
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Miettinen, Jari; Vorobyov, Sergiy; Ollila, Esa

Robust Least Mean Squares Estimation of Graph Signals

Published in:

44th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2019; Brighton; United Kingdom; 12-17 May 2019 : Proceedings

DOI:

[10.1109/ICASSP.2019.8683193](https://doi.org/10.1109/ICASSP.2019.8683193)

Published: 01/05/2019

Document Version

Peer reviewed version

Please cite the original version:

Miettinen, J., Vorobyov, S., & Ollila, E. (2019). Robust Least Mean Squares Estimation of Graph Signals. In *44th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2019; Brighton; United Kingdom; 12-17 May 2019 : Proceedings* (pp. 5416-5420). [8683193] (Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing). IEEE.
<https://doi.org/10.1109/ICASSP.2019.8683193>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

ROBUST LEAST MEAN SQUARES ESTIMATION OF GRAPH SIGNALS

Jari Miettinen, *Member, IEEE*, Sergiy A. Vorobyov, *Fellow, IEEE* and Esa Ollila, *Member, IEEE*

Department of Signal Processing and Acoustics, Aalto University, P.O. Box 15400, FIN-00076 Aalto

ABSTRACT

Recovering a graph signal from samples is a central problem in graph signal processing. Least mean squares (LMS) method for graph signal estimation is computationally efficient adaptive method. In this paper, we introduce a technique to robustify LMS with respect to mismatches in the presumed graph topology. It builds on the fact that graph LMS converges faster when the graph topology is specified correctly. We consider two measures of convergence speed, based on which we develop randomized greedy algorithms for robust interpolation of graph signals. In simulation studies, we show that the randomized greedy robust least mean squares (RGRLMS) outperforms the regular LMS and has even more potential given a robust sampling design.

Index Terms— Graph signal processing, Laplacian matrix, least mean squares

1. INTRODUCTION

Signal processing on graphs has been a rapidly developing field of research during the past few years [1, 2, 3]. It deals with signals lying on irregular domains, which are represented by graphs. Each unit of a graph signal is associated to a node in the underlying graph, and the edges of the graph indicate pairwise connections between the units. The choice of the graph topology requires some prior information, which may be based on physical structure of a network, training data, expert perception, or some other source of data on the proximity between the units. One of the most important problems in graph signal processing is interpolation from samples. If a signal is bandlimited with respect to a graph, and this graph is known, the signal can be recovered accurately by sampling only small part of the signal.

In this paper, we consider recovering a graph signal, when the presumed graph differs from the underlying graph. The effect of such graph error has been of increasing interest recently (see [4, 5, 6]), and as far as we know, the first robust methods with respect to errors in graph topology was proposed in [7], where graph signal spectral analysis and clustering were considered. However, the authors of [7] assume knowledge of edgewise probabilities of perturbation, which

differs from our approach as we only rely on the estimate of the graph. Our method uses least mean squares estimation (LMS) of graph signals [8, 9] and is based on connection between the speed of convergence and the quality of the estimate. It appears to be less important to improve the sampling strategy part of the solution, since it tolerates more errors in the presumed graph. However, it is still of interest for a future work.

Notation: We use boldface capital letters for matrices, boldface lowercase letters for vectors, and capital calligraphic letters for sets. The exceptions are $\mathbf{1}_N$ which is the N -dimensional vector full of ones, the $M \times N$ matrix full of ones $\mathbf{1}_{M \times N} = \mathbf{1}_M \mathbf{1}_N^\top$, and $\mathbf{1}_A$ is a matrix of the same size as \mathbf{A} , such that $[\mathbf{1}_A]_{ij} = 1$, if $a_{ij} \neq 0$ and $[\mathbf{1}_A]_{ij} = 0$, if $a_{ij} = 0$. The matrix $\mathbf{I}_{N \times N}$ is the $N \times N$ identity matrix. The notations $(\cdot)^\top$, \odot , $\|\cdot\|$, $\text{tr}\{\cdot\}$, $\mathbb{P}(\cdot)$, $\mathbb{E}\{\cdot\}$, and $\text{var}(\cdot)$ stand for the transpose, Hadamard product, Euclidian norm of a vector, trace of a matrix, probability, mathematical expectation, and variance, respectively.

2. LEAST MEAN SQUARES ESTIMATION OF GRAPH SIGNALS

Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ consists of a set of N nodes \mathcal{N} and the set of edges \mathcal{E} . The adjacency matrix of the graph \mathcal{G} , denoted as \mathbf{A} , is an $N \times N$ matrix such that $a_{ii} = 0$ for $i = 1, \dots, N$ and $a_{ij} \neq 0$ if and only if $(i, j) \in \mathcal{E}$, i.e., there is a link from j th node to i th node. In this paper, we consider unweighted graphs and $a_{ij} \in \{0, 1\}$. If the graph is undirected, the adjacency matrix is symmetric and then graph Laplacian matrix is defined as $\mathbf{L} = \text{diag}(\mathbf{1}^\top \mathbf{A}) - \mathbf{A}$, where $\text{diag}(\mathbf{1}^\top \mathbf{A})$ is a diagonal matrix of node degrees of the graph. The Laplacian matrix is positive semi-definite, and it has the eigendecomposition $\mathbf{L} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^H$, where the columns of \mathbf{V} are the eigenvectors of \mathbf{L} and $\mathbf{\Lambda}$ is a diagonal matrix consisting of the eigenvalues of \mathbf{L} .

A graph signal \mathbf{x} on the graph \mathcal{G} is an N -dimensional vector of complex numbers, whose i th element is attributed to the i th node of the graph. The Graph Fourier Transform (GFT) of the signal \mathbf{x} is defined as $\mathbf{s} = \mathbf{V}^H \mathbf{x}$. The graph signal \mathbf{x} is bandlimited if the cardinality of the support of \mathbf{s} , denoted as $\mathcal{F} = \{i \in \{1, \dots, N\} \mid s_i \neq 0\}$, is smaller than N , and the bandwidth is $|\mathcal{F}|$. Finally, let us define $N \times |\mathcal{F}|$ matrix $\mathbf{V}_{\mathcal{F}}$ which contains the columns of \mathbf{V} given by \mathcal{F} .

This research was supported in parts by the Academy of Finland grants No. 299243 and No. 298118

Let us assume that \mathbf{x} is a bandlimited signal on the graph \mathcal{G} so that its GFT \mathbf{s} is nonzero only on the indices of \mathcal{F} . Then we can represent \mathbf{x} as $\mathbf{x} = \mathbf{V}_{\mathcal{F}}\mathbf{s}$. The bandwidth $|\mathcal{F}|$ is often, including the examples of this paper, assumed to be known, but for example in [8], estimation of the bandwidth is considered. The graph signal is sampled at M time points and the sampling strategy will be probabilistic following [9]. $\mathcal{S}[n]$ denotes the set of sampled nodes at time point n and $\mathbf{D}_{\mathcal{S}[n]} = \text{diag}(d_1[n], \dots, d_N[n])$, where $d_i[n] = 1$ if $i \in \mathcal{S}[n]$, and 0 otherwise. The model for observations is given by

$$\mathbf{y}[n] = \mathbf{D}_{\mathcal{S}[n]}(\mathbf{x} + \boldsymbol{\nu}[n]), \quad n = 1, \dots, M,$$

where $\boldsymbol{\nu}[n]$ is the measurement noise with zero mean and covariance matrix \mathbf{C}_{ν} , and it is assumed to be temporally independent. The LMS method for graph signal interpolation, as suggested in [8, 9], finds the estimate for \mathbf{s} by solving the following optimization problem

$$\min_{\mathbf{s}} \mathbb{E} \|\mathbf{D}_{\mathcal{S}[n]}(\mathbf{y}[n] - \mathbf{V}_{\mathcal{F}}\mathbf{s})\|^2. \quad (1)$$

Using stochastic gradient descent approach to estimate \mathbf{s} , and after transforming back to signal space, an update step for estimating \mathbf{x} is given by

$$\hat{\mathbf{x}}[n+1] = \hat{\mathbf{x}}[n] + \mu \mathbf{B}_{\mathcal{F}} \mathbf{D}_{\mathcal{S}[n]}(\mathbf{y}[n] - \hat{\mathbf{x}}[n]) \quad (2)$$

where μ is step-size and $\mathbf{B}_{\mathcal{F}} = \mathbf{V}_{\mathcal{F}} \mathbf{V}_{\mathcal{F}}^H$.

Having a good sampling strategy is an important factor in the success of LMS. Several designs for probabilistic sampling were proposed in [9]. The strategy that we will use aims to find optimal vector \mathbf{p} which assigns probabilities for the nodes to be sampled, by minimizing the mean square deviation of the LMS estimate subject to constraints on the sum of probabilities and convergence rate. The corresponding optimization problem can be written as

$$\min_{\mathbf{p}} \frac{\text{tr} \{ \mathbf{V}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_{\nu} \mathbf{V}_{\mathcal{F}} \}}{\lambda_{\min}(\mathbf{V}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{V}_{\mathcal{F}})} \quad (3)$$

$$\text{s.t. } \mathbf{1}^{\top} \mathbf{p} \leq P, \quad 0 \leq p_i \leq p_i^{\max}, \quad i = 1, \dots, N$$

$$\lambda_{\min}(\mathbf{V}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{V}_{\mathcal{F}}) \geq \frac{1 - \alpha}{2\mu}$$

where $\lambda_{\min}(\cdot)$ yields the smallest eigenvalue of the matrix, P can be seen as a budget constraint which limits the sampling probabilities, α is a prescribed target value for convergence rate and $p_1^{\max}, \dots, p_N^{\max}$ give nodewise bounds for the probabilities.

The objective function is based on approximation of the mean square deviation derived in [9]:

$$\text{tr} \left\{ (\mathbf{V}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{V}_{\mathcal{F}})^{-1} \mathbf{V}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_{\nu} \mathbf{V}_{\mathcal{F}} \right\}. \quad (4)$$

The objective function in (3), which is of the form convex over concave, is an upper bound of the non-convex (4). Thus,

the problem (3) leads to a pseudoconvex optimization problem, which can be solved with Dinkelbach algorithm using for example convex program solver CVX [10, 11]. Finding the global optimum with the Dinkelbach method is guaranteed [12]. For more information on this and other approaches to deal with the non-convexity of (4), see [9].

3. ROBUST LEAST MEAN SQUARES ESTIMATION OF GRAPH SIGNALS

Success of recovering a graph signal from samples using LMS estimation depends on two key factors: the sampling probability \mathbf{p} and the matrix $\mathbf{B}_{\mathcal{F}}$. Both of them depend on the presumed underlying graph, but the latter appears to be more sensitive to errors in the adjacency matrix. While there is a significant difference in performance between \mathbf{p} derived from (3) using the correct graph and random \mathbf{p} , even quite rough approximation of the graph adjacency matrix for (3) would still yield satisfactory outcome if $\mathbf{B}_{\mathcal{F}}$ is accurate. The opposite case where \mathbf{p} is optimal and $\mathbf{B}_{\mathcal{F}}$ is erroneous, leads to significantly worse outcome. Therefore, we first focus on coping with the issue of having erroneous estimate of $\mathbf{B}_{\mathcal{F}}$.

Our method to estimate \mathbf{x} robustly with respect to the choice of the adjacency matrix, is based on the observation that the LMS algorithm converges the faster, the better the matrix $\mathbf{B}_{\mathcal{F}}$ corresponds to \mathbf{x} . This connection is utilized so that the speed of convergence is maximized with respect to the adjacency matrix behind $\mathbf{B}_{\mathcal{F}}$, and the maximizing $\mathbf{B}_{\mathcal{F}}$ is used for graph LMS. We consider two criteria for the speed of convergence. The first one is to use initial value $\hat{\mathbf{x}}_0[0] = \mathbf{0}$ and maximize the energy

$$\|\hat{\mathbf{x}}_0[M_0]\| \quad (5)$$

of the estimate after $M_0 \leq M$ steps of algorithm (2). The second one is to generate two random initial values $\hat{\mathbf{x}}_1[0]$ and $\hat{\mathbf{x}}_2[0]$ and maximize the correlation

$$\text{cor}(\hat{\mathbf{x}}_1[M_0], \hat{\mathbf{x}}_2[M_0]). \quad (6)$$

Both of the corresponding optimization problems are non-convex. Thus, we propose a greedy algorithm which involves sampling candidate adjacency matrices around the prevailing adjacency matrix estimate, and proceeding to the matrix which gives the maximal value of (5) or (6). This is repeated until the adjacency matrix estimate does not change. The sampling is done using one of the graph error models presented in [4]. The model is written as

$$\mathbf{A}_{\epsilon_1, \epsilon_2} = \mathbf{A} - \Delta_{\epsilon_1} \odot \mathbf{A} + \Delta_{\epsilon_2} \odot (\mathbf{1}_{N \times N} - \mathbf{I}_{N \times N} - \mathbf{A}), \quad (7)$$

where Δ_{ϵ} denotes a symmetric random matrix whose elements $(\Delta)_{ij} = (\Delta)_{ji} = 1$ with probability ϵ and $(\Delta)_{ij} = (\Delta)_{ji} = 0$ with probability $1 - \epsilon$, i.e., realization of Δ_{ϵ} is adjacency matrix of the Erdős-Rényi graph [13]. The parameters ϵ_1 and ϵ_2 define the probabilities for deleting and adding

edges, respectively. The random sampling approach that we are using is supposedly not optimal, and possibly the sampling can be directed somehow in order to reach faster convergence. Specifically, if prior knowledge on the edge probabilities is available as assumed in [7], it can be incorporated via changing the sampling procedure.

We call the proposed algorithm *randomized greedy robust least mean squares* (RGRMLS) estimator, and summarize it in Algorithm 1. The outcome of the algorithm is stochastic due to the random generation of the candidate matrices. Therefore, averaging over several runs of the algorithm is an option to enhance the accuracy. In Section 4, we compare the two approaches, (a) and (b) (see Algorithm 1), in means of simulations, and consider a hybrid of them.

There are two parameters which need to be specified for the algorithm. The first one is $\tilde{\epsilon}_1$ and it can be seen as a step size, since it defines how close to $\hat{\mathbf{A}}_0$ the candidate matrices are picked. In step 3 of the proposed algorithm, five values of $\tilde{\epsilon}_2$ are determined so that the number of edges may increase or decrease. In model (7), the expected number of edges does not change when $\epsilon_2 = (\epsilon_1 \|\mathbf{A}\|_0) / (N(N-1) - \|\mathbf{A}\|_0)$. Parameter K states how many times one value of $\tilde{\epsilon}_2$ is used, the total number of candidate matrices then being $5K$. Too small value of K leads to getting stuck in a suboptimal solution whereas excessively large value of K increases the computational cost without benefit.

Algorithm 1 Randomized greedy robust LMS algorithms

Input: $\mathbf{y}[1], \dots, \mathbf{y}[M], \mathbf{D}_{S[1]}, \dots, \mathbf{D}_{S[M]}$, adjacency matrix estimate $\hat{\mathbf{A}}$, step size $\tilde{\epsilon}_1, K$

Output: $\hat{\mathbf{x}}$

Initialisation :

(a) $\hat{\mathbf{x}}_0[0] = \mathbf{0}$

(b) random initial values $\hat{\mathbf{x}}_1[0]$ and $\hat{\mathbf{x}}_2[0]$

- 1: **while** $\hat{\mathbf{A}} \neq \hat{\mathbf{A}}_0$ **do**
 - 2: $\hat{\mathbf{A}}_0 \leftarrow \hat{\mathbf{A}}$
 - 3: Set $\tilde{\epsilon}_2^i = \frac{(i+1)\tilde{\epsilon}_1 w}{4(1-w)}$ for $i = 1, \dots, 5$, where $w = \frac{\|\hat{\mathbf{A}}\|_0}{N(N-1)}$
 - 4: Draw a set of adjacency matrices $\hat{\mathbf{A}}_1, \dots, \hat{\mathbf{A}}_{5K}$ from the graph error model (7) around $\hat{\mathbf{A}}_0$ using $\tilde{\epsilon}_1$ and each of $\tilde{\epsilon}_2^i$ K times.
 - 5: (a) Run (2) for $\mathbf{B}_{\mathcal{F}}^0, \mathbf{B}_{\mathcal{F}}^1, \dots, \mathbf{B}_{\mathcal{F}}^{5K}$ with initial value $\hat{\mathbf{x}}_0[0] = \mathbf{0}$.
(b) Run (2) for $\mathbf{B}_{\mathcal{F}}^0, \mathbf{B}_{\mathcal{F}}^1, \dots, \mathbf{B}_{\mathcal{F}}^{5K}$ with initial values $\hat{\mathbf{x}}_1[0]$ and $\hat{\mathbf{x}}_2[0]$.
 - 6: Find the adjacency matrix $\hat{\mathbf{A}}_{min}$ which yields largest value of
(a) $\|\hat{\mathbf{x}}_0[M]\|$
(b) $\text{cor}(\hat{\mathbf{x}}_1[M], \hat{\mathbf{x}}_2[M])$
 - 7: $\hat{\mathbf{A}} \leftarrow \hat{\mathbf{A}}_{min}$
 - 8: **end while**
 - 9: **return** $\hat{\mathbf{x}}[M]$ given by LMS with $\hat{\mathbf{A}}$
-

4. SIMULATION STUDY

In this section, we conduct a simulation study where we both illustrate the performance of the robust estimator and also give some insights into why it works. The dimensions and parameters of the data are as follows: $N = 100, M = 50, |\mathcal{F}| = 5$, the noise covariance matrix \mathbf{C}_v is diagonal and the elements drawn uniformly on $[0.1, 0.2]$, $\mathbf{x} = \mathbf{V}_{\mathcal{F}} \mathbf{1}_{|\mathcal{F}|}$ is given by the $|\mathcal{F}|$ lowest frequency components of \mathbf{A} and it is standardized so that $\mathbf{1}^\top \mathbf{x} = 0$ and $\mathbf{x}^\top \mathbf{x} = N - 1$. The number of edges in \mathbf{A} is 0.1 times the number of pairs of nodes. The average sampling probability, i.e., $\mathbf{1}^\top \mathbf{p}$ is set to 0.125. Most of the probabilities are zeros. Depending on the specific \mathbf{A} and \mathbf{C}_v , the number of nonzero probabilities varies between 19 and 25 most of the time. In Algorithm 1, we use values $\tilde{\epsilon}_1 = 0.01$ and $K = 4$, i.e., the number of candidate matrices is 20. We have noticed these values to be reasonably good in different setups, but we have not aimed to optimize them for this specific simulation setup.

Fig. 1 shows results for different levels of adjacency matrix mismatch in three setups. On the left-hand side, we have the realistic setting where \mathbf{p} is obtained using $\hat{\mathbf{A}}$ given by the graph error model (7) with $\epsilon_1 = 0, 0.1, 0.2$ and $\epsilon_2 = 0, 0.025, 0.05$. On the right-hand side, \mathbf{p} is derived from \mathbf{A} , but otherwise the same matrices $\hat{\mathbf{A}}$ are used as on the left. However, since $\hat{\mathbf{A}} = \mathbf{A}$ when $\epsilon_1 = \epsilon_2 = 0$, we use the position on top right to display the results in the case that \mathbf{p} is replaced by $\tilde{\mathbf{p}} = \mathbf{1}_{\mathbf{p}>0}$, and $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.025$. Then the sampling is deterministic. The boxplots are made of correlations between \mathbf{x} and $\hat{\mathbf{x}}$ in 400 repetitions with differing matrices \mathbf{A} . The following six estimates are included:

- 'LMS' The LMS estimate given by $\hat{\mathbf{A}}$. Benchmark for what is achieved when the adjacency matrix is mislearned.
- 'ab($\hat{\mathbf{A}}$)' The average of one run of (a) and one run of (b).
- '2a($\hat{\mathbf{A}}$)' The average of two runs of (a).
- '2b($\hat{\mathbf{A}}$)' The average of two runs of (b).
- '2ab($\hat{\mathbf{A}}$)' The average of two runs of (a) and two runs of (b).
- 'A' The oracle LMS estimate given by \mathbf{A} .

Table 1 shows the averages of the mean squared errors (MSE's).

When the graph topology is known exactly, the LMS estimate is naturally very good, and the robust methods lose some efficiency, (b) more than (a) and the combination is in between. However, when there are errors in the presumed graph, the robust methods become favourable. The difference is much larger when \mathbf{p} is based on \mathbf{A} , which implies the demand of robust sampling procedure as well. The top right plot indicates that RGRMLS needs varying samples, because the results with $\tilde{\mathbf{p}}$ are worse even though the number of sampled nodes is higher at each time point.

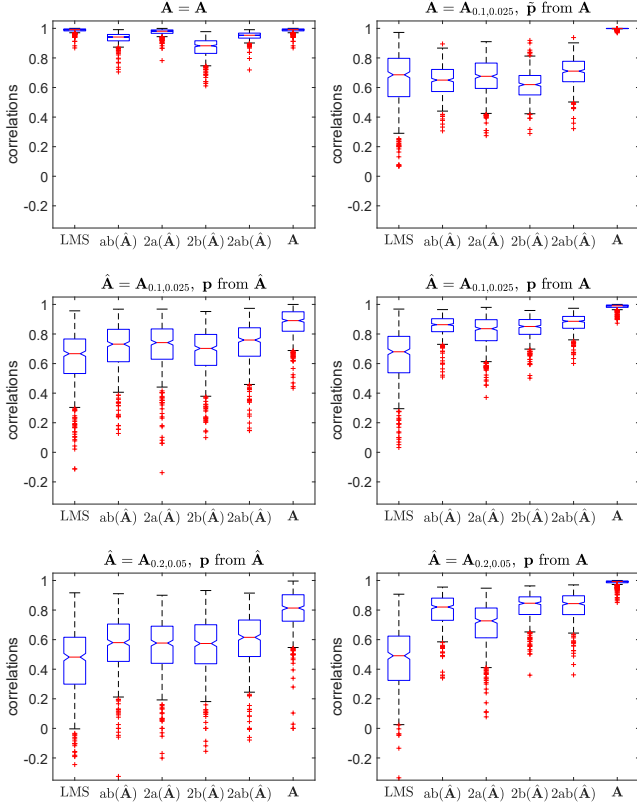


Fig. 1. Boxplots of correlations between \mathbf{x} and $\hat{\mathbf{x}}$ for different estimates. On the left-hand side, the sampling probabilities are optimized using the erroneous adjacency matrix $\hat{\mathbf{A}}$, whereas \mathbf{A} is used on the right-hand side. On the top right, $\tilde{\mathbf{p}}$ is used instead of \mathbf{p} . In a boxplot, the red line marks median value, the boundaries of the box are 25th and 75th percentiles, and the maximum whisker length is defined so that the range within whiskers would cover approximately 99.3% of the points if they were normally distributed with those 25th and 75th percentiles. The points outside the whiskers are marked by red plus signs.

We also study how the measures of convergence speed in Algorithm 1 and the estimation accuracy are related at different numbers of time points M . For this, 20 adjacency matrices are generated and for each of them the LMS sequence $\hat{\mathbf{x}}[1], \dots, \hat{\mathbf{x}}[100]$ is computed with respect to both \mathbf{p} and $\tilde{\mathbf{p}}$. Fig. 2 shows the averages of correlations between $\text{cor}(\mathbf{x}, \hat{\mathbf{x}}[M])$ and (a) $\|\hat{\mathbf{x}}_0[M]\|$ and (b) $\text{cor}(\hat{\mathbf{x}}_1[M], \hat{\mathbf{x}}_2[M])$ for $M = 1, \dots, 100$. On the left, the sampling probabilities are optimized with respect to $\hat{\mathbf{A}}$, whereas on the right, \mathbf{A} is used. Congruently with the results in Fig. 1, we notice from Fig. 2 that the correlations are higher when the sampling probabilities correspond to the underlying adjacency matrix \mathbf{A} . For small M , the correlations are higher with $\tilde{\mathbf{p}}$ than with \mathbf{p} , probably due to the fact that $\mathbf{1}_N^\top \tilde{\mathbf{p}} > \mathbf{1}_N^\top \mathbf{p}$. However, in

Table 1. Average MSE's for the estimators in the same simulation setup as in Fig. 1. The setups are numbered as follows: 1. $\hat{\mathbf{A}} = \mathbf{A}$, 2. $\hat{\mathbf{A}} = \mathbf{A}_{0.1,0.025}$, $\tilde{\mathbf{p}}$ from $\hat{\mathbf{A}}$, 3. $\hat{\mathbf{A}} = \mathbf{A}_{0.2,0.05}$, $\tilde{\mathbf{p}}$ from $\hat{\mathbf{A}}$, 4. $\hat{\mathbf{A}} = \mathbf{A}_{0.1,0.025}$, $\tilde{\mathbf{p}}$ from \mathbf{A} , 5. $\hat{\mathbf{A}} = \mathbf{A}_{0.1,0.025}$, \mathbf{p} from \mathbf{A} , 6. $\hat{\mathbf{A}} = \mathbf{A}_{0.2,0.05}$, \mathbf{p} from \mathbf{A} .

	LMS	ab($\hat{\mathbf{A}}$)	2a($\hat{\mathbf{A}}$)	2b($\hat{\mathbf{A}}$)	2ab($\hat{\mathbf{A}}$)	\mathbf{A}
1.	0.28	0.36	0.30	0.44	0.35	0.28
2.	0.70	0.63	0.62	0.65	0.61	0.44
3.	0.87	0.78	0.78	0.78	0.75	0.53
4.	0.86	1.02	0.96	1.14	0.86	0.03
5.	0.69	0.48	0.51	0.47	0.47	0.28
6.	0.84	0.55	0.64	0.50	0.54	0.28

the case of $\tilde{\mathbf{p}}$, i.e., when the same set of nodes is sampled at each time point, the correlation fades as M increases. In the case of probabilistic sampling, the correlation increases with M before leveling off around $M = 40$.

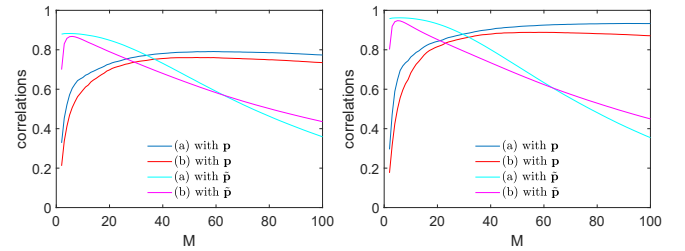


Fig. 2. Correlations between $\text{cor}(\mathbf{x}, \hat{\mathbf{x}})$ and the values given in step 5 of Algorithm 1 for $M = 1, \dots, 100$. On the left-hand side \mathbf{p} and $\tilde{\mathbf{p}}$ are derived from $\hat{\mathbf{A}}$ which is a realization of (7) with $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.025$, and on the right-hand side \mathbf{p} and $\tilde{\mathbf{p}}$ are derived from the underlying adjacency matrix \mathbf{A} .

5. CONCLUSION

For graph signal recovery from samples, we have proposed randomized greedy robust LMS method, which aims to be robust with respect to mismatch between the presumed graph topology and that of the underlying graph signal. In graph LMS, the mismatch affects the choices of the sampling and interpolation strategies. The proposed RGRLMS algorithm only modifies the interpolation part and it is based on the correlation between how fast LMS converges and how good the LMS estimate is for different graph topology assumptions. The correlation is stronger when the sampling probabilities are optimal, which implies that finding a robust sampling design is of interest as well, but it is left as a topic for future research. Also, it is worth studying whether the randomized greedy approach could be replaced by a more efficient way to handle the introduced optimization problem.

6. REFERENCES

- [1] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [2] 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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [3] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura and P. Vandergheynst, "Graph signal processing," *arXiv preprint arXiv:1712.00468*, 2017.
- [4] J. Miettinen, S.A. Vorobyov and E. Ollila, "Graph error effect in graph signal processing," in *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP Calgary, Canada*, 2018, pp. 4164–4168.
- [5] E. Ceci and S. Barbarossa. "Small perturbation analysis of network topologies," in *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP Calgary, Canada*, 2018, pp. 4194–4198.
- [6] E. Isufi, A. Loukas, A. Simonetto and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4406–4421, 2017.
- [7] E. Ceci and S. Barbarossa. "Robust graph signal processing in the presence of uncertainties on graph topology," *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications, SPAWC Kalamata, Greece*, 2018, pp. 1–5.
- [8] P. Di Lorenzo, S. Barbarossa, P. Banelli and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 555–568, 2016.
- [9] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Transactions on Signal Processing*, vol. 66, no. 13, pp. 3584–3598, 2018.
- [10] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," *Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura*, es, pp. 95–110, Lecture Notes in Control and Information Sciences, Springer, 2008 http://stanford.edu/~boyd/graph_dcp.html
- [11] M. Grant and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming, version 2.1, <http://cvxr.com/cvx>, March 2014.
- [12] W. Dinkelbach, "On nonlinear fractional programming," *Management science*, vol. 13, no. 7, pp. 492–498, 1967.
- [13] P. Erdős and A. Rényi, "On random graphs," *Publ. Math Debrecen*, vol. 6, pp. 290–297, 1959.