Adaptive Least Mean Squares Estimation of Graph Signals

ASP Final Report (Anoushka Vyas - 20171057)

***Abstract*— The aim of this project is to propose a least mean squares (LMS) strategy for adaptive estimation of signals defined over graphs. Assuming the graph signal to be band-limited, over a known bandwidth, the method enables reconstruction, with guaranteed performance in terms of mean-square error, and tracking from a limited number of observations over a subset of vertices.**
Furthermore, to cope with the case where the bandwidth is not known beforehand, a method is proposed that performs a sparse online estimation of the signal support in the (graph) frequency domain, which enables online adaptation of the graph sampling strategy.

I. LMS ESTIMATION OF GRAPH SIGNALS

A. Assumptions

The signal is initially assumed to be perfectly band-limited, i.e. its spectral content is different from zero only on a limited set of frequencies F and the non-zero frequency nodes are known.

B. Localization operators

Given a subset of vertices $S \subseteq V$, we define a vertex-limiting operator as the diagonal matrix:

$$D_S = diag(1_S) \tag{1}$$

where 1_S is the set indicator vector, whose $i-th$ entry is equal to one, if $x_0 \in S$, or zero otherwise. Similarly, given a subset of frequency indices $F \subseteq V$, we introduce the filtering operator:

$$B_F = U\Sigma_F U^{\mathrm{H}} \tag{2}$$

where Σ_F is a diagonal matrix defined as:

$$\Sigma_F = diag(1_F) \tag{3}$$

C. Model

Let us consider a signal $x_0 \in C^N$ defined over the graph $G = (V, E)$. The signal is initially assumed to be perfectly band-limited, i.e. its spectral content is different from zero only on a limited set of frequencies F. Let us consider partial observations of signal x_0 , i.e. observations over only a subset of nodes. Denoting with S the sampling set (observation subset), the observed signal at time n can be expressed as:

$$y[n] = D(x_0 + v[n]) = DBx_0 + Dv[n] \tag{4}$$

where D is the vertex-limiting operator defined in (1) , which takes nonzero values only in the set S, and $v[n]$ is a zero-mean, additive noise with covariance matrix C_v . The second equality in (4) comes from the bandlimited assumption, i.e. $Bx_0 = x_0$, with B denoting the operator in (2) that projects onto the (known) frequency set F .

D. Theorems for reconstruction

$Theorem1$: [1] There is a vector x, perfectly localized over both vertex set S and frequency set F (i.e. $x \in B_F \cap D_S$) if and only if the operator BDB (or DBD) has an eigenvalue equal to one; in such a case, x is an eigenvector of BDB associated to the unitary eigenvalue.

$Theorem2$: [1] Any band-limited signal x_0 can be reconstructed from its samples taken in the set S, if and only if $\|\bar{D}B\| < 1$, i.e. if the matrix $B\bar{D}B$ does not have any eigenvector that is perfectly localized on $\bar{S}$ and bandlimited on F . Here $\bar{D}$ = $I - D$.

E. Optimization

Following an LMS approach , the optimal estimate for x_0 can be found as the vector that solves the following optimization problem:

$$\min_{\boldsymbol{x}} \ \mathbf{E}\,\|\boldsymbol{y}[n] - \mathbf{DB}\boldsymbol{x}\|^2$$
$$\text{s.t.} \quad \mathbf{B}\boldsymbol{x} = \boldsymbol{x},$$

For a stationary $y[n]$ the optimal solution is given by $\hat{x}$ which satisfies:

$$BDB\hat{x} = BDE(y[n]) \tag{5}$$

Here the value $\hat{x}$ minimizes the optimization problem and is bandlimited i.e. it satisfies $B\hat{x} = \hat{x}$.
The issue here is that to calculate the expectation in (5) we need the entire information about $y[n]$ which may not be available so we use the steepest-descent method.

F. Sampling strategies

The properties of the LMS algorithm strongly depend on the choice of the sampling set S, i.e. on the vertex limiting operator D. The sampling strategy must be carefully designed in order to:

- enable reconstruction of the signal;
- guarantee stability of the algorithm; and
- impose a desired mean-square error at convergence.

Both the number of samples and their location is fundamental for the performance of the algorithm. Three algorithms were used to find the samples to be taken for reconstruction algorithm:

- Minimum Mean square deviation
- Maximization of determinant
- Maximization of the minimum eigenvalue

1) Minimum Mean square deviation

The method iteratively selects the samples from the graph that lead to the largest reduction in terms of steady state MSD. The steady state MSD is given as:

$$\mathrm{MSD} = \lim_{n\to\infty} \mathbf{E}\|\widetilde{x}[n]\|^2 = \lim_{n\to\infty} \mathbf{E}\|\widehat{s}[n]\|^2$$
$$= \mu^2 \mathrm{vec}(\mathbf{G})^T (\mathbf{I}-\mathbf{Q})^{-1} \mathrm{vec}(\mathbf{I}).$$

$$\mathbf{G} = \mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{C}_v \mathbf{D} \mathbf{U}_{\mathcal{F}}$$
$$\mathbf{Q} = (\mathbf{I} - \mu\, \mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}}) \otimes (\mathbf{I} - \mu\, \mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}}).$$

where $\widetilde{s[n]}$ is the GFT of $x[n]-x_0$ and U_F the matrix having as columns the eigenvectors of the Laplacian matrix associated to the frequency indices F.

Sampling strategy 1: Minimization of MSD

Input Data : M, the number of samples.
Output Data : $\mathcal{S}$, the sampling set.
Function : initialize $\mathcal{S} \equiv \emptyset$
while $|\mathcal{S}| < M$
$s = \arg\min_j \ \mathrm{vec}(\mathbf{G}(\mathbf{D}_{\mathcal{S}\cup\{j\}}))^T (\mathbf{I} - \mathbf{Q}(\mathbf{D}_{\mathcal{S}\cup\{j\}}))^{\dagger} \mathrm{vec}(\mathbf{I})$;
$\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$;
end

2) Maximization of determinant

The rationale underlying this strategy is to design a well suited basis for the graph signal that is to be estimated. This criterion coincides with the maximization of the the pseudo-determinant of the matrix $U_F{}^{\mathrm{H}} D U_F$ i.e. the product of all non zeros eigenvalues. The proof is as follows:

$$\mathrm{MSD} = \mu^2 \mathrm{vec}(\mathbf{G})^T (\mathbf{I}-\mathbf{Q})^{-1} \mathrm{vec}(\mathbf{I})$$
$$= \mu^2 \mathrm{vec}(\mathbf{G})^T \mathbf{V} (\mathbf{I}-\mathbf{\Lambda})^{-1} \mathbf{V}^H \mathrm{vec}(\mathbf{I})$$
$$= \mu^2 \sum_{i=1}^{|\mathcal{F}|^2} \frac{p_i \cdot q_i}{1-\lambda_i(\mathbf{Q})}$$

$$\boldsymbol{p} = \{p_i\} = \mathbf{V}^H \mathrm{vec}(\mathbf{G}),\ \boldsymbol{q} = \{q_i\} = \mathbf{V}^H \mathrm{vec}(\mathbf{I}).$$

$$\lambda_i(\mathbf{Q}) = \left(1 - \mu \lambda_k(\mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}})\right)\left(1 - \mu \lambda_l(\mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}})\right)$$

Sampling strategy 2: Maximization of $\left|\mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}}\right|_+$

Input Data : M, the number of samples.
Output Data : $\mathcal{S}$, the sampling set.
Function : initialize $\mathcal{S} \equiv \emptyset$
while $|\mathcal{S}| < M$
$s = \arg\max_j \ \left|\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}\cup\{j\}} \mathbf{U}_{\mathcal{F}}\right|_+$;
$\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$;
end

For minimum MSD we want these eigenvalues to be as far from 1 as possible. Thus, the eigenvalues of matrix $U_F{}^{\mathrm{H}} D U_F$ has to be as far away from zero as possible. Thus, it equivalent to maximizing the determinant.

3) Maximization of the minimum eigenvalue

Sampling strategy 3: Maximization of $\lambda_{\min}^{+}\left(\mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{U}_{\mathcal{F}}\right)$

Input Data : M, the number of samples.
Output Data : $\mathcal{S}$, the sampling set.
Function : initialize $\mathcal{S} \equiv \emptyset$
while $|\mathcal{S}| < M$
$s = \arg\max_j \ \lambda_{\min}^{+}\left(\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}\cup\{j\}} \mathbf{U}_{\mathcal{F}}\right)$;
$\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$;
end

G. LMS Estimation Algorithm

In the LMS algorithm for graph signals two things are important, D is an idempotent operator and $Bx[n] = x[n]$ (i.e., $x[n]$ is band-limited) for all n. The algorithm is given as:

Algorithm 1: LMS algorithm for graph signals

Start with $\boldsymbol{x}[0] \in \mathcal{B}_{\mathcal{F}}$ chosen at random. Given a sufficiently small step-size $\mu > 0$, for each time $n > 0$, repeat:

$$\boldsymbol{x}[n+1] = \boldsymbol{x}[n] + \mu\, \mathbf{B}\mathbf{D}\,(\boldsymbol{y}[n] - \boldsymbol{x}[n]) \qquad (12)$$

1) Mean-Square Analysis

In this section, the mean-square behavior of the proposed LMS strategy is studied, illustrating how the sampling operator D affects its stability and steady-state performance. From now on, we view the estimates $x[n]$ as realizations of a random process and analyze the performance of the LMS algorithm in terms of its mean-square behaviour. Let $\widetilde{x[n]} = x[n] - x_0$ be the error vector at time n. After subtracting x_0 and using the relation $B\widetilde{x[n]} = \widetilde{x[n]}$:

$$x\widetilde{[n+1]} = (I - \mu BDB)\widetilde{x[n]} + \mu BDv[n] \qquad (6)$$

Applying GFT $\widetilde{s[n]} = U^{\mathrm{H}}\widetilde{x[n]}$ on both sides and writing the structure of B matrix:

$$s\widetilde{[n+1]} = (I - \mu\Sigma U^{\mathrm{H}} D U \Sigma)\widetilde{s[n]} + \mu\Sigma U^{\mathrm{H}} D v[n] \qquad (7)$$

Since, not all frequency nodes are available, the above equation can be reduced using $\hat{s}[n] = U_F{}^{\mathrm{H}}\widetilde{x[n]}$:

$$\hat{s}[n+1] = (I - \mu U_F{}^{\mathrm{H}} D U_F)\hat{s}[n] + \mu U_F{}^{\mathrm{H}} D v[n] \qquad (8)$$

Using weighted mean square defined as:

$$\|\widehat{s}[n]\|^2_{\mathbf{\Phi}} = \widehat{s}[n]^H \mathbf{\Phi} \widehat{s}[n],$$

where ϕ is any complex Hermitian nonnegative-definite matrix.Then the equation becomes:

$$\mathbf{E}\|\widehat{s}[n+1]\|^2_{\mathbf{\Phi}} = \mathbf{E}\|\widehat{s}[n]\|^2_{\mathbf{\Phi}'} + \mu^2\, \mathbf{E}\{\boldsymbol{v}[n]^H \mathbf{D}\mathbf{U}_{\mathcal{F}} \mathbf{\Phi} \mathbf{U}_{\mathcal{F}}^H \mathbf{D} \boldsymbol{v}[n]\}$$
$$= \mathbf{E}\|\widehat{s}[n]\|^2_{\mathbf{\Phi}'} + \mu^2\, \mathrm{Tr}(\mathbf{\Phi} \mathbf{U}_{\mathcal{F}}^H \mathbf{D} \mathbf{C}_v \mathbf{D} \mathbf{U}_{\mathcal{F}}) \qquad (18)$$

where Tr(.) denotes the trace operator and:

$$\mathbf{\Phi}' = \left(\mathbf{I} - \mu\,\mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{U}_{\mathcal{F}}\right)\mathbf{\Phi}\left(\mathbf{I} - \mu\,\mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{U}_{\mathcal{F}}\right).$$

Let $\varphi = vec(\phi)$, where notation $vec(.)$ stacks the columns of ϕ on top of each other. Using the Kronecker product property:

$$\mathrm{vec}(\mathbf{X}\mathbf{\Phi}\mathbf{Y}) = (\mathbf{Y}^{H} \otimes \mathbf{X})\mathrm{vec}(\mathbf{\Phi}),$$

and the trace property:

$$\mathrm{Tr}(\mathbf{\Phi}\mathbf{X}) = \mathrm{vec}(\mathbf{X}^{H})^{T}\mathrm{vec}(\mathbf{\Phi}),$$

The final equation obtained is:

$$\mathrm{E}\|\widehat{s}[n+1]\|_{\varphi}^{2} = \mathrm{E}\|\widehat{s}[n]\|_{\mathbf{Q}\varphi}^{2} + \mu^{2}\mathrm{vec}(\mathbf{G})^{T}\varphi$$

where

$$\mathbf{G} = \mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{C}_{v}\mathbf{D}\mathbf{U}_{\mathcal{F}}$$
$$\mathbf{Q} = (\mathbf{I} - \mu\,\mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{U}_{\mathcal{F}}) \otimes (\mathbf{I} - \mu\,\mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{U}_{\mathcal{F}}).$$

The following theorem guarantees the asymptotic mean-square stability (i.e., convergence in the mean and mean-square error sense) of the LMS algorithm:

$Theorem 3$: [1] The LMS strategy asymptotically converges in the mean-square error sense if the sampling operator D and the step-size μ are chosen to satisfy $Therom 2$ and:

$$0 < \mu < \frac{2}{\lambda_{\max}\left(\mathbf{U}_{\mathcal{F}}^{H}\mathbf{D}\mathbf{U}_{\mathcal{F}}\right)},$$

2) Steady-State Performance

Taking the limit of n to infinity, the expression is:

$$\lim_{n\to\infty} \mathrm{E}\|\widehat{s}[n]\|_{(\mathbf{I}-\mathbf{Q})\varphi}^{2} = \mu^{2}\mathrm{vec}(\mathbf{G})^{T}\varphi.$$

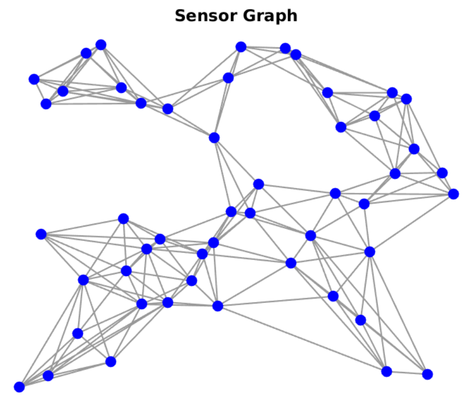
For evaluating the steady-state mean square deviation of the LMS strategy, selecting $\varphi = inv(I - Q)vec(I)$, the expression is:

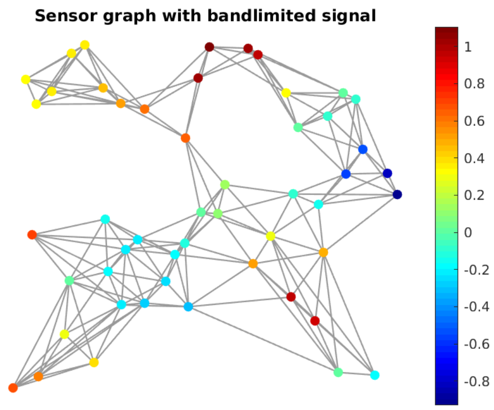
$$\mathrm{MSD} = \lim_{n\to\infty} \mathrm{E}\|\widetilde{x}[n]\|^{2} = \lim_{n\to\infty} \mathrm{E}\|\widehat{s}[n]\|^{2}$$
$$= \mu^{2}\mathrm{vec}(\mathbf{G})^{T}(\mathbf{I} - \mathbf{Q})^{-1}\mathrm{vec}(\mathbf{I}).$$

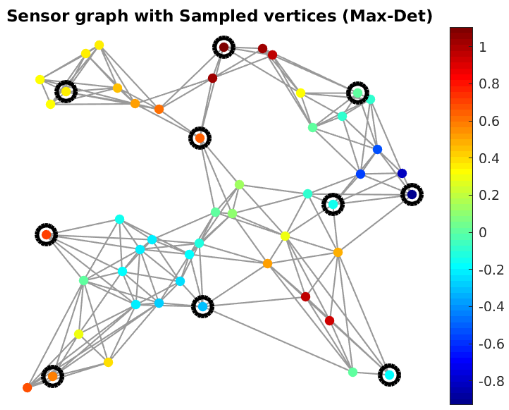
H. Results

1) Sampling Strategy

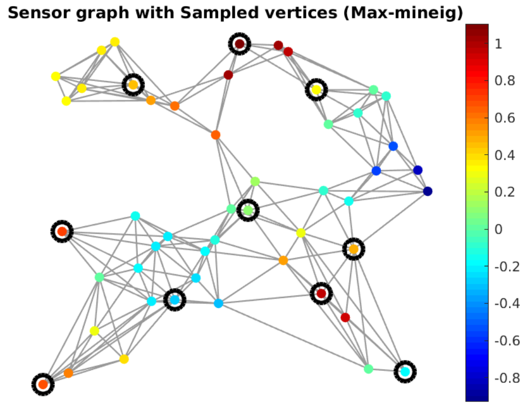
Number of nodes is 50, Bandwidth is 10 and learning rate is 0.5.



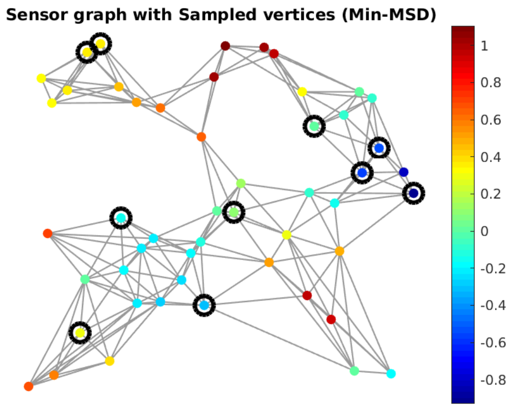





```
Sampled set of vertices for Max-Det Algorithm
     3
    48
    24
    40
    50
    44
    28
     5
     6
    25
```



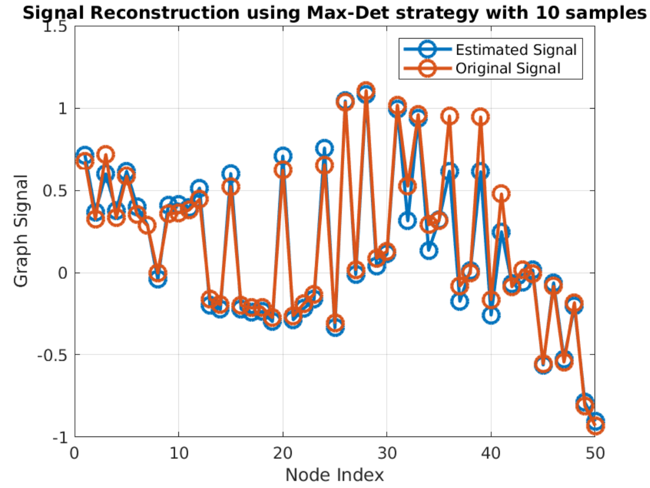

```
Sampled set of vertices for Max-mineig Algorithm
    12
    19
    48
     1
    28
    35
    41
    36
    29
     3
```




```
Sampled set of vertices for Min-MSD Algorithm
    38
    47
     7
    45
    29
    10
    25
     9
    50
    13
```

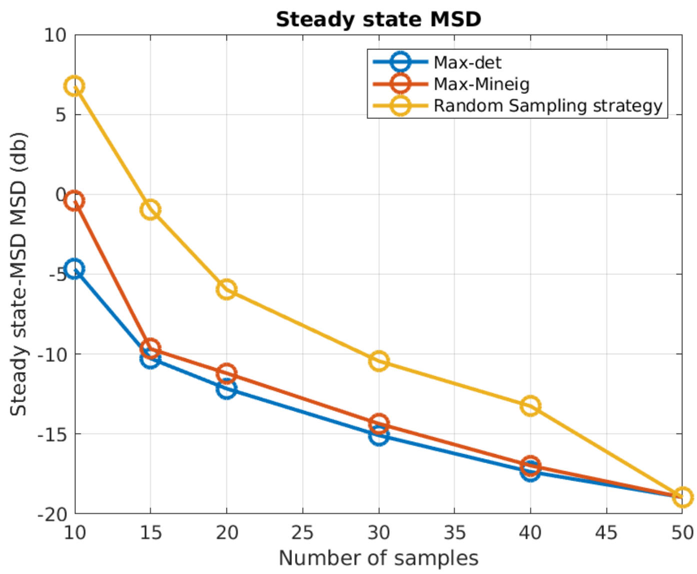
2) *Reconstructed signal*

Learning rate = 0.5, Iterations of the LMS algorithm = 100, Number of samples used to reconstruct the original signal = 10, Sampling strategy = Max-Det, Bandwidth = 10, Noise is zero mean with diagonal covariance matrix with elements uniformly random between 0 and 0.01.



3) *Effect of sampling strategies*

Learning rate = 0.5, Iterations of the LMS algorithm = 100 averaged over 200 independent simulations, Bandwidth=10, Noise is zero mean with diagonal covariance matrix with elements uniformly random between 0 and 0.01.

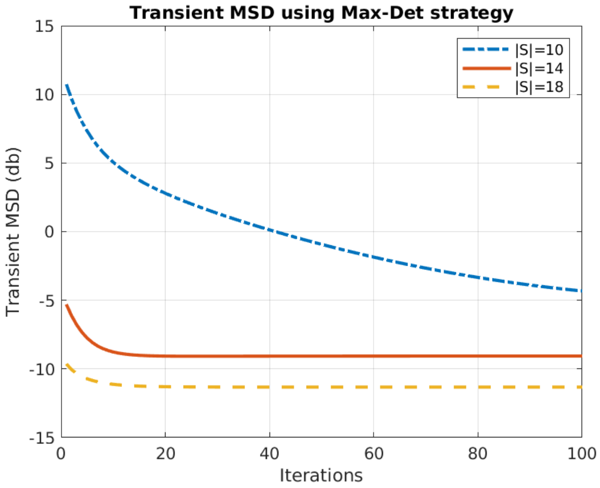


Observations:

- The LMS algorithm with random sampling can perform quite poorly, especially at low number of samples. This poor result of random sampling emphasizes that, when sampling a graph signal, what matters is not only the number of samples, but also (and most important) where the samples are taken.
- The Max-Det strategy outperforms the Max-Mineig strategy as it considers all the eigenvalues and not just one.
- The Min-MSD strategy is highly costly to compute but will give the most minimum value for some samples as it takes into account information from both graph topology and spatial distribution of the observation noise.

4) Effect of number of samples

Learning rate = 0.5, Iterations = 100 averaged over 200 independent iterations, Number of samples used for reconstruction = 10, 14, 18, Sampling strategy = Max-Det, Bandwidth = 10, Noise is zero mean with diagonal covariance matrix with elements uniformly random between 0 and 0.01.



Observation:
The error in reducing with increase in number of samples as more information is available to reconstruct the signal.

II. LMS ESTIMATION WITH ADAPTIVE GRAPH SAMPLING

A. Introduction

In many practical situations, the prior knowledge of F is unrealistic, due to the possible variability of the graph signal over time at various levels: the signal can be time varying according to a given model; the signal model may vary over time, for a given graph topology; the graph topology may vary as well over time. In all these situations, we cannot always assume that we have prior information about the frequency support F , which must then be inferred directly from the streaming data $y[n]$ in (4).

B. Model definition

The important case where the graph is fixed, and the spectral content of the signal can vary over time in an unknown manner is considered and the model is defined as:

$$y[n] = DUs_0 + Dv[n] \tag{9}$$

The problem then translates in estimating the coefficients of the GFT s_0, while identifying its support, i.e.the set of indexes where s_0 is different from zero.

C. Optimization

The overall problem can be formulated as the joint estimation of sparse representation s and sampling strategy D from the observations $y[n]$ as:

$$\min_{\boldsymbol{s}, \mathbf{D} \in \mathcal{D}} \mathbf{E}\|\boldsymbol{y}[n] - \mathbf{D}\mathbf{U}\boldsymbol{s}\|^2 + \lambda f(\boldsymbol{s}),$$

f($\cdot$) is a sparsifying penalty function (typically, l_0 or l_1 norms), and $\lambda > 0$ is a parameter that regulates how sparse we want the optimal GFT vector s.

D. Algorithm

The algorithm alternates between the optimization of the vector s and the selection of the sampling operator D.

Algorithm 2: LMS with Adaptive Graph Sampling

Start with $\boldsymbol{s}[0]$ chosen at random, $\mathbf{D}[0] = \mathbf{I}$, and $\mathcal{F}[0] = \mathcal{V}$. Given $\mu > 0$, for each time $n > 0$, repeat:

1) $\boldsymbol{s}[n+1] = T_{\lambda\mu}\left(\boldsymbol{s}[n] + \mu \mathbf{U}^H \mathbf{D}[n]\left(\boldsymbol{y}[n] - \mathbf{U}\boldsymbol{s}[n]\right)\right);$
2) Set $\mathcal{F}[n+1] = \{i \in \{1, \ldots, N\} : s_i[n+1] \neq 0\}$;
3) Given $\mathbf{U}_{\mathcal{F}[n+1]}$, select $\mathbf{D}[n+1]$ according to one of the criteria proposed in Sec. III.D;

The algorithm starts with a random initialisation of s and a full sampling for D i.e. $D = I$. First, fixing the value of the sampling operator $D[n]$ at time n, the estimate of the GFT vector s is updated using:

$$\boldsymbol{s}[n+1] = \boldsymbol{T}_{\lambda\mu}\left(\boldsymbol{s}[n] + \mu \mathbf{U}^H \mathbf{D}[n]\left(\boldsymbol{y}[n] - \mathbf{U}\boldsymbol{s}[n]\right)\right)$$

$T_\gamma(s)$ is a thresholding function that depends on the sparsity-inducing penalty f(.).

E. Thresholding functions

The aim is to is to estimate the GFT s_0 of the graph signal x_0 in (4), while selectively shrinking to zero all the components of s_0 that are outside its support, i.e., which do not belong to the bandwidth of the graph signal.

1) Lasso constraint

In this case, the vector threshold function $T_\gamma(s)$ is the component- wise thresholding function $T_\gamma(s_m)$ applied to each element of vector s, with:

$$T_\gamma(s_m) = \begin{cases} s_m - \gamma, & s_m > \gamma; \\ 0, & -\gamma \le s_m \le \gamma; \\ s_m + \gamma, & s_m < -\gamma. \end{cases}$$

The function $T_\gamma(s)$ tends to shrink all the components of the vector s and, in particular, sets to zero the components whose magnitude are within the threshold γ. Since the Lasso constraint is known for introducing a large bias in the estimate, the performance would deteriorate for vectors that are not sufficiently sparse, i.e. graph signals with large bandwidth.

2) Garotte estimator

$$T_\gamma(s_m) = \begin{cases} s_m\left(1 - \gamma^2/s_m^2\right), & |s_m| > \gamma; \\ 0, & |s_m| \le \gamma; \end{cases}$$

3) *Hard Thresholding*

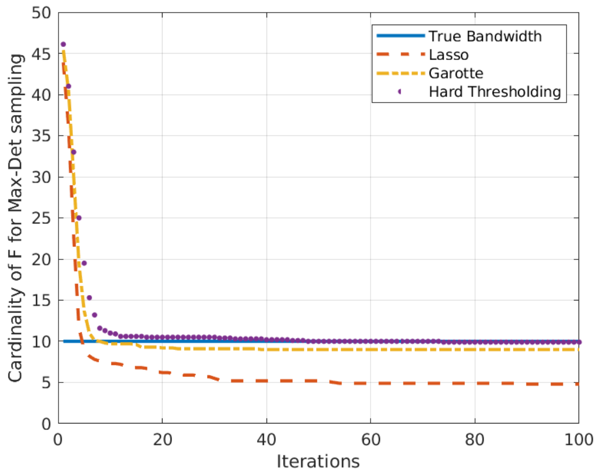
Completely removes the bias over the large components.

$$T_\gamma(s_m) = \begin{cases} s_m, & |s_m| > \gamma; \\ 0, & |s_m| \le \gamma; \end{cases}$$

F. Results

1) *Cardinality of F*

Learning rate = 0.5, Lambda= 0.1, Iterations = 100 averaged over 10 independent iterations, Sampling strategy = Max-Det, Noise is zero mean with diagonal covariance matrix with elements 0.0004.



In the Max-Det method, where the number of samples $M[n]$ to be selected at each iteration is chosen to be equal to the current estimate of cardinality of the set $F[n]$.
Observations:

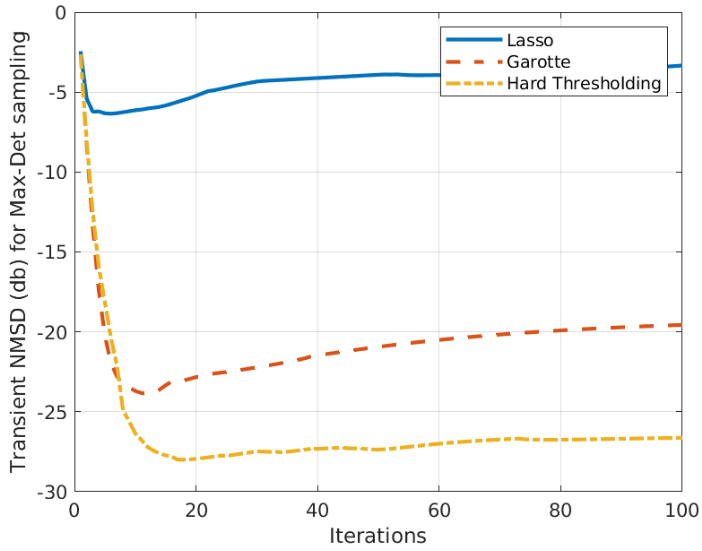
- The Garotte and hard thresholding strategies are able to learn the true bandwidth of the graph signal better than Lasso.
- Lasso strategy underestimates the bandwidth of the signal, i.e. the cardinality of the set F.

2) *Normalised MSD*

Taking the transient behaviour of normalised MSD as:

$$\mathrm{NMSD}[n] = \frac{\|s[n] - s_0\|^2}{\|s_0\|^2},$$

Learning rate = 0.5, Lambda= 0.1, Iterations = 100 averaged over 10 independent iterations, Sampling strategy = Max-Det, Noise is zero mean with diagonal covariance matrix with elements 0.0004.



Observations:

- The algorithm based on the hard thresholding function outperforms the other strategies in terms of steady-state NMSD, while having the same learning rate.
- The Garotte based algorithm has slightly worse performance with respect to the method exploiting hard thresholding, due to the residual bias introduced at large values by the thresholding function.
- The LMS algorithm based on Lasso may lead to very poor performance, due to misidentifications of the true graph bandwidth.

III. CHALLENGES

- Optimization of the Min-MSD sampling method can be done to get better results for the sampled set to get better reconstruction.
- The Algorithm can be extended to reconstruct signal when the graph is also changing.

IV. CONCLUSION

The proposed strategies are able to exploit the underlying structure of the graph signal, which can be reconstructed from a limited number of observations properly sampled from a subset of vertices, under a band-limited assumption. Furthermore, to cope with time-varying scenarios, an LMS method with adaptive graph sampling is used, which estimates and tracks the signal support in the (graph) frequency domain, while at the same time adapting the graph sampling strategy.

V. REFERENCES

[1] Adaptive Least Mean Squares Estimation of Graph Signals by Paolo Di Lorenzo, Member, IEEE, Sergio Barbarossa, Fellow, IEEE, Paolo Banelli , Member, IEEE, and Stefania Sardellitti, Member, IEEE.

VI. CODING TOOLS

- MATLAB
- GSP toolbox: https://epfl-lts2.github.io/gspbox-html/