [15] H. P. Hsieh, S. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 437–446.
- [16] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, pp. 608–622, 2016.
- [17] D. M. Nguyen, E. Tsiligianni, and N. Deligiannis, "Extendable neural matrix completion," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.
- [18] S. Chouvardas, S. Valentin, M. Draief, and M. Leconte, "A method to reconstruct coverage loss maps based on matrix completion and adaptive sampling," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 6390–6394.
- [19] Y. Yu, J. JQ. Yu, V. OK. Li, and J. CK. Lam, "Low-rank singular value thresholding for recovering missing air quality data," in *IEEE International Conference on Big Data*, 2017, pp. 508–513.
- [20] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 3697–3707.
- [21] R. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *stat*, vol. 1050, pp. 7, 2017.
- [22] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandenberghenst, "Matrix completion on graphs," *arXiv preprint arXiv:1408.1717*, 2014.
- [23] W. Huang, A. G. Marques, and A. Ribeiro, "Matrix completion as graph bandlimited reconstruction," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4039–4043.
- [24] W. Huang, A. G. Marques, and A. R. Ribeiro, "Rating prediction via graph signal processing," *IEEE Transactions on Signal Processing*, vol. 66, pp. 5066–5081, 2018.
- [25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [26] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *World Wide Web Conference*, 2018, pp. 689–698.
- [27] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *NIPS Workshop on Bayesian Deep Learning*, 2016.
- [28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.
- [29] G. V. Brummelen, *Heavenly mathematics: The forgotten art of spherical trigonometry*, Princeton University Press, 2012.
- [30] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, pp. 1, 2010.
- [31] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.
- [32] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1273–1284, 2014.