Quality-Aware Graph Learning Regularization For Heterogeneous Air Quality Sensor Networks

Pau Ferrer-Cid
Universitat Politecnica de Catalunya, Spain
pau.ferrer.cid@upc.edu

Jose M. Barcelo-Ordinas
Universitat Politecnica de Catalunya, Spain
jose.maria.barcelo@upc.edu

Jorge Garcia-Vidal
Universitat Politecnica de Catalunya, Spain
jorge.garcia@upc.edu

Abstract

In recent years, the graph signal processing (GSP) field has brought signal processing techniques to numerous areas. Among them, graphs have been used for various applications in sensor networks for air quality monitoring. One of the main tasks consists of learning the graph that describes the relationships between sensors in a network. Although many data-driven graph learning techniques exist, the heterogeneous nature of air quality low-cost sensor networks, where low-cost sensors and high-precision instruments coexist, imposes the need to include information about the quality of the sensors used. Therefore, in this paper, we propose a graph learning regularization framework that allows for taking into account the reliability of the different nodes of an IoT network, in what we call *quality-aware graph learning regularization (QAGLR)*. The regularization framework is evaluated for signal reconstruction and it is also assessed in the case of the creation of GSP-based virtual sensors, showing the benefits of introducing information about sensors' reliability.

Categories and Subject Descriptors

H.3.3.3 [**Information Systems**]: Sensor Networks

General Terms

Sensor networks, graph signal processing, virtual sensors.

Keywords

Air quality, low-cost sensors, graph learning, regularization.

1 Introduction

The field of graph signal processing (GSP) has opened up the possibility of applying classical signal processing techniques, e.g., signal filtering, signal reconstruction, or spectral analysis, to signals defined on an irregular domain such as graphs [23, 24, 25]. In this way, signal processing techniques on graphs have been developed and many applications have appeared due to the flexibility of representation that graphs have in fields like sensor networks, biological data, image analysis, or machine learning [25]. Moreover, the use of graphs allows for exploiting the implicit structure of the data as well as improving the interpretability of the techniques applied to these data lying on an irregular domain [9]. Other examples of graph-based techniques are graph neural networks (GNN), which are a generalization of neural networks on graphs [26].

A key component for the application of the GSP to a domain is the network topology that describes the relationships between the different graph nodes. In GSP, different matrices have been used to describe the topology of a graph, e.g., the adjacency matrix, the weight matrix, the Laplacian matrix, or more generally the graph shift operator (GSO) [20, 8]. Therefore, much research has been focused on learning the graph that describes a specific data set. There exist different approaches to finding the graph that best describes a data set, among them we can find approaches based on statistics (e.g., graphical Lasso), approaches based on GSP, where a graph can be defined from the geodesic distances between nodes or from the notion of signal smoothness, as well as graph learning (GL) methods based on diffusion processes [20, 8]. The notion of signal smoothness has been widely used to develop data-driven GL methods, where one tries to obtain a topology over which the graph signals are smooth [7, 10, 3, 17]. A network signal is considered to be smooth if strongly connected nodes have similar values. For instance, Dong *et al.* [7] defined a GL model based on signal smoothness to learn the Laplacian matrix from the data, resulting in an alternate convex optimization model that is solved iteratively. Similarly, Kalofolias [17] defined a GL model based on signal smoothness to learn the graph weight matrix.

The network topology can be applied in conjunction with GSP and machine learning techniques for a large number of applications in the field of wireless sensor networks (WSN) [18]. Jablonski *et al.* [15] applied a graph learned from data to a Polish tropospheric ozone (O_3) sensor network to perform clustering tasks. Likewise, Do *et al.* [6] developed a matrix completion model based on GNNs to impute sensor values in an air quality sensor network. Moreover, data reconstruction methods have been applied to air quality sensor networks using GSP [16, 14, 12, 11]. Other applications of the graph structure and machine learning techniques have also appeared, such as the detection of malfunctioning low-

cost air quality sensors [13].

In short, we highlight how graph-based applications have gained interest in recent years in the field of sensor networks and, in particular, in air quality sensor networks. Nevertheless, these air quality sensor networks have special characteristics, namely, they are heterogeneous, i.e., composed of high-precision sensors and low-cost sensors (LCSs) [22]. LCSs are known for their long-term accuracy issues and reliability issues, so the main focus of study in relation to LCSs has been the improvement of the quality of LCS data [5, 21, 19]. It must therefore be taken into account that LCSs in a heterogeneous air quality monitoring network are prone to errors and long-term failures. Hence, GL techniques need to be adapted to this environment of higher and lower reliability nodes by encouraging connections between LCSs and high-precision sensors. This is important because many GSP techniques rely on the topology of the network, and if the relationships between sensors change due to the deterioration of some of them, the performance of such applications may also worsen. Similarly, in the sensor placement problem, modifications have been made to take into account nodes of higher or lower cost and reliability [4].

In this paper, we propose the *quality-aware graph learning regularization* (QAGLR) framework to learn the topology of a heterogeneous air quality sensor network by introducing prior information about the quality of the sensors. We present the framework as a generalization of widely known GL models. In addition, we present a practical example of the framework applied to the creation of GSP-based virtual sensors and analyze the robustness provided by QAGLR in the case where sensors in the network deteriorate.

2 Quality-aware GL regularization

In this section, we introduce the quality-aware graph learning regularization (QAGLR) framework as well as we link it to well-known GL techniques.

Definition 1. We define a graph $\mathcal{G}$ as the triplet $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{S}\}$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of graph nodes, $\mathcal{E} = \{(i,j) : S_{ij} \neq 0\}$ is the set of edges, and $\mathbf{S} \in \mathbb{R}^{N \times N}$ is the graph shift operator (GSO).

The GSO [23], $\mathbf{S}$, can be used to describe the existing relationships between the graph nodes representing the different sensors in a heterogeneous LCS IoT network. For this specific case, it makes sense to assume that the graph edges are undirected given that the similarity between sensors is reciprocal, meaning that the matrix $\mathbf{S}$ is symmetric $\mathbf{S} = \mathbf{S}^\mathsf{T}$.

Definition 2. A graph signal $\mathbf{x} \in \mathbb{R}^N$ can be seen as the map $x: \mathcal{V} \to \mathbb{R}$ that maps a graph node i to the measurement recorded by the i-th sensor at a given time, $x_i \in \mathbb{R}$.

2.1 General GL framework

Among the existing GL methods, we focus on those that use sensor network data to learn a data-driven graph. This approach has already proven to be a good choice for air quality LCS networks [11]. Accordingly, we can define a general data-driven GL framework to learn the GSO matrix $\mathbf{S}$ using a set of graph signals, i.e., network measurements, $\mathbf{X} \in \mathbb{R}^{N \times N_s}$, where $N_s \in \mathbb{N}$ is the number of measurements and therefore the number of graph signals, as:

$$\begin{aligned} \min_{\mathbf{S} \in \mathbb{R}^{N \times N}} \quad & F(\mathbf{X}, \mathbf{S}) + \lambda \cdot R(\mathbf{S}) \\ s.t. \quad & \mathbf{S} \in \mathcal{S} \end{aligned} \tag{1}$$

Where the function $F(\cdot)$ takes as input a set of graph signals $\mathbf{X}$ and a GSO matrix $\mathbf{S}$ and evaluates the goodness-of-fit of the GSO with respect to the observed sensor network measurements $\mathbf{X}$. The second part of the objective function corresponds to function $R(\cdot)$ which acts as regularizer of the solution $\mathbf{S}$ with controlling hyperparameter $\lambda \in \mathbb{R}$. This regularization is usually associated with the complexity and connectivity of the resulting graph, e.g., the sparsity of the resulting matrix $\mathbf{S}$. The constraint in eq. (1) forces the shift matrix to belong to the set of valid GSO matrices $\mathcal{S}$. The conditions for the matrix $\mathbf{S}$ to be valid depend on the choice of GSO. There are different options for the GSO matrix $\mathbf{S}$, and among them, we find the adjacency matrix $\mathbf{A}$, the weight matrix $\mathbf{W}$, or the Laplacian matrix $\mathbf{L}$. All three matrices have been studied in the GSP field [23, 24]. Henceforth, we focus on learning the Laplacian matrix $\mathbf{L}$, given its use as GSO in the GSP field, where its eigendecomposition is used to obtain the graph Fourier bases [24]. Thus, we can reformulate the framework in eq. (1) as:

$$\begin{aligned} \min_{\mathbf{L} \in \mathbb{R}^{N \times N}} \quad & F(\mathbf{X}, \mathbf{L}) + \lambda \cdot R(\mathbf{L}) \\ s.t. \quad & L_{ij} = L_{ji},\ L_{ij} \leq 0 \quad , 1 \leq i \neq j \leq N \\ & \mathbf{L}1 = 0 \end{aligned} \tag{2}$$

The different constraints force the resulting Laplacian matrix $\mathbf{L}$ to be symmetric and valid. There are different criteria for the function $F(\cdot)$, which evaluates the goodness-of-fit of the resulting graph $\mathcal{G}$. The graph signal smoothness is a criterion widely used in GL tasks [7, 17, 3]. Intuitively a signal is said to be smooth with respect to the graph if sensors with similar measurements are strongly connected, and conversely, sensors that are not similar are weakly connected or disconnected. Thus, this criterion results in finding a Laplacian matrix $\mathbf{L}$ that is coherent with the observed signals $\mathbf{X}$.

Definition 3. The smoothness of a graph signal can be evaluated through the total variation (TV), also known as Laplacian quadratic form or Dirichlet energy: $\mathrm{TV}(\mathbf{x}, \mathbf{L}) = \mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}$

Regarding the regularization term $R(\mathbf{L})$, different metrics can be used to enforce a level of connectivity in the graph. For instance, the Frobenius norm ($\|\mathbf{L}\|_\mathsf{F}$) or the $l_{p,q}$-matrix norm ($\|\mathbf{L}\|_{p,q}$) can be used to force a connectivity level jointly with the signal smoothness criterion.

2.2 QAGLR framework

Right now, we can define the QAGLR[1] framework to take into account the heterogeneous nature of air quality sensor networks. To this end, quality-aware knowledge about the different types of sensors that make up the network, e.g., LCSs or high-precision nodes, can be introduced to force low-quality sensors to have higher-quality sensors as neighbors. We can add a new regularization term $P(\cdot)$ to intro-

[1] A python implementation of the QAGLR using the CVXPY solver is available at https://bitbucket.org/sans-rg/ewsn-qaglr/.

duce this prior information through a quality-aware matrix $\mathbf{P}\in\mathbb{R}^{N\times N}$ so that we can define the QAGLR as:

$$\begin{aligned}\min_{\mathbf{S}\in\mathbb{R}^{N\times N}} \quad & F(\mathbf{X},\mathbf{S})+\lambda\cdot R(\mathbf{S})+\gamma\cdot P(\mathbf{S},\mathbf{P}) \\ s.t. \quad & \mathbf{S}\in\mathcal{S}\end{aligned} \tag{3}$$

Where $\gamma\in\mathbb{R}$ is the hyperparameter that controls the importance of the quality-aware information introduced by $\mathbf{P}$ in the objective function of the GL problem. Now, a good choice for the quality-aware regularization is $P(\mathbf{S},\mathbf{P})=\|\mathbf{S}\odot\mathbf{P}\|_{1,1}$, where $\odot$ is the Hadamard product. The connection between sensors is penalized according to their reliability defined by the weights P_{ij}, resulting in a weighted $l_{1,1}$-norm of matrix $\mathbf{S}$. There exist different ways to define the quality-aware matrix $\mathbf{P}$, we can define it as:

$$P_{ij}=\begin{cases}\frac{1}{1-(P_i\cdot P_j)} & \text{if } i\neq j \\ 1 & \text{otherwise}\end{cases} \tag{4}$$

Where $0\leq P_i<1$ denotes the probability of failure of the i-th sensor. For instance, in a heterogeneous LCS network with two types of nodes, high-precision instrumentation $\mathcal{V}_1$ and LCSs $\mathcal{V}_2$, we can set $P_{\mathcal{V}_1}=0$ and $P_{\mathcal{V}_2}=0.9$[2]. We can assign large P_i to LCSs to penalize more. All in all, applying the quality-aware regularization to the GL model defined by Dong *et al.* [7] results in the following optimization model:

$$\begin{aligned}\min_{\mathbf{L}\in\mathbb{R}^{N\times N}} \quad & \alpha\cdot\frac{1}{N_s}\mathrm{tr}(\mathbf{X}^\mathsf{T}\mathbf{L}\mathbf{X})+\lambda\cdot\|\mathbf{L}\|_\mathsf{F}^2+\gamma\cdot\|\mathbf{L}\odot\mathbf{P}\|_{1,1} \\ s.t. \quad & tr(\mathbf{L})=N \\ & L_{ij}=L_{ji},\ L_{ij}\leq 0 \ \ ,1\leq i\neq j\leq N \\ & \mathbf{L}1=0\end{aligned} \tag{5}$$

The above optimization problem is convex given that all the functions of the objective function are convex and the constraints are linear. All three terms control the sparsity of the resulting Laplacian matrix $\mathbf{L}$ while the first term promotes a smooth graph with respect to the training signals $\mathbf{X}$. These components are controlled by the hyperparameters $\{\alpha,\lambda\}$. The different constraints force the Laplacian to be valid, for more details on the constraints refer to [7].

In this particular case, the QAGLR optimization model shown in eq. (5) reduces to the model defined by Dong *et al.* [7] plus the quality-aware regularization. The Laplacian ($\mathbf{S}=\mathbf{L}$) is learned and the signal smoothness is used as a goodness-of-fit criterion of the graph ($F(\mathbf{X},\mathbf{L})=tr(\mathbf{X}^T\mathbf{L}\mathbf{X})$). In addition, the Frobenius norm of the Laplacian is used to control the density of the resulting graph ($R(\mathbf{L})=\|\mathbf{L}\|_\mathsf{F}^2$). Then, the quality-aware regularization is applied ($P(\mathbf{L},\mathbf{P})=\|\mathbf{L}\odot\mathbf{P}\|_{1,1}$) resulting in the optimization problem shown in eq. (5).

For illustrative purposes, we show how the QAGLR can be applied to another well-known GL method, the model defined by Kalofolias [17]. Here the weight matrix $\mathbf{S}$=$\mathbf{W}$ is learned as GSO, and the author also uses the notion of smoothness of the signal but using its form by means of the matrix $\mathbf{W}$, $F(\mathbf{X},\mathbf{W})=\|\mathbf{W}\odot\mathbf{Z}\|_{1,1}$, where $\mathbf{Z}\in\mathbb{R}^{N\times N}$ is the matrix of pairwise distances. Regarding the GSO regularization, the author forces the graph to have at least one neighbor per node and promotes the connectivity of the graph using the Frobenius norm, $R(\mathbf{W})=-\alpha\cdot 1^\mathsf{T}log(\mathbf{W}1)+\frac{\beta}{2}\|\mathbf{W}\|_\mathsf{F}^2$. Now, the quality-aware regularization defined in eq. (5) can be given in this case using the definition of the Laplacian matrix, $P(\mathbf{W},\mathbf{P})=\gamma\cdot\left(\|\mathbf{W}\odot\mathbf{P}\|_{1,1}+(\mathbf{W}1)^\mathsf{T}diag(\mathbf{P})\right)$, where $diag(\cdot)$ extracts the diagonal elements of a matrix. As it may be noticed, the new regularization term modifies the goodness-of-fit term $F(\cdot)$ by introducing weights based on the quality-aware information. The resulting optimization model is:

$$\begin{aligned}\min_{\mathbf{W}\in\mathbb{R}^{N\times N}} \quad & \underbrace{\|\mathbf{W}\odot\mathbf{Z}\|_{1,1}}_{F(\mathbf{X},\mathbf{W})}\underbrace{-\alpha\cdot 1^\mathsf{T}log(\mathbf{W}1)+\frac{\beta}{2}\|\mathbf{W}\|_\mathsf{F}^2}_{R(\mathbf{W})}+ \\ & \underbrace{\gamma\cdot\left(\|\mathbf{W}\odot\mathbf{P}\|_{1,1}+(\mathbf{W}1)^\mathsf{T}diag(\mathbf{P})\right)}_{P(\mathbf{W},\mathbf{P})} \\ s.t. \quad & \mathbf{W}\in\mathcal{W}\end{aligned} \tag{6}$$

3 Experimental evaluation

In this section, we experimentally evaluate the proposed framework using real data from a heterogeneous network of air quality sensors. First, we describe the data set used to carry out the different experiments. Secondly, we show how QAGLR works depending on the different hyperparameters. Then, we evaluate the performance of the quality-aware graph in the scenario where the network sensors may present different noise levels. And finally, we evaluate the use of a QAGLR in the case of creating virtual sensors using a nonlinear graph filter. We set the model defined by Dong *et al.* [7] as the baseline (which we will call "GL-Dong" from now on) and we compare the results with the QAGLR applied to Dong's model. We divide the data set into 50% for training and cross-validation (CV) and 50% for testing.

3.1 Heterogeneous low-cost air quality sensor network

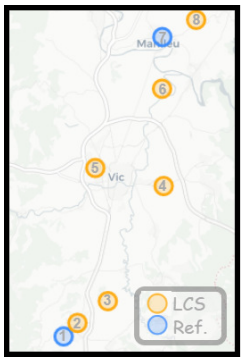
In order to experiment with the QAGLR on a heterogeneous network of air quality sensors, we use the data set obtained by the sensor network deployed by the H2020 Captor project. This sensor network has been previously used for research purposes and the data is openly available [1]. More precisely, it was deployed during the summer of 2017 in the Vic area (Spain) and it was composed of twenty-five IoT LCS nodes and three reference stations[3]. The nodes contained an Arduino Yun as processing unit, a modem 3G to provide connectivity to a central database, four metal-oxide ozone (O_3) SGX Sensortech MICS 2614 sensors, and a Grove temperature and relative humidity sensor per node.

Table 1. Data set metrics

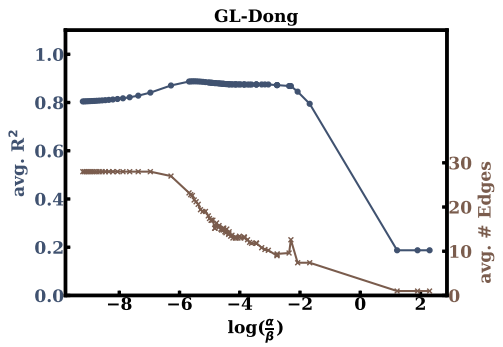
Nodes	# Network Measurements	Resolution	Avg. O^3
6 LCSs + 2 Ref.	2612	30 min	66.68 μg/m^3

[2]Note that this case, where there are two sets of sensors, high-precision and LCSs, is an example. Different sets of sensors of different quality can coexist in a network.

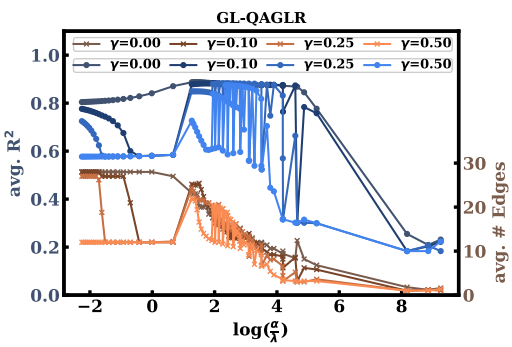
[3]Reference stations provide ground-truth measurements for different pollutants and their reliability is optimal and they act as high-cost sensors.



(a) Heterogeneous LCS network location and disposition.



(b) CV average R^2 and average number of edges for the GL-Dong model and different values for $\{\alpha,\beta\}$.



(c) CV average R^2 and average number of edges for the GL-QAGLR model with different values for $\{\alpha,\lambda,\gamma\}$.

Figure 1. On the left the heterogeneous LCS network and on the right the CV results for the GL models.

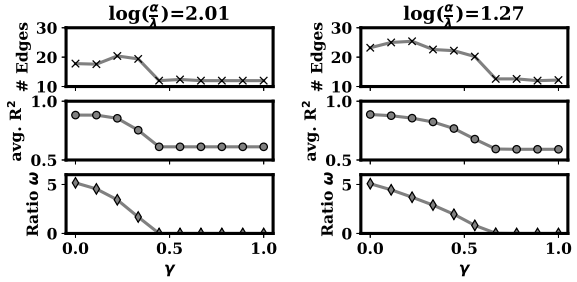
To perform the different experiments we focus on a set of eight nodes, six LCSs, which have been previously calibrated in-situ in nearby reference stations, and two reference stations (Table 1). Figure 1.a) depicts the locations of the different nodes forming the sensor network. This network allows for evaluating the QAGLR since it contains both LCSs and reference stations.

3.2 QAGLR: learning the graph

In this experiment, we evaluate how the QAGLR framework (applied to the model defined by Dong *et al.* [7]) behaves in comparison with the GL-Dong model. Then, by performing CV over the training, we perform a grid search over the hyperparameters $\{\alpha,\beta\}$ and $\{\alpha,\lambda,\gamma\}$ to assess the evolution of the resulting graphs in terms of the number of edges. To approximate the goodness-of-fit of the resulting Laplacian $\mathbf{L}$, we reconstruct the signals of each of the sensors through their neighboring sensors, $\mathcal{N}(i)=\{j: L_{ij}\neq 0\}$, using the Laplacian interpolated regularization, which reconstructs the signal by minimizing the TV [2]:

$$\hat{x}_i = -L_{ii}^{-1}\mathbf{L}_{i\mathcal{N}(i)}\mathbf{x}_{\mathcal{N}(i)}, \quad \forall i \in \mathcal{V} \tag{7}$$

Where $x_i \in \mathbb{R}$ and $\mathbf{x}_{\mathcal{N}(i)} \in \mathbb{R}^{|\mathcal{N}(i)|}$. Figures 1.b) and c) show the CV performance of GL-Dong and GL-QAGLR in terms of the average CV R^2 and the average number of edges. As seen in Figure 1.b), the GL-Dong model is completely defined by the ratio α/β of its hyperparameters, as the ratio increases, sparser graphs are obtained. Moreover, in the central part $log(\alpha/\beta)\in[-6,-2]$ we see how as α (which promotes the smoothness of the graph) gains importance, the average reconstruction improves, being able to obtain an average CV reconstruction R^2 of 0.88 with graphs with a wide variety of sparsity. Figure 2.c) shows the same results for the GL-QAGLR, as it can be seen, for γ=0 the behavior is exactly the same as GL-Dong since the GL model coincides with the one defined by Dong *et al.* [7]. Nevertheless, as γ increases, we observe how the results, both in terms of graph edges and reconstruction performance present a lot of variability. This is because now the ratio α/λ is not governing the GL behavior but the γ also influences the results. Indeed, the α and γ present a joint behavior since the introduction of quality-aware information penalizes the smoothness term. Figure 1.c) shows how introducing $\gamma \neq 0$ tends to reduce the reconstruction performance as well as the number of edges. In fact, larger γ values are associated with smaller reconstruction capabilities and sparser graphs given that a low number of high-precision instrumentation is available in this data set. This is because we give less importance to the smoothness term, which evaluates the goodness-of-fit, and we give importance to the quality-aware information, which means, as we will see in the following sections 3.3 and 3.4, that we sacrifice some performance in exchange for a more robust graph. For a small penalization, γ=0.10, the reconstruction performance is only worsened a little, but in some hyperparameters ratio range, e.g., [-1, 1], we observe how the penalization gains importance in the optimization and the result is a sparse graph with little reconstruction capabilities. It is worth noting that this particular behavior is linked to the sensor network and the design of the quality-aware matrix $\mathbf{P}$ but other values would produce coherent results with the penalizations given.



(a) Ratio $log(\alpha/\lambda)$=2.01. (b) Ratio $log(\alpha/\lambda)$=1.27.

Figure 2. Behavior of the QAGLR for a pair of α/λ ratios and different γ values.

Figures 2.a) and b) show a particular example of two fixed ratios α/λ and increasing γ values. As it can be seen, as γ increases, the reconstruction performance decreases given that

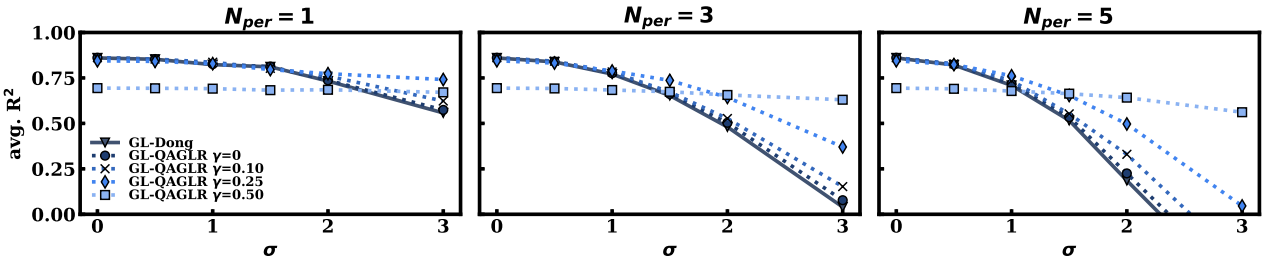


Figure 3. Average LCS reconstruction test R^2 for different number of perturbed sensors (N_{per}), different levels of noise (σ), different γ values, and ten repetitions for each level of noise.

the optimization model prioritizes the quality-aware information over the smoothness of the graph with respect to the data. Besides, the number of edges increases and then decreases given that new connections between LCSs and high-precision nodes may be added and some other connections between LCSs may be removed. Moreover, we observe how the impact of the γ value depends on the ratio α/λ as depicted by the slope of the decrease of reconstruction performance and the number of edges. Finally, in order to show the effect of the regularization, we have plotted what we call the ratio $\omega = \sum_{i\in\mathcal{V}_2} L_{ii}^{-1} \sum_{j\in\mathcal{V}_2, j\neq i} |L_{ij}|$ which represents the ratio between the Laplacian weights assigned to LCS-to-LCS connections and all weights assigned to all LCSs. As it is seen, as the γ value increases, less importance is given to LCS-to-LCS connections until ω=0 which represents that there are no connections between LCSs.

3.3 QAGLR: data reconstruction with noisy sensors

Now, once we have obtained the graphs that minimize the CV reconstruction R^2 (without regularization, γ=0), we evaluate how the regularization affects the case in which the LCSs may present noise. For this purpose, we test different $\gamma = \{0, \ldots, 0.50\}$ values and introduce additive independent white Gaussian noise of increasing variance $\varepsilon \sim N(0, \sigma^2)$ in a variable percentage of the LCSs during the testing. Therefore, we reconstruct each one of the network nodes when a variable percentage of sensors present errors. Due to the random nature of the perturbation we perform ten repetitions.

Figure 3 shows the average reconstruction R^2 when one, three, and five sensors have been perturbed with noise. When only one sensor is perturbed (N_{per}=1) it can be seen how there is little impact on the average reconstruction accuracy since one out of eight sensors may not influence the average reconstruction performance. Nevertheless, we can observe that for σ=2 there is a decrease in the R^2 and the larger the γ the less affected is the reconstruction, for γ=0.25 the average R^2 is 0.77 while GL-Dong obtains an average R^2 of 0.73. For the case N_{per}=3, the same pattern is observed but now the reconstruction performance is more affected by the noise. Again, the GL-QAGLR with γ=0.50 obtains a slightly worse performance than GL-Dong for $\sigma \leq 1.5$ but in return, when the error increases, i.e., for $\sigma > 1.5$, the reconstruction is not as bad as with the other GL method, R^2 of 0.67. For the extreme case, N_{per}=5, the same patterns can be observed although in this case the average reconstruction R^2 is much more affected by the introduced noise.

3.4 QAGLR: virtual sensing

In this last experiment, we present a scenario where the graph is learned so that it can be fed to a graph filter to reconstruct the signal of a subset of sensors, thus creating a set of virtual sensors. We use a graph filter based on the Volterra series of third order [27]. Consequently, we select two sensors of the network (sensors 5 and 8 in Figure 1.a)) and we perform CV over the training and graph hyperparameters to find the best graph that coupled with the graph filter obtains the lowest CV reconstruction R^2 for the virtual sensors. Afterwards, during the testing, we add additive white Gaussian noise of increasing variance $\varepsilon \sim N(0, \sigma^2)$ to all the LCSs and perform ten repetitions. We can define the graph filter virtual sensing model as:

$$\hat{\mathbf{x}}_{\mathcal{S}} = \mathbf{C}\mathbf{x} = \mathbf{C}\,\mathbf{f}(\mathbf{L}, \tilde{\mathbf{x}}) \tag{8}$$

Where $\mathcal{S} \subseteq \mathcal{V}$ is the set of nodes that corresponds to the virtual sensors, $\mathbf{f}: \mathbb{R}^{N\times N} \times \mathbb{R}^N \to \mathbb{R}^N$ is the graph filter that takes as input the Laplacian matrix $\mathbf{L}$ and $\tilde{\mathbf{x}}$ which is a perturbed version of the input graph signal such that $\tilde{\mathbf{x}}_{\mathcal{S}}$=0. Finally, $\mathbf{C} \in \mathbb{B}^{|\mathcal{S}|\times N}$ is the sampling matrix, such that $\mathbf{C}_{\mathcal{S}}$=1. Then, the filter coefficients are learned by minimizing the residual sum of squares over the training samples. For simplicity, we use a third order Volterra-like graph filter (D=3) and filter depth K=2 to limit the error propagation.

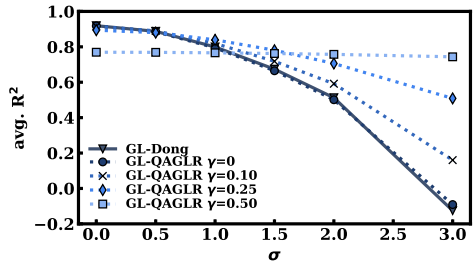


Figure 4. Average test R^2 for the virtual sensors for different levels of noise (σ) and ten repetitions.

Figure 4 shows the average R^2 obtained for the two virtual sensors (and ten repetitions) for different noise levels and different GL models. As it can be seen, the GL-Dong and the GL-QAGLR with γ=0 obtain the same performance since for γ=0 the QAGLR reduces to the GL-Dong method. Moreover,

it is seen how introducing little quality-aware regularization (γ=0.25) mitigates the error in the reconstruction by promoting connections of the virtual sensors with the high-precision nodes. For instance, for $\sigma^2=\{1.5, 2.0\}$ the GL-QAGLR with γ=0.25 is able to obtain an average test R^2 of 0.78 and 0.71 respectively, improving the performance of the GL-Dong by 0.11 and 0.20. In the extreme case, where the regularization is large (γ=0.50), the reconstruction performance is invariant regardless of the noise introduced since the resulting graph connects the virtual sensors only with high precision nodes, resulting in an average test R^2 of 0.76. Therefore, we can conclude that there exists a trade-off between the goodness-of-fit of the learned graph and the robustness it provides. However, we highlight that introducing a low level of regularization may result in a good graph and a graph that is more robust against possible LCS failures.

4 Conclusions

In this paper, we have proposed the quality-aware graph learning regularization (QAGLR) framework for GL models that allows the introduction of sensor quality-aware information from a heterogeneous sensor network into a GL model. This solution arises from the need to force connections between high-precision sensors and LCSs to perform signal reconstruction tasks more reliably since in the air quality monitoring paradigm sensors of different quality coexist in a network. The results have shown how in cases where LCSs can become noisy, forcing connections with reliable sensors can mitigate the error in the case the graph is fed to a graph signal reconstruction model. More precisely, we have shown the case of two virtual sensors implemented using a graph filter. The regularization framework has allowed mitigating the impact of the noise introduced in the LCSs given the robustness provided to the learned graph by encouraging connections of LCSs with more reliable sensors, therefore, taking into account the quality of the sensors. As a future work, sensor placement in heterogeneous networks is crucial, i.e., the study of high-precision nodes placement to provide the network with robustness in the long term.

Acknowledgements

Work supported by projects PID 2019-107910RB-I00, 2021 SGR-01059, and CDTI MIG-20221061.

5 References

[1] J. M. Barcelo-Ordinas, P. Ferrer-Cid, J. Garcia-Vidal, M. Viana, and A. Ripoll. H2020 project captor dataset: Raw data collected by low-cost mox ozone sensors in a real air pollution monitoring network. *Data in brief*, 36:107127, 2021.

[2] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1-4, 2004. Proceedings 17*, pages 624–638. Springer, 2004.

[3] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero. Learning sparse graphs under smoothness prior. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6508–6512. IEEE, 2017.

[4] E. Clark, S. L. Brunton, and J. N. Kutz. Multi-fidelity sensor selection: Greedy algorithms to place cheap and expensive sensors with cost constraints. *IEEE Sens. J.*, 21(1):600–611, 2020.

[5] F. Concas, J. Mineraud, E. Lagerspetz, S. Varjonen, X. Liu, K. Puolamäki, P. Nurmi, and S. Tarkoma. Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis. *ACM Transactions on Sensor Networks (TOSN)*, 17(2):1–44, 2021.

[6] T. H. Do, E. Tsiligianni, X. Qin, J. Hofman, V. P. La Manna, W. Philips, and N. Deligiannis. Graph-deep-learning-based inference of fine-grained air quality from mobile iot sensors. *IEEE Internet Things J.*, 7(9):8943–8955, 2020.

[7] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016.

[8] X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Process. Mag.*, 36(3):44–63, 2019.

[9] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Process. Mag.*, 37(6):117–127, 2020.

[10] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under laplacian and structural constraints. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):825–841, 2017.

[11] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Graph learning techniques using structured data for iot air pollution monitoring platforms. *IEEE Internet Things J.*, 8(17):13652–13663, 2021.

[12] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Data reconstruction applications for iot air pollution sensor networks using graph signal processing. *J. Netw. Comput. Appl.*, 205:103434, 2022.

[13] P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Volterra graph-based outlier detection for air pollution sensor networks. *IEEE Trans. Netw. Sci. Eng.*, 9(4):2759–2771, 2022.

[14] J. H. Giraldo, A. Mahmood, B. Garcia-Garcia, D. Thanou, and T. Bouwmans. Reconstruction of time-varying graph signals via sobolev smoothness. *IEEE Transactions on Signal and Information Processing over Networks*, 8:201–214, 2022.

[15] I. Jabłoński. Graph signal processing in applications to sensor networks, smart grids, and smart cities. *IEEE Sens. J.*, 17(23), 2017.

[16] X. Jiang, Z. Tian, and K. Li. A graph-based approach for missing sensor data imputation. *IEEE Sens. J.*, 21(20):23133–23144, 2021.

[17] V. Kalofolias. How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pages 920–929. PMLR, 2016.

[18] Y. Li, S. Xie, Z. Wan, H. Lv, H. Song, and Z. Lv. Graph-powered learning methods in the internet of things: A survey. *Machine Learning with Applications*, 11:100441, 2023.

[19] B. Maag, Z. Zhou, and L. Thiele. A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet Things J.*, 5(6):4857–4870, 2018.

[20] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Process. Mag.*, 36(3):16–43, 2019.

[21] L. Morawska, P. K. Thai, X. Liu, A. Asumadu-Sakyi, G. Ayoko, A. Bartonova, A. Bedini, F. Chai, B. Christensen, M. Dunbabin, et al. Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone? *Environment international*, 116:286–299, 2018.

[22] N. H. Motlagh, E. Lagerspetz, P. Nurmi, X. Li, S. Varjonen, J. Mineraud, M. Siekkinen, A. Rebeiro-Hargrave, T. Hussein, T. Petaja, et al. Toward massive scale air quality monitoring. *IEEE Communications Magazine*, 58(2):54–59, 2020.

[23] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[24] A. Sandryhaila and J. M. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.*, 31(5), 2014.

[25] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013.

[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[27] Z. Xiao, H. Fang, and X. Wang. Distributed nonlinear polynomial graph filter and its output graph spectrum: Filter analysis and design. *IEEE Transactions on Signal Processing*, 69:1725–1739, 2021.