OPTIMIZED QUANTIZATION IN DISTRIBUTED GRAPH SIGNAL PROCESSING

Isabela Cunha Maia Nobre and Pascal Frossard

Signal Processing Laboratory (LTS4), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
Email: isabela.nobre@epfl.ch, pascal.frossard@epfl.ch

ABSTRACT

Distributed graph signal processing methods require that the graph nodes communicate by exchanging messages. These messages have a finite precision in a realistic network, which may necessitate to implement quantization. Quantization, in turn, generates errors in the distributed processing tasks, compared to perfect settings. This paper proposes a novel method to minimize the quantization error without compromising the communication costs by bounding the exchanged messages along with allocating a limited bit budget through the network in an optimized way. In particular, the quantization adapts to the network topology and message importance in the iterative distributed processing algorithm. Our results show that the proposed method is efficient in minimizing the quantization error and that it outperforms baseline algorithms when the bit budget is limited.

***Index Terms*—** Graph signal processing, quantization, distributed processing, wireless sensor networks

1. INTRODUCTION

The field of signal processing on graphs provides many powerful tools to process signals in diverse applications, such as compression, denoising or reconstruction of sensor data [1]. Many of these applications require that the signal defined on a graph be processed distributively. In order to enable this distributed computation, linear graph signals operators can be approximated by shifted Chebyshev polynomials [2]. There are many other studies on distributed processing for graph signals or networked data [3, 4, 5, 6, 7], but only few deal with the fact that, in real case scenarios, the network is subject to communication constraints which limit the precision of the messages exchanged by distributed algorithms.

Some works have considered different aspects of quantization in graph signal processing. The works in [8] and [9] considered quantization in a linear prediction filter applied to graph signals using graph signal processing. The work in [10] studied the effect of quantization in the representation of graph signals, mitigating the numerical effects caused by the finite-precision machines that centrally realize the filtering process. The authors specifically designed graph filters that are robust to finite precision effects. However, they do not specifically consider distributed processing of graph signals with limited communication between nodes. There are also many works that focused on solving consensus problems in a network subject to quantized communication [11, 12, 13, 14, 15, 16], but they merely focus on average computation, and not more general processing tasks.

In this paper, we propose an adaptive quantization scheme in distributed graph signal processing tasks for minimizing the total error caused by the accumulated effects of each quantization error without compromising too much of the communication costs and energy consumption. We first propose a distributed processing algorithm where the messages exchanged by network nodes are bounded. We then compute the error due to quantization in our new algorithm and minimize it by allocating a limited bit budget in an efficient way. The performance of our quantizer is evaluated and compared to the performance of an uniform quantizer (using the same modified algorithm with bounded messages). The results show that that the bit allocation optimization improves the performance in terms of mean square error (MSE) if compared to the uniform distribution. They also show that a more uniform graph tends towards more uniform bit distribution. Also, since the errors propagate, it is more efficient to allocate more bits in the first steps of the iterative distributed processing algorithm.

Also focusing in more general processing tasks, the work [17] proposed an algorithm that learns graph dictionaries to sparsely approximate graph signals while staying robust to quantization noise. The work however doesn't fully optimize the bit allocation. Our paper focuses on the design of a quantization scheme that minimizes the quantization error in general graph signal processing tasks, by bounding the transmitted messages and by optimizing the bit allocation.

2. DISTRIBUTED GRAPH SIGNAL PROCESSING

Consider a weighted, undirected graph represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ represents a set of vertices, $\mathcal{E}$ represents a set of edges and W is the weight matrix, whose entries $W[i, j]$ typically depend on the distance between nodes i and j. The number of nodes in the graph is $N = |\mathcal{V}|$. We de-

Isabela Nobre is partially supported by CAPES (grant number 88881.174577/2018-01).

fine D as the diagonal degree matrix whose elements are the sum of each row of W. The normalized graph Laplacian operator is given by $\mathcal{L} = I - D^{-1/2}WD^{-1/2}$, is a real symmetric positive semi-definite matrix. We denote $\{\lambda_n\}_{n=0..N-1}$ as the set of eigenvalues of $\mathcal{L}$, which we order as $\{0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq .. \leq \lambda_{N-1} \leq 2\}$, and its eigenvectors as $\mathcal{X} = [\mathcal{X}_0, \mathcal{X}_1, ..., \mathcal{X}_{N-1}]$. Finally, we denote by Λ the diagonal matrix with the eigenvalues $\{\lambda_n\}_{n=0..N-1}$ on its diagonal.

A graph signal is a function $f : \mathcal{V} \longrightarrow \mathbb{R}$ defined on the vertices of the graph, which is represented by a vector $\mathbf{f} \in \mathbb{R}^N$. The graph Fourier transform of f can be defined as an expansion of the function in terms of the eigenvectors of $\mathcal{L}$, that is $\hat{f}(\lambda_n) = \sum_{i=1}^{N} f(i)\mathcal{X}_n(i)$. Given $g(\lambda)$ the transfer function of a filter, we can process the signal f_{in} by doing $f_{out} = g(\mathcal{L})f_{in}$. If $g(\mathcal{L}) = \sum_{k=0}^{K} \alpha_k \mathcal{L}^k$ is a polynomial function of order K of the Laplacian, (or if it can be approximated by a polynomial [2]), with $\{\alpha_k\}_{k=0..K}$ as the polynomial coefficients, $f_{out} = \sum_{k=0}^{K} \alpha_k \mathcal{L}^k f_{in}$ can be implemented in a distribute way, that is, each node computes its own values by exchanging messages with neighbors in K interactions. To that end, the implementation requires the computation of the different powers of the Laplacian matrix. We firstly define $z_k = \mathcal{L}^k f_{in}$ and begin with $z_0 = f_{in}$. Sensor n exchanges its value $z_0[n]$ to its one-hop neighbors of the graph. After all values of z_0 are exchanged in the network, all nodes update its component with the relation $z_1 = \mathcal{L}z_0$. To that end, each node n will only calculate its value $z_1[n]$ by doing $\mathcal{L}_n^T z_0$, where $\mathcal{L}_n$ is the line n of $\mathcal{L}$, and z_0 would be filled with the values of the messages exchanged from the neighbors of node n. Since $\mathcal{L}_n$ will be zero at the nodes that are not neighbors of n, node n doesn't need the values of z_0 at these nodes to calculate $z_1[n]$. The messages with the values of z_1 are again locally exchanged in the same way and consequently $z_2 = \mathcal{L}z_1$ is obtained in a similar fashion, and this procedure repeats until K iterations.

Finally, after knowing $\{z_0[n], z_1[n], ..z_K[n]\}$, the node n computes the value of the filtered signal in its own node using the relation [17]

$$f_{out}[n] = (g(\mathcal{L})f_{in})[n] = \sum_{k=0}^{K} \alpha_k z_k[n]. \quad (1)$$

In a more realistic setting, the messages have to be quantized before transmission. The quantized message at node n at step k can be written as $\tilde{z}_k[n] = z_k[n] + \epsilon_k[n]$, where $\epsilon_k[n]$ is the quantization error. After its transmission, the distributed update equation becomes $z_{k+1} = \mathcal{L}\tilde{z}_k[n] = \mathcal{L}(z_k[n] + \epsilon_k[n])$.

We can now make the following observation on the evolution of the quantization error with the iterations of the distributed processing algorithm. At step k, the maximum value (in an absolute sense) of the messages to be transmitted is $||\mathcal{L}^k f_{in}||_\infty$. Considering that, for $p > r > 0$, we have $\| v \|_r \geq \| v \|_p$, for any v pertaining to the vector space where these norms are defined, we can write

$$\| \mathcal{L}^k f_{in} \|_\infty \leq \| \mathcal{L}^k f_{in} \|_2 \leq \| \mathcal{L}^k \|_2 \cdot \| f_{in} \|_2, \quad (2)$$

where $\| \mathcal{L}^k \|_2$ is a matrix norm induced from the 2-norm for vectors, which can be computed by $\| \mathcal{L}^k \|_2 = \sigma_{max}(\mathcal{L}^k)$, where $\sigma_{max}(\mathcal{L}^k)$ represents the largest singular value of matrix $\mathcal{L}^k$ [18]. It corresponds to the square root of the largest eigenvalue of the positive-semidefinite matrix $(\mathcal{L}^k)^T(\mathcal{L}^k)$. Since the Laplacian matrix is diagonalizable and symmetric, we can write $(\mathcal{L}^k)^T = (\mathcal{L}^k)$. We further know that the eigenvalues of $\mathcal{L}^{2k}$ are the same as the eigenvalues of $\mathcal{L}$ to a power of $2k$. Hence, since

$$\| \mathcal{L}^k \|_2 = \sigma_{max}(\mathcal{L}^k) = \sqrt{\lambda_{max}(\mathcal{L}^{2k})} = \sqrt{\lambda_{N-1}^{2k}}, \quad (3)$$

and considering that all eigenvalues of the Laplacian are real and positive values, we finally have

$$\| \mathcal{L}^k f_{in} \|_\infty \leq \lambda_{N-1}^k \cdot \| f_{in} \|_2 . \quad (4)$$

This means that, as k increases, the transmitted messages can increase their ranges proportionally to the eigenvalues of $\mathcal{L}$, as shown in Eq. (4). This also means that, at a high value of k, the value of the respective error ϵ_k will be very high, hence increasing the value of the total error.

3. OPTIMIZED DISTRIBUTED GRAPH SIGNAL PROCESSING WITH QUANTIZATION

Based on the above observations, we first propose a modification of the classical distributed processing algorithm. Instead of using the normalized Laplacian $\mathcal{L}$ at every step of the distributed algorithm, we use $\dot{\mathcal{L}} = \mathcal{L} - I$. Hence, the eigenvalues of $\dot{\mathcal{L}}$ will be bounded in $[-1, 1]$, instead of $[0, 2]$, as it happens with $\mathcal{L}$. Therefore the values of the messages being transmitted at step k will surely not surpass the range of the original signal $y = f_{in}$, as shown in Eq. (4).

In more details, we modify the distributed algorithm of [17] as follows. Firstly we do $z_0 = f_{in}$, followed by the quantization $\tilde{z}_0 = z_0 + \epsilon_0$. The values $\tilde{z}_0$ are then exchanged by the neighbored nodes, as before. Now, instead of multiplying the received values by $\mathcal{L}$, the nodes do $\dot{z}_1 = \dot{\mathcal{L}}\tilde{z}_0$, which then is quantized $\tilde{\dot{z}}_1 = \dot{z}_1 + \epsilon_1$. The quantized value $\tilde{\dot{z}}_1$ is then exchanged, and so on, in a similar way to [17].

In a scenario without quantization, or when $\epsilon_k = 0$ for all k, we have $\dot{z}_k = \dot{\mathcal{L}}^k z_0 = (\mathcal{L} - I)^k z_0$. We observe that, at each step, we can perfectly recover $z_k = \mathcal{L}^k z_0$ from $\dot{z}_k$. Since the identity matrix commutes with all matrices, $\mathcal{L}$ and I commute, so we can use the Binomial formula and derive

$$\dot{z_k} = \left(\sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} \mathcal{L}^i \right) z_0 = \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} z_i, \quad (5)$$

or equivalently,

$$z_k = \dot{z}_k - \sum_{i=0}^{k-1} \binom{k}{i} (-1)^{k-i} z_i = \sum_{i=0}^{k} \binom{k}{i} \dot{z}_i. \quad (6)$$

Therefore, with Eq. (6), we can build a distributed algorithm where the values of $\dot{z}_k$ are exchanged, but only the values of z_k are stored.

When messages are quantized, the value of $\dot{z}_k$ becomes

$$\dot{z}_k = (\mathcal{L} - I)^k z_0 + \sum_{l=0}^{k-1} (\mathcal{L} - I)^{k-l} \epsilon_l, \qquad \text{for } k > 0. \quad (7)$$

To recover z_k from $\dot{z}_k$ we use the same process as in Eq. (6). However, the recovery is not perfect anymore due to quantization. We compute below the error due to quantization.

Combining (7) in (6), we first obtain

$$z_k = \mathcal{L}^k z_0 + \sum_{i=1}^{k} \binom{k}{i} \sum_{l=0}^{i-1} (\mathcal{L} - I)^{i-l} \epsilon_l, \qquad \text{for } k > 0. \quad (8)$$

The filtering of the graph signal f_{in} can then be written as

$$g(\mathcal{L}) f_{in} = \sum_{k=0}^{K} \alpha_k \mathcal{L}^k z_0 + \sum_{k=1}^{K} \alpha_k \sum_{i=1}^{k} \binom{k}{i} \sum_{l=0}^{i-1} (\mathcal{L} - I)^{i-l} \epsilon_l. \quad (9)$$

The total error due to quantization is given by the second term in Eq. (9), and can be rewritten as

$$Q = \sum_{k=0}^{K-1} \left[\sum_{i=1}^{K-k} \alpha_{k+1} \sum_{j=1}^{i} \binom{k+i}{k+j} (\mathcal{L} - I)^j \right] \epsilon_k. \quad (10)$$

For the sake of clarity, we now write

$$H_k = \sum_{i=1}^{K-k} \alpha_{k+1} \sum_{j=1}^{i} \binom{k+i}{k+j} (\mathcal{L} - I)^j, \quad (11)$$

and

$$F_k[n] = \left(H_k^T H_k \right)[n, n]. \quad (12)$$

We define $x[n, k]$ as the number of bits used to represent the message sent from node n at step k, with uniform quantization. Considering that the quantization step size is determined by the ratio of the total quantization range ($2 \cdot \parallel f_{in} \parallel_\infty$ for al k) over the number of quantization intervals, that is

$$\Delta[n, k] = \frac{2 \cdot \parallel f_{in} \parallel_\infty}{2^{x[n,k]}}. \quad (13)$$

It follows that the quantization error on the message transmitted by node n at step k is

$$E[\epsilon_k[n]^2] = \frac{\parallel y \parallel_\infty^2}{12} \cdot 2^{-2x[n,k]}. \quad (14)$$

Considering $\epsilon_k[n]$ and $\epsilon_p[m]$ as statistically independent for $k \neq p$ or $n \neq m$, we finally obtain the expected value of the total mean squared error as

$$E[\parallel Q \parallel^2] = \frac{\parallel y \parallel_\infty^2}{12} \sum_{k=0}^{K-1} \sum_{n=0}^{N-1} F_k[n] 2^{-2x[n,k]}. \quad (15)$$

Our objective is finally to minimize the total quantization error given the available bit budget. To that end, it is necessary to find the values of $x[k, n]$ for every combination of k and n that obey the budget constraint and minimizes Q. It can be described by following optimization problem:

$$\boxed{\begin{array}{ll} \underset{x}{\text{minimize}} & E[\parallel Q \parallel^2] \\ \text{subject to} & \displaystyle\sum_{k=0}^{K-1} \sum_{n=0}^{N-1} x[n, k] d[n] \leq B \end{array}} \quad (16)$$

Here, $d[n]$ is the degree of node n, that drives transmission costs, and B is the total bit budget constraint. Using the Lagrange Multipliers method, the solution of the optimization problem (16) will be the values of $x[n, k]$ that satisfy the equation

$$\nabla(E[\parallel Q \parallel^2]) = \gamma \nabla(\sum_{k=0}^{K-1} \sum_{n=0}^{N-1} x[n, k] d[n]), \quad (17)$$

where the gradient is computed with respect to $x[n, k]$, and γ is the Lagrange multiplier. The solution to Eq. (17) is given by

$$x[n, k] = -\frac{1}{2 \ln(2)} \ln \left(\frac{\gamma d[n]}{F_k[n]} \right). \quad (18)$$

We observe that the number of bits $x[n, k]$ in (18) depends on $d[n]$ and $F_k[n]$. As $d[n]$ grows, the communication cost in node n grows as well, since it shares its messages with more neighbors. In order to satisfy the budget constraint, $x[n, k]$ has to be smaller in that node. $F_k[n]$ is related to the topology and to the filter coefficients. From Eqs. (10), (11) and (12), we can see that it weights the contribution of each individual error $\epsilon_k[n]$ to the global error. If we have a big $F_k[n]$, it means that the relative contribution of $\epsilon_k[n]$ is big, so that its contribution needs to be reduced by increasing $x[n, k]$. Also, for the same value of n, there are more error terms in the global error computation when k is low, which means that $F_k[n]$ is high in this case. It further means that the contribution of the error terms ϵ_k in the first iterations of the distributed processing algorithm is higher. This is in agreement with the fact that the first error terms leads to higher propagation effects.

4. RESULTS

The performance of our quantizer is now evaluated and compared to the performance of a uniform quantizer. First,

$N = 50$ nodes are uniformly placed at random in unit square. The weight matrix is generated based on a thresholded Gaussian kernel function that takes into account the physical distance between nodes.

A graph signal is defined as $s = x^2 + y^2 - 1$ [2], and a noise with zero-mean normal distribution is added to it. To denoise it, the low-pass filter $g(\lambda) = \frac{\tau}{\tau+5\lambda}$ with $\tau = 2$, is used. In order to be implemented distributively, a Chebyshev polynomial approximation of order $K = 17$ is performed, and its polynomial coefficients $\{\alpha_k\}_{k=0..K}$ are determined.

The distributed graph signal processing is firstly performed without quantization. Then, the processing is performed with uniform quantization. The number of bits used to represent the transmitted messages is the same for every node n and step k. Finally, another processing is performed, this time using the optimization scheme described in this paper. In all three cases, the bounding scheme from the previous section was used, and we compare the performance of the quantization scheme to the unquantized method in terms of MSE.

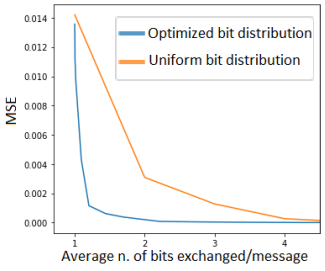

Fig. 1. MSE vs average number of bits for $N = 50$, $K = 17$, number of edges = 73 and high discrepancy between edges weights. The algorithm proposed in this paper is used.

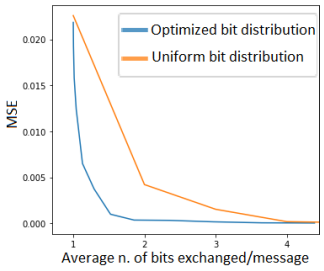

Fig. 2. MSE vs average number of bits for $N = 50$, $K = 17$, number of edges = 155 and low discrepancy between edges weights. The algorithm proposed in this paper is used.

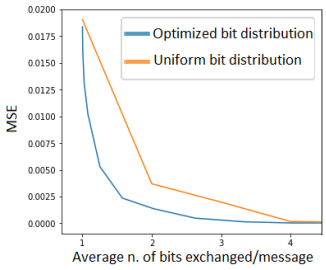

Fig. 3. MSE vs average number of bits for $N = 50$, $K = 9$, number of edges = 73 and high discrepancy between edges weights. The algorithm proposed in this paper is used.

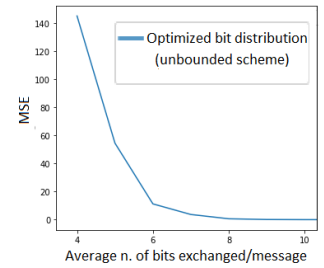

Fig. 4. MSE vs average number of bits for $N = 50$, $K = 17$, number of edges = 73 and high discrepancy between edges weights. The algorithm proposed in [17] is used.

It can be seen from Fig. 1 that the optimization of the bit allocation proposed in this paper improves the performance in terms of MSE if compared to a quantizer with the uniform bit distribution.

We can analyze the effects of the graph structure on the performance of the optimization scheme. The same processing is now applied into a graph with a more even connectivity between nodes and a lower discrepancy between edges weights, that is, a more regular graph. The result can be seen in Fig. 2.

The optimal bit allocation still brings better results, but the difference between the uniform and the optimized bit distribution in Fig. 2 is slightly smaller than in Fig. 1. It means that a more regular graph tends towards a more uniform bit distribution.

As for the impact of K on the performance, another run was done with $K = 9$ on the same graph as the one used in Fig. 1. The results are in Fig. 3.

The difference between the uniform and the optimized bit distributions in Fig. 3 is smaller than in Fig. 1. When k grows, the errors propagate and the optimized bit allocation tries to compensate that by allocating more bits in the first steps. This allocation improves the performance of the optimized scheme with respect to the uniform scheme, and this improvement effect is more visible for greater K, since the errors propagate for more iterations.

Lastly, to evaluate the efficiency of the bounding scheme, we process the signal using the distributed algorithm proposed in [17], where the messages are not bounded. Comparing Figs. 1, 2 and 3 (resulted from the modified algorithm proposed in this paper), with Fig. 4 (resulted from the baseline algorithm in [17]), we can see that, regardless of optimizing the bit distribution or not, our proposed modified algorithm yields much lower MSE values compared to the baseline algorithm. This is due to the fact that, without bounding the transmitted messages, they grow substantially and in consequence, the quantization errors grow as well.

5. CONCLUSION

In this paper, we have shown how the expected value of the square of quantization error in distributed graph signal processing tasks can be minimized by bounding the transmitted messages and by optimizing the bit allocation in the network. Experimental results show that our distributed processing algorithm substantially decreases the quantization error in regard to the baseline algorithm proposed in [17]. They also show that our bit allocation scheme further decreases the error compared to an uniform bit allocation scheme. Further investigations are needed to implement non-uniform quantization (varying quantization step size), to take into account noise and to deal with the coefficients finite precision in realistic scenarios.

6. REFERENCES

[1] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing," *arXiv preprint arXiv:1712.00468*, 2017.

[2] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via chebyshev polynomial approximation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 736–751, Dec 2018.

[3] Elvin Isufi, Andreas Loukas, Andrea Simonetto, and Geert Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[4] Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[5] Aliaksei Sandryhaila, Soummya Kar, and José MF Moura, "Finite-time distributed consensus through graph filters," in *Procedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1080–1084.

[6] Sam Safavi and Usman A Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 54–57, 2015.

[7] Xuesong Shi, Hui Feng, Muyuan Zhai, Tao Yang, and Bo Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1113–1117, 2015.

[8] Aliaksei Sandryhaila and José MF Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[9] Jiani Liu, Elvin Isufi, and Geert Leus, "Filter design for autoregressive moving average graph filters," *arXiv preprint arXiv:1711.09086*, 2017.

[10] Luiz FO Chamon and Alejandro Ribeiro, "Finite-precision effects on graph filters," in *Procedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017, pp. 603–607.

[11] Ruggero Carli, Fabio Fagnani, Paolo Frasca, and Sandro Zampieri, "Gossip consensus algorithms via quantized communication," *Automatica*, vol. 46, no. 1, pp. 70–80, 2010.

[12] Soummya Kar and José MF Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1383–1400, 2010.

[13] Akshay Kashyap, Tamer Başar, and Ramakrishnan Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[14] Shengyu Zhu, Mingyi Hong, and Biao Chen, "Quantized consensus admm for multi-agent distributed optimization," in *Procedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4134–4138.

[15] Shengyu Zhu and Biao Chen, "Quantized consensus by the admm: probabilistic versus deterministic quantizers," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1700–1713, 2016.

[16] Dorina Thanou, Effrosyni Kokiopoulou, Ye Pu, and Pascal Frossard, "Distributed average consensus with quantization refinement," *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 194–205, 2013.

[17] D. Thanou and P. Frossard, "Distributed signal processing with graph spectral dictionaries," in *Procedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 1391–1398.

[18] Carl D Meyer, *Matrix analysis and applied linear algebra*, vol. 71, Siam, 2000.