LEARNING SHEAF LAPLACIANS FROM SMOOTH SIGNALS

Jakob Hansen* *and Robert Ghrist**†

*Department of Mathematics, †Department of Electrical & Systems Engineering
University of Pennsylvania
{jhansen, ghrist}@math.upenn.edu

ABSTRACT

Cellular sheaves are a mathematical structure specifying consistency relations for data associated to vertices and edges of a graph, generalizing connection graphs and matrix weighted graphs. We consider the problem of learning such a sheaf from a collection of highly consistent or smooth signals associated to the vertices of the underlying graph.

1. INTRODUCTION

A recent theme in network-based analysis of data has been the introduction of additional data or structure to a graph. *Connection graphs* and their *graph connection Laplacians* are perhaps the most broadly studied example [1]. In this setting, an $n \times n$ orthogonal matrix ρ_{uv} is associated with each oriented edge $u \sim v$ of a weighted graph; these matrices specify local consistency relations for vector-valued data on vertices. The connection Laplacian is a symmetric positive semidefinite block matrix with diagonal block entries $L_{vv} = d_v I_n$ and off-diagonal block entries $L_{uv} = w_{uv}\rho_{uv}$ and $L_{vu} = w_{uv}\rho_{uv}^T$.

The connection Laplacian has been considered as the generator of a diffusion on the tangent bundle of a discretized manifold, yielding an embedding of the manifold [1–3]. It has also been studied in connection with synchronization problems [4], where one wishes to align certain parameters given their pairwise relationships [5–7].

Another recently studied instance of extra structure attached to a graph is the *matrix-weighted graph* [8, 9]. For these, an $n \times n$ symmetric positive semidefinite matrix W_{uv} is assigned to each edge $u \sim v$ of a graph, representing a generalization of standard nonnegative weights. The Laplacian of a matrix-weighted graph is a block matrix with block diagonal entries $L_{vv} = \sum_{u \sim v} W_{uv}$ and block off-diagonal entries $L_{uv} = -W_{uv}$.

Various authors have considered the problem of learning graphs from signals supported on their vertices [10–13]. However, to our knowledge, the analogous problem of learning these additional structures from data has not been addressed.

This work was supported with funding from DARPA and the AFRL under agreement number FA8650-18-2-7840.

As an example of a situation in which learning higher algebraic structure may be useful, consider the multireference alignment problem. Here we have noisy copies of the same signal, permuted cyclically, which we wish to align for averaging. One approach—angular synchronization—computes the best shift aligning each pair of signals and constructs a connection graph representing these shifts, from which a global alignment is extracted. Rather than using only local information to construct this graph, we might wish to learn the graph from the global structure of the data, choosing the connection graph is most consistent with the observed signals. This approach would additionally allow identification of clusters of distinct signals. Such an approach would be similar to, but motivated differently from, the one proposed by Bandeira et al in [14].

In this paper, we propose a framework for learning various types of graphs with extra structure from data. The structures we consider fall under the category of *cellular sheaves*, generalizing both connection graphs and matrix-weighted graphs.

1.1. Cellular Sheaves

A cellular sheaf $\mathcal{F}$ associated to a graph G is specified by the following data:

- A vector space $\mathcal{F}(v)$ for each vertex v of G;
- A vector space $\mathcal{F}(e)$ for each edge e of G;
- A linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident vertex-edge pair $v \trianglelefteq e$, called a *restriction map*.

The vector spaces $\mathcal{F}(v)$ and $\mathcal{F}(e)$ are called the *stalks* over v and e. There are two natural vector spaces associated with a cellular sheaf, given by the direct sums of stalks over vertices and over edges, respectively, denoted $C^0(G;\mathcal{F}) = \bigoplus_{v \in V(G)} \mathcal{F}(v)$ and $C^1(G;\mathcal{F}) = \bigoplus_{e \in E(G)} \mathcal{F}(e)$. Vectors in $C^0(G;\mathcal{F})$ may be thought of as signals associated with the vertices of G, while vectors in $C^1(G;\mathcal{F})$ may be thought of as signals on edges, or as local discrepancies between assignments to vertices.

A *section* of $\mathcal{F}$ is a globally consistent choice of data on vertex stalks over G — a choice $x_v \in \mathcal{F}(v)$ for all $v \in V(G)$ such that for every edge $e = u \sim v$, $\mathcal{F}_{u \trianglelefteq e} x_u = \mathcal{F}_{v \trianglelefteq e} x_v$. Thus, a sheaf specifies local linear relationships for consistency of data, and sections of the sheaf are globally defined data which respect all local consistency relationships. The sections of a sheaf form a vector space, denoted $H^0(G;\mathcal{F})$, which is a subspace of $C^0(X;\mathcal{F})$. It is the kernel of the *coboundary* map $\delta : C^0(X;\mathcal{F}) \to C^1(X;\mathcal{F})$, which is given on an oriented edge $e = u \sim v$ by $(\delta x)_e = \mathcal{F}_{u \trianglelefteq e} x_u - \mathcal{F}_{v \trianglelefteq e} x_v$. Given a choice of bases for vertex and edge stalks, δ has a natural representation as a block matrix where each block row has two nonzero blocks.

1.2. The Sheaf Laplacian

The matrix δ behaves similarly to the transpose of the signed incidence matrix in spectral graph theory. In particular, it is possible to use δ to construct a Laplacian matrix associated with a sheaf. This is the matrix $L_{\mathcal{F}} = \delta^T \delta$. This matrix has a block structure, with blocks indexed by vertices of G and block sizes determined by the dimensions of $\mathcal{F}(v)$. The off-diagonal block entries associated with an edge $e = u \sim v$ are given by $L_{uv} = -\mathcal{F}_{u \trianglelefteq e}^T \mathcal{F}_{v \trianglelefteq e}$, while the diagonal block entries are $L_{vv} = \sum_{e : v \trianglelefteq e} \mathcal{F}_{v \trianglelefteq e}^T \mathcal{F}_{v \trianglelefteq e}$. L is clearly positive semidefinite, and its kernel is the same as the kernel of δ, which is the space of sections of $\mathcal{F}$.

The study of sheaf Laplacians subsumes the study of graph Laplacians and their more recent variants. The weighted graph Laplacian is the sheaf Laplacian of a sheaf with all stalks $\mathbb{R}$ and restriction maps $\mathcal{F}_{u \trianglelefteq e} = \mathcal{F}_{v \trianglelefteq e} = \sqrt{w_{uv}}$ for each edge $e = u \sim v$. To see that a graph connection Laplacian is in fact a sheaf Laplacian, let $\mathcal{F}(v) = \mathcal{F}(e) = \mathbb{R}^n$ for all vertices and edges, and for an (oriented) edge $e = u \sim v$, let $\mathcal{F}_{u \trianglelefteq e} = \sqrt{w_{uv}} I_n$ and $\mathcal{F}_{v \trianglelefteq e} = \sqrt{w_{uv}} \rho_{uv}$. Similarly, the Laplacian of a matrix-weighted graph is the Laplacian of a cellular sheaf. Here we may choose any factorization of the weight matrices $W_{uv} = \rho_{uv}^T \rho_{uv}$; the Cholesky factorization or the eigendecomposition serve well. We then simply let $\mathcal{F}_{u \trianglelefteq e} = \mathcal{F}_{v \trianglelefteq e} = \rho_{uv}$.

Indeed, a cellular sheaf may be thought of as combining the rotational aspects of a connection graph with the weighting aspects of a matrix-weighted graph. However, the interactions of these two aspects can lead to more complicated behavior than either alone. For instance, one may interpret sheaves with all stalks $\mathbb{R}$ in a dynamic sense: each edge has a gain, which pushes the values observed at its head to be a scalar multiple of the values at its tail, as well as a weight, giving the strength of the force imposing this tendency.

The study of the sheaf Laplacian falls under the purview of *spectral sheaf theory* [15], which generalizes certain aspects of spectral graph theory to cellular sheaves.

1.3. Smooth Signals on a Sheaf

The graph Laplacian is used to quantify a notion of *smoothness* for signals on the vertices of a graph [16]. A signal x with $x^T L x$ small is considered smooth, because it has small variation over edges. The smoothest signals on a graph are therefore the scalar multiples of the constant vector $\mathbf{1}$. Similarly, the sheaf Laplacian quantifies smoothness of sheaf-valued signals on graph with respect to the consistency relations specified by the sheaf. A signal $x \in C^0(G;\mathcal{F})$ with $x^T L_{\mathcal{F}} x$ small is smooth, because it is nearly consistent, and the smoothest sheaf-valued signals are the sections of $\mathcal{F}$.

Various authors have considered the question of recovering a graph from a collection of signals known to be smooth on the graph [10–13]. Given a matrix X whose columns are the smooth signals, the problem considered is usually one of the form

$$\min_{L \in \mathcal{L}} \; g(L, X) + f(L), \tag{1}$$

where $\mathcal{L}$ is the set of all graph Laplacians, g is a cost function enforcing the constraint that L represent a graph on which the signals in X are smooth, and f is a cost function enforcing structure such as sparsity and connectivity of the graph. A particularly simple and effective choice for g is $g(X, L) = \operatorname{tr}(X^T L X)$, which is linear in L and hence easy to optimize.

In this paper, we consider the generalization of this optimization framework to sheaf Laplacians, thus broadening the class of structures that can be learned from smooth signals.

2. CONES OF SHEAF LAPLACIANS

A key part of the solution of problem (1) is the restriction of the optimization domain to the set of graph Laplacians. This set is easy to define: they are the symmetric matrices with nonnegative diagonal entries and row sum zero. Graph Laplacians form a convex cone that lies within the cone of symmetric positive semidefinite (SPSD) matrices.

In order to solve an analogous problem for sheaf Laplacians, we must understand the corresponding domain of interest. The set of sheaf Laplacians with constant-dimensional vertex stalks is also a convex cone lying within the SPSD cone. Its description is complicated both by the additional degrees of freedom allowed by sheaf Laplacians and the block structure of the matrices.

From the formula $L = \delta^T \delta$, we see that L is the sum of matrices L_e, one corresponding to each edge. The edge $e = u \sim v$ contributes a block-sparse SPSD matrix L_e, with nonzero entries in the blocks (u,u), (u,v), (v,u), and (v,v). It is not hard to see that by choosing the restriction maps $\mathcal{F}_{u \trianglelefteq e}$ and $\mathcal{F}_{v \trianglelefteq e}$ carefully, we can make L_e equal to any positive semidefinite matrix that has the appropriate sparsity pattern. As a result, we can check whether a matrix L is a sheaf Laplacian for a given choice of vertex stalk dimensions via a semidefinite feasibility program:

$$\begin{aligned} L &= \sum_{1 \le i < j \le n} M_{ij} \\ \text{s.t. } & M_{ij} \succeq 0 \\ & M_{ij}(s,t) = 0 \text{ for } s,t \neq i,j. \end{aligned}$$

When the vertex stalks are one-dimensional, this can be converted to a second-order cone program. This reformulation has been used in [17] to produce a subset of the semidefinite cone for which membership can be efficiently checked. We will denote the cone of sheaf Laplacians on graphs with n vertices and d-dimensional vertex stalks $\mathcal{L}_{\text{sheaf}}(n, d)$.

The common use of more particular subclasses of matrices such as connection Laplacians suggests optimizing over smaller cones of matrices. The set of connection Laplacians is not a convex cone, since convex combinations of orthogonal matrices are not orthogonal. However, a simple convex cone that contains the set of connection Laplacians is the cone of sheaf Laplacians whose diagonal blocks are scalar multiples of the identity. We will denote this cone $\mathcal{L}_{\text{CL}}(n, d)$, where n is the number of vertices and d the block size.

The cone of Laplacians of matrix-weighted graphs is also simpler to define. Here the SPSD matrices in the sum forming L have nonzero entries given by

$$\begin{bmatrix} S_{ij} & -S_{ij} \\ -S_{ij} & S_{ij} \end{bmatrix},$$

where S_{ij} is an SPSD matrix. This reduces the size of the semidefinite program defining the cone by a factor of four, since we only need to consider a matrix of one-fourth the size for each edge. This cone will be denoted $\mathcal{L}_{\text{MW}}(n, d)$.

3. LEARNING SHEAVES FROM SIGNALS

We now consider the problem of recovering the sheaf Laplacian from sampled smooth signals. To do this, we minimize the total energy of the sampled signals over all possible sheaf Laplacians. This is the problem

$$\min_{L \in \mathcal{L}_{\text{sheaf}}} \operatorname{tr}(X^T L X) + f(L), \tag{2}$$

where f enforces connectivity and sparsity. Here we consider different functional forms for f. We will write $f(L) = \alpha f_c(L) + \beta f_s(L)$, with f_s controlling sparsity and f_c encouraging connectivity.

A common way to encourage connectivity of a graph is to put a lower barrier on the degrees of vertices. However, the cone of sheaf Laplacians contains block diagonal matrices, so this constraint does not perfectly enforce connectivity for sheaves. We let

$$f_c(L) = -\sum_i \log(\operatorname{tr}(L_{ii})),$$

thus requiring that each diagonal block have at least one nonzero diagonal entry. Note that since the diagonal blocks are positive semidefinite, the trace is equal to the nuclear norm, and we can view this term as putting a barrier on the nuclear norm of the diagonal blocks.

The objective without $f_s(L)$ already encourages sparsity of L, since we are minimizing a linear function within a particular cone. Any summand L_e in the decomposition of L that would increase $\operatorname{tr}(X^T L X)$ will not appear in the solution. Only those blocks that decrease the overall energy of the vectors in X will appear in the solution. Thus, rather than adding a term to encourage sparsity, we add a regularization term to counterbalance this tendency toward sparsity. This term is

$$f_s(L) = \sum_{i<j} \|L_{ij}\|_F^2,$$

where i and j index blocks of L. As an L^2-type norm, this encourages uniformity in the size of off-diagonal entries. These particular functional forms for f_c and f_s are analogous to ones introduced and tested by Kalofolias [12] in the setting of graph Laplacians.

4. NUMERICAL EXPERIMENTS

To evaluate this method numerically, we produced random graphs, constructed sheaves on these graphs, and then attempted to recover the sheaf Laplacian from smooth signals on the sheaf. Smooth signals were obtained by sampling random Gaussian vectors and then smoothing them according to their expansion in terms of the eigenvectors of the sheaf Laplacian L_0. This may be interpreted as a filtering step in the Fourier domain associated with the sheaf Laplacian. If $x = \sum_i a_i v_i$, where v_i are the eigenvectors of $L_{\mathcal{F}}$, we smooth x by multiplying the coefficients a_i by a filter function $\phi(\lambda_i)$, i.e. $\tilde{x} = \sum_i \phi(\lambda_i) a_i v_i$. We use the filter function $\phi(\lambda) = \frac{1}{1+10\lambda}$ inspired by the smoothing that occurs in Tikhonov regularization, where we have scaled L_0 so that its eigenvalues lie in $[0, 1]$. This smoothing method and others were considered in [12]. After normalizing the smoothed vectors, we add Gaussian noise with a standard deviation of $\sigma = 10^{-2}$.

We then attempt to recover L from the sampled data X by following the optimization framework described in section 3. We perform this optimization over several cones: the cone $\mathcal{L}_{\text{sheaf}}$ of sheaf Laplacians, the cone $\mathcal{L}_{\text{MW}}$ of Laplacians of matrix weighted graphs, the cone $\mathcal{L}_{\text{conn}}$ containing graph connection Laplacians, and finally, as controls, the cones $\mathcal{L}_{\text{graph}}$ and $\mathcal{S}$ of graph Laplacians and positive semidefinite matrices. Once a Laplacian matrix is recovered, it is rescaled to have minimal Frobenius distance from the original matrix. This allows our recovery method to be invariant to changes in the hyperparameters that hold α/β constant.

The random graphs were of Erdos-Renyi type with probability parameter just above the connectivity threshold, $p = 1.1 \log(N_v)/N_v$. To generate sheaves over the graphs, we used three methods of choosing arbitrary restriction maps. One is to choose random matrices with Gaussian distributions for their entries for each restriction map; this can produce an arbitrary sheaf. We also performed the same randomization but forced the two matrices corresponding to a given edge to be equal; this produces a matrix-weighted graph. Finally, we constructed sheaves whose restriction maps were randomly chosen orthogonal matrices, yielding connection Laplacians.

To the best of our knowledge, no framework has previously been proposed to recover any of the classes of Laplacians considered here from smooth sampled signals. As a result, we compare to optimization over the cone of graph Laplacians and the cone of positive semidefinite matrices as baselines for performance.

Results are contained in Table 1, giving average relative L^1 and L^2 elementwise error in the recovered Laplacian and adjacency matrices, as well as the F-score (harmonic mean of precision and recall) for the sparsity pattern of off-diagonal blocks. Results are averaged over 20 trials. We tested sheaves with 1- and 2-dimensional stalks, with $N_v = 100$ and $N_v = 50$ vertices, respectively, to give Laplacians of constant size 100×100. The results reported are those corresponding to the optimal parameter of β for the L^1 error, obtained via a grid search.

5. DISCUSSION

Optimizing over the appropriate cone outperforms optimization over the cone of graph Laplacians or the semidefinite cone. This is particularly true when considering the sparsity measures and the SPSD cone. If we suspect our data has richer structure than can be described by graph Laplacians, but is still generated by pairwise interactions, it is appropriate to optimize over the cone of sheaf Laplacians. Interestingly, optimizing over $\mathcal{L}_{\text{sheaf}}$ gave better results than optimizing over $\mathcal{L}_{\text{CL}}$ even when matrices were sampled from the set of connection Laplacians.

While the overall error rates are high, the recovered matrices exhibit a remarkable amount of qualitative similarity to the original matrices. In particular, the sparsity patterns of the learned matrices are fairly precise predictors of the true sparsity pattern: a nonzero block in the learned matrix is a strong indicator that the corresponding block in the original matrix is nonzero.

It should be noted that when stalks are one-dimensional, matrix weighted graphs are simply weighted graphs, and connection Laplacians correspond to Laplacians of signed graphs.[1] Thus, the results for matrix-weighted graphs with 1-dimensional stalks are quite similar to those for graphs.

The relative error in the adjacency matrices is significantly higher than the relative error for the Laplacians. Part of this is due to the fact that our output matrices are rescaled to minimize the distance between the Laplacians, and part of this is due to the optimization objective itself. The diagonal is much more informative for the class of sheaf Laplacians than it is for graph Laplacians, as it is not recoverable from the adjacency matrix alone. This means that in many situations the Laplacian error is the most appropriate metric.

[1]There are two definitions of signed Laplacians; the appropriate one in this context is the one given by changing only the signs of off-diagonal entries of the standard graph Laplacian. This ensures positive semidefiniteness of the signed Laplacian.

It would be interesting to study the recovery of sheaf Laplacians from a statistical perspective, as done in [10] and [13] by consideration of Gaussian Markov random fields (GMRFs) and inverse covariance matrices. Sheaf Laplacians with one-dimensional stalks correspond to *pairwise normalizable* GMRFs [18], where the probability density function can be factored as a product of Gaussian densities each of which depends on only two coordinates. Sheaf Laplacians with higher-dimensional stalks would correspond to a blockwise version of this condition, which to our knowledge has not yet been studied. The algorithm we have proposed is preliminary, and we believe many improvements may be made to its performance and analysis.

Cellular sheaves offer opportunities for extending graph signal processing to richer forms of data and more complex interaction schemes. Further extensions are possible by combining sheaf theory with the recently introduced field of signal processing on simplicial complexes [19, 20]. Such advances will allow understanding of new types of data and relationships between them.

Table 1. Error rates for sheaf recovery

	Gaussian		**Connection**		**Matrix-Weighted**	
	1	2	1	2	1	2
Sheaf						
L^1 L	0.3568	0.5178	0.2869	0.4128	0.3676	0.4514
L^2 L	0.2531	0.4461	0.2139	0.3035	0.3094	0.3840
L^1 A	0.6571	0.6752	0.5097	0.5488	0.5794	0.5348
L^2 A	0.5451	0.6091	0.4850	0.5172	0.4992	0.4863
F	0.5053	0.6414	0.6965	0.7179	0.5596	0.7563
CL						
L^1 L	0.5435	0.7150	0.5711	0.6362	0.6294	0.7794
L^2 L	0.4235	0.5640	0.4397	0.4678	0.5626	0.6438
L^1 A	1.0165	1.0315	1.0260	1.0410	1.0289	1.0461
L^2 A	1.0027	1.0068	1.0063	1.0111	1.0042	1.0096
F	0.1226	0.1885	0.1280	0.1882	0.1177	0.1789
MW						
L^1 L	0.7209	0.9409	0.5780	0.8242	0.3615	0.4435
L^2 L	0.4108	0.6479	0.3647	0.6354	0.3064	0.3762
L^1 A	1.5512	1.5382	1.0597	0.9838	0.5601	0.5296
L^2 A	1.0697	1.2882	0.8332	0.9759	0.4740	0.4791
F	0.2439	0.2803	0.2905	0.5553	0.5094	0.7478
Graph						
L^1 L	0.9098	0.8668	0.6920	0.6967	0.4468	0.5984
L^2 L	0.5724	0.5253	0.4901	0.4832	0.3303	0.4635
L^1 A	1.5081	1.3677	1.0101	0.9752	0.6454	0.7491
L^2 A	0.9895	0.9550	0.8283	0.8330	0.4843	0.6343
F	0.1861	0.2709	0.3288	0.3466	0.4701	0.4090
SPD						
L^1 L	0.5443	0.6636	0.5701	0.6253	0.6317	0.7251
L^2 L	0.4213	0.5033	0.4321	0.4601	0.5567	0.5990
L^1 A	1.0190	1.0133	1.0247	1.0169	1.0317	1.0208
L^2 A	0.9946	0.9943	0.9875	0.9890	0.9917	0.9909
F	0.2440	0.2554	0.2470	0.3508	0.1456	0.2104

6. REFERENCES

[1] Amit Singer and Hau-Tieng Wu, "Vector Diffusion Maps and the Connection Laplacian," *Communications in Pure and Applied Mathematics*, vol. 65, no. 8, 2012.

[2] Noureddine El Karoui and Hau-Tieng Wu, "Graph Connection Laplacian Methods can be Robust to Noise," *Annals of Statistics*, vol. 44, no. 1, pp. 346–372, 2016.

[3] Hau-Tieng Wu, "Embedding Riemannian manifolds by the heat kernel of the connection Laplacian," *Advances in Mathematics*, vol. 304, pp. 1055–1079, Jan. 2017.

[4] Afonso Bandeira, *Convex Relaxations for Certain Inverse Problems on Graphs*, Ph.D. thesis, Princeton University, 2015.

[5] Amit Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and Computational Harmonic Analysis*, vol. 30, pp. 20–36, 2011.

[6] Afonso S. Bandeira, Amit Singer, and Daniel A. Spielman, "A Cheeger Inequality for the Graph Connection Laplacian," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 4, pp. 1611–1630, 2013.

[7] Tingran Gao, Jacek Brodzki, and Sayan Mukherjee, "The Geometry of Synchronization Problems and Learning Group Actions," *arXiv:1610.09051*, 2016.

[8] S. Emre Tuna, "Synchronization under matrix-weighted Laplacian," *Automatica*, vol. 73, pp. 76–81, Nov. 2016.

[9] Minh Hoang Trinh, Chuong Van Nguyen, Young-Hun Lim, and Hyo-Sung Ahn, "Matrix-weighted consensus and its applications," *Automatica*, vol. 89, pp. 415–419, Mar. 2018.

[10] Brenden Lake and Joshua Tenenbaum, "Discovering Structure by Learning Sparse Graphs," *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 32, no. 32, 2010.

[11] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 3736–3740, IEEE.

[12] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, Cadiz, Spain, 2016.

[13] 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*, vol. 11, no. 6, pp. 825–841, Sept. 2017.

[14] Afonso S. Bandeira, Moses Charikar, Amit Singer, and Andy Zhu, "Multireference Alignment Using Semidefinite Programming," in *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, New York, NY, USA, 2014, ITCS '14, pp. 459–470, ACM.

[15] Jakob Hansen and Robert Ghrist, "Toward a Spectral Theory of Cellular Sheaves," *arXiv:1808.01513 [math]*, Aug. 2018.

[16] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[17] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization," in *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2014, pp. 1–5.

[18] Dmitry M Malioutov, Jason K Johnson, and Alan S Willsky, "Walk-Sums and Belief Propagation in Gaussian Graphical Models," *Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.

[19] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 6425–6429.

[20] Sergio Barbarossa, Stefania Sardellitti, and Elena Ceci, "Learning from Signals Defined over Simplicial Complexes," in *IEEE Data Science Workshop*, Lausanne, Switzerland, 2018, p. 5.