LEARNING FROM SIGNALS DEFINED OVER SIMPLICIAL COMPLEXES

Sergio Barbarossa, Stefania Sardellitti, and Elena Ceci

Sapienza University of Rome, DIET Dept., Via Eudossiana 18, 00184, Rome, Italy
e-mail: {sergio.barbarossa, stefania.sardellitti, elena.ceci}@uniroma1.it

ABSTRACT

In the last years, several new tools have been devised to analyze signals defined over the vertices of a graph, i.e. over a discrete domain whose structure is described by *pairwise* relations. In this paper, we expand these tools to the analysis of signals defined on simplicial complexes, whose domain has a structure specified by various *multi-way* relations. Within this framework, we show how to filter signals and how to reconstruct edge and vertex signals from a subset of observations. Finally, we propose two alternative algorithms to infer the structure of the simplicial complex from the observations.

Index Terms— Algebraic topology, topological data analysis, topology inference.

1. INTRODUCTION

Processing signals defined over a graph is a topic that has attracted a lot of research in the last years, because of the multiple applications as well as for the inherent theoretical challenges [1]. One of the interesting aspects of graph signal processing is that the analysis tools, like the Graph Fourier Transform for example, depend on the graph topology. Graphs can be seen as a set of points having a structure specified in terms of their *pairwise* relations, represented as edges of the graph. In this paper, we are interested in moving forward by considering signals defined over discrete domains, whose structure is specified in terms of *multiway* relations of various order. In other words, given a set of points (vertices), the signal domain is the ensemble of subsets of these points. The subsets may have different size (number of elements). A formal way to characterize the domain and its intrinsic structure is to resort to algebraic topology tools [2]. In this work, we characterize the domain in terms of simplicial complexes and define signals as the mapping from every subset of points to the real domain. Straightforward examples of applications are signals defined over the edges (pairs of vertices) of a graph, like flow of vehicles or data traffic over the Internet network, but the approach is described in the most general terms, to be applicable in principle to any dimension of the subsets.

This work has been supported by H2020 EU/Japan Project 5G-MiEdge Nr. 723171.

The use of algebraic topology tools for the extraction of information from data is not new, as it is the primary purpose of *topological data analysis*, see e.g. [3]. Interesting applications of algebraic topology tools have been proposed to control systems [4], statistical ranking from incomplete data [5], distributed coverage control of sensor networks [6], [7], [8], wheeze detection [9]. The extraction of topological invariants to provide qualitative information about signal is the subject of topological signal processing [10]. In this work, we wish to propose methods for analyzing signals defined over simplicial complexes, based on the eigen-decomposition of the higher order Laplacians. We consider sampling and recovery of signals defined over simplicial complexes of various order. Then, we propose two methods to infer the structure of the simplicial complex from (possibly noisy) measurements.

2. REVIEW OF ALGEBRAIC TOPOLOGY TOOLS

In this section we recall the basic principles of algebraic topology [2] and discrete calculus [11], as they will form the basis for deriving the basic signal processing tools to be used in later sections. We start describing the structural properties of discrete domains and then we will recall the description of discrete forms, i.e. signals defined over discrete domains. The starting point is a *discrete* domain composed of a set of points having a structure represented by the neighborhood relations between them. Given a finite set V of N points (vertices) $\{v_0, \dots, v_{N-1}\}$, a k-simplex is an unordered set of $k+1$ points, say $\{v_0, \dots, v_k\}$. A *face* of the k-simplex $\{v_0, \dots, v_k\}$ is a $(k-1)$-simplex of the form $\{v_0, \dots, v_{i-1}, v_{i+1}, \dots, v_k\}$, for some $0 \leq i \leq k$. Every k-simplex has exactly $k+1$ faces. An *abstract simplicial complex* X is a finite collection of simplices that is closed with respect to inclusion of faces, i.e., if $\sigma \in X$, then all faces of σ also belong to X. In particular, the intersection of any two sets in X is also a set in X, and the empty set is an element of every simplicial complex. The dimension of a simplex is one less than its cardinality. Then, every vertex is a 0-dimensional simplex, an edge has dimension 1, and so on. The dimension of a simplicial complex is the largest dimension of any simplex. The subsets of a simplex are called its faces. A graph is a particular case of an abstract simplicial complex containing only simplices of order 0 (vertices) and

1 (edges). As mentioned before, the interesting aspect of a complex is its structure, which is given by the neighborhood relations of its subsets. A $(k-1)$-face σ_j^{k-1} of a k simplex σ_i^k is called a boundary element of σ_i^k.

The structure of a k-complex can be described algebraically defining the *incidence* matrices of various order. As with graphs, the definition of an incidence matrix requires the introduction of an orientation. Given a simplex $\sigma_i^k = \{v_0, \dots, v_k\}$ with no orientation, a possible corresponding oriented simplex is denoted as $[v_0, \dots, v_k]$. For each k, $C_k(X, \mathbb{R})$ denotes the vector space obtained by the linear combination, using real coefficients, of the set of oriented k-simplices of X. By definitions, in algebraic topology, the elements of $C_k(X, \mathbb{R})$ are called *k-chains*. If $\{\sigma_1^k, \dots, \sigma_{n_k}^k\}$ is the set of k-simplices in X, a k-chain τ_k can be written as $\tau_k = \sum_{i=1}^{n_k} \alpha_i \sigma_i^k$. Then, given the basis $\{\sigma_1^k, \dots, \sigma_{n_k}^k\}$, a chain τ^k can be represented by the vector of its expansion coefficients $(\alpha_1, \dots, \alpha_{n_k})$.

An abstract simplicial complex is just an ensemble of subsets of a finite set. They have a geometric counterpart, denoted as *geometric simplicial complexes*, defined as follows. A set of points in a real space $\mathbb{R}^D$ of dimension D is *affinely independent* if it is not contained in a hyperplane; an affinely independent set in $\mathbb{R}^D$ contains at most $D+1$ points. A geometric k-simplex is the *convex hull* of a set of $k+1$ affinely independent points, called its vertices. So, a tetrahedron is a 3-simplex, a triangle is a 2-simplex, a line segment is a 1-simplex, a point is a 0-simplex.

An important operator acting on ordered chains is the *boundary operator*. The boundary of the ordered k-chain $[v_0, \dots, v_k]$ is

$$\partial_k [v_0, \dots, v_k] := \sum_{j=0}^{k} (-1)^j [v_0, \dots, v_{j-1}, v_{j+1}, \dots, v_k]. \tag{1}$$

Intuitively speaking, the boundary operator maps a k-chain to its faces. It is straightforward to verify, by simple substitution, that the boundary of a boundary is zero, i.e., $\partial_k \partial_{k+1} = 0$. Given a finite set of oriented simplices, their neighborhood relations can be encoded in the so called *incidence matrix* $\mathbf{B}_k^T$, which establishes which k-simplices are incident to which $(k-1)$-simplices and its entries are defined as follows: 1) $B_k(i,j) = 0$, if σ_i^{k-1} is not on the boundary of σ_j^k; 2) $B_k(i,j) = 1$, if σ_i^{k-1} is a boundary element of σ_j^k and its orientation is the same as σ_j^k; 3) $B_k(i,j) = -1$, if σ_i^{k-1} is a boundary element of σ_j^k and its orientation is the opposite of σ_j^k. It is easy to verify from (1) that the boundary of a boundary is zero. This property is expressed in matrix form as

$$\boldsymbol{B}_k \boldsymbol{B}_{k+1} = \mathbf{0}. \tag{2}$$

The structure of a K-simplicial complex is fully described by its k-th order Laplacian matrices, with $k = 0, \dots, K$, defined as

$$\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^T, \tag{3}$$

$$\mathbf{L}_k = \mathbf{B}_k^T \mathbf{B}_k + \mathbf{B}_{k+1} \mathbf{B}_{k+1}^T, k = 1, \dots K-1, \tag{4}$$

$$\mathbf{L}_K = \mathbf{B}_K^T \mathbf{B}_K. \tag{5}$$

Graph-based learning methods are deeply based on the eigendecomposition of $\boldsymbol{L}_0$. In particular, if a graph has a *modular* structure, i.e. it is composed of a set of clusters, then the eigenvectors of $\boldsymbol{L}_0$ associated with the smallest eigenvalues reflect the clustering properties of the graph. They are in fact smoothly varying within each cluster, whereas they assume different sign over different clusters or may be zero within some cluster. We can then expect that the eigendecompostion of the higher order Laplacian will also carry important information about higher order chains. Interestingly, the following properties hold true for the eigendecomposition of Laplacian matrices of any order k, with $k = 1, \dots, K-1$:

1. the eigenvectors associated with the nonnull eigenvalues of $\mathbf{B}_k^T \mathbf{B}_k$ are orthogonal to the eigenvectors associated with the nonnull eigenvalues of $\mathbf{B}_{k+1} \mathbf{B}_{k+1}^T$ and viceversa;

2. if $\boldsymbol{v}$ is an eigenvector of $\mathbf{B}_k \mathbf{B}_k^T$ associated with the eigenvalue λ, then $\mathbf{B}_k^T \boldsymbol{v}$ is an eigenvector of $\mathbf{B}_k^T \mathbf{B}_k$, associated with the same eigenvalue;

3. the eigenvectors associated with the nonnull eigenvalues λ of $\boldsymbol{L}_k$ are either the eigenvectors of $\mathbf{B}_k^T \mathbf{B}_k$ or those of $\mathbf{B}_{k+1} \mathbf{B}_{k+1}^T$;

4. the nonnull eigenvalues of $\boldsymbol{L}_k$ are either the eigenvalues of $\mathbf{B}_k^T \mathbf{B}_k$ or those of $\mathbf{B}_{k+1} \mathbf{B}_{k+1}^T$.

All above properties are very easy to prove. Property 1) is straightforward: If $\boldsymbol{B}_k^T \boldsymbol{B}_k \boldsymbol{v} = \lambda \boldsymbol{v}$, then

$$\mathbf{B}_{k+1} \mathbf{B}_{k+1}^T \boldsymbol{v} = \mathbf{B}_{k+1} \mathbf{B}_{k+1}^T \boldsymbol{B}_k^T \boldsymbol{B}_k \boldsymbol{v} / \lambda = \mathbf{0} \tag{6}$$

because of (2). Similarly, for the converse. Property 2) is also straightforward: If $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{B}_k \boldsymbol{B}_k^T$ associated with a nonvanishing eigenvalue λ, then

$$(\boldsymbol{B}_k^T \boldsymbol{B}_k) \boldsymbol{B}_k^T \boldsymbol{v} = \boldsymbol{B}_k^T (\boldsymbol{B}_k \boldsymbol{B}_k^T) \boldsymbol{v} = \lambda \boldsymbol{B}_k^T \boldsymbol{v}. \tag{7}$$

When applied to the case $k = 1$, this implies that, if the graph is composed of a few clusters, the eigenvectors of $\boldsymbol{B}_1^T \boldsymbol{B}_1$ tend to assume very small values on the edges within the clusters, whereas they assume large values, in modulus, on the edges connecting different clusters. Furthermore, the eigenvectors associated with $\boldsymbol{B}_1^T \boldsymbol{B}_1$ are orthogonal to the eigenvectors associated with $\boldsymbol{B}_2 \boldsymbol{B}_2^T$.

3. ANALYSIS OF SIGNALS DEFINED OVER A SIMPLICIAL COMPLEX

Given the vector space of k-chains $C_k(X, \mathbb{R})$, its dual space, denoted as $C^k(X, \mathbb{R})$, is composed of linear functionals, called *k-cochains*, that map every k-chain to a scalar value. Since the vector space of the chains is finite, the corresponding k-cochain vector space is also finite. We define signals over a k-complex as a map from a k-chain to real numbers. In algebraic topology terms, a signal is a k-cochain. Given a K-simplicial complex, we may define signals (cochains) of different order. So, for example, given a graph, we can have a signal defined over the vertices of a graph and over the edges of a graph. Graph Signal Processing (GSP) is essentially concerned with the analysis of 0-cochains (or vertex signals). We want to extend this analysis to higher order cochains. We focus on 1-cochains, or edge (flow) signals, but the generalization to higher order cochains follows the same principles. An example of 1-cochain is reported in Fig. 1 representing the simulation of packet flows over a network, including measurement noise. The signal values are encoded in the gray color of each link. Based on the properties 1. to 4. of the higher order Laplacians derived in Section 2, the space $C^1(X, \mathbb{R})$ of 1-cochains admits the following orthogonal decomposition

$$C^1(X, \mathbb{R}) \doteq \text{img}(\mathbf{B}_1^T) \oplus \ker(\mathbf{L}_1) \oplus \text{img}(\mathbf{B}_2) \qquad (8)$$

where $\ker(\mathbf{L}_1) \doteq \ker(\mathbf{B}_1) \cap \ker(\mathbf{B}_2^T)$ and $\oplus$ denotes the direct sum between spaces. This decomposition is known as *Hodge decomposition* [12] and it is the discrete counterpart of *Helmholtz decomposition* valid for vector fields defined on a continuous space domain. A fundamental property of geometric simplicial complexes of order k is that the dimensions of $\ker(\mathbf{L}_k)$, for $k = 0, \dots, K$ are *topological invariants* of the K-simplicial complex, i.e. topological features that are preserved under homeomorphic transformations of the space. These dimensions are also known as Betti number β_k of order k: β_0 is the number of connected components of the graph, β_1 is the number of holes, β_2 is the number of cavities, and so on [12]. According to the above decomposition, any flow signal can be decomposed in two components, i.e.

$$\boldsymbol{s}^1 = \boldsymbol{s}^1_{\text{irrot}} + \boldsymbol{s}^1_{\text{sol}} \qquad (9)$$

where $\boldsymbol{s}^1_{\text{irrot}}$ represents the irrotational (curl-free) flow, having zero flow-sum along any triangle, and $\boldsymbol{s}^1_{\text{sol}}$ is the solenoidal (divergence-free) component, whose in-flow on each node equals the total out-flow. Equivalently, these two components can be written as $\boldsymbol{s}^1_{\text{irrot}} = \boldsymbol{B}_1^T \boldsymbol{s}^0$ and $\boldsymbol{s}^1_{\text{sol}} = \boldsymbol{B}_2 \boldsymbol{s}^2$, respectively, with $\boldsymbol{s}^0$ being a 0-cochain and $\boldsymbol{s}^2$ a 2-cochain. If $\ker(\mathbf{L}_1)$ is empty, these two components are orthogonal. Interestingly, if we know the 1-cochain $\boldsymbol{s}^1$, we can recover the 0-cochain $\boldsymbol{s}^0$. In fact, exploiting the equality $\mathbf{B}_1 \boldsymbol{s}^1_{\text{sol}} = \mathbf{0}$, we can write

$$\mathbf{L}_0 \boldsymbol{s}^0 = \mathbf{B}_1 \boldsymbol{s}^1. \qquad (10)$$

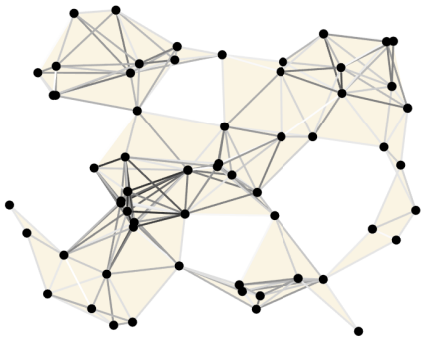

Fig. 1: Observed flow on a simplicial complex.

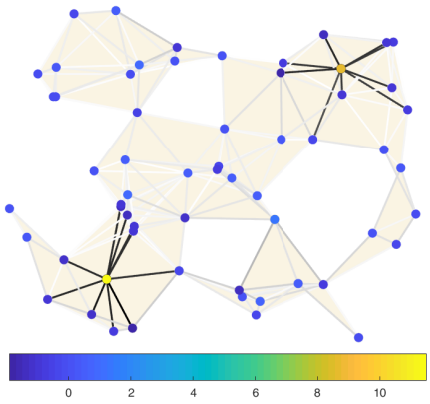

Fig. 2: Reconstruction of irrotational flow.

$\mathbf{L}_0$ is not invertible and, for connected graphs, it has rank $N - 1$ and its kernel is the span of the vector $\mathbf{1}$ of all ones. However, since $\mathbf{B}_1 \boldsymbol{s}^1$ is also orthogonal to $\mathbf{1}$, system (10) admits a nontrivial solution (at least for connected graphs). The representation in (9) suggests also possible ways to extract the solenoidal or irrotational components from noisy observations. Suppose, for example, that given the noisy observation of packet flows, as shown in Fig. 1, we wish to detect whether there are nodes that inject significant amount of data in the network. If this happens, this component would represent the irrotational part. To emphasize this component, we can project the observed flow vector over the space orthogonal to the solenoidal part, i.e. compute the vector

$$\boldsymbol{y}^1 = \left(\boldsymbol{I} - \boldsymbol{U}_{\text{sol}} \boldsymbol{U}_{\text{sol}}^T \right) \boldsymbol{s}^1 \qquad (11)$$

where $\boldsymbol{U}_{\text{sol}}$ is the matrix whose columns are the eigenvectors associated with the nonnull eigenvalues of $\boldsymbol{B}_2 \boldsymbol{B}_2^T$. An example of application to the data reported in Fig. 1 is shown in Fig. 2. We can clearly see how now the only edges with a significant contributions are the ones located around a few nodes,

i.e. the ones injecting traffic in the network whose divergence is encoded by the node color. In many practical cases, it is not possible to measure the data traffic along all the links. Suppose we only observe a subset of traffic data. The question is whether it may be possible to recover the whole traffic from a subset of observations. To answer this question, we may use the theory developed in [13] for signals on graph, and later extended to hypergraphs [14]. Let us denote by $\mathbf{D}$ the diagonal matrix of dimension equal to the number of edges, with a one in the positions where we measure the flow, and zero elsewhere. If the flow signal $\boldsymbol{s}^1$ is bandlimited, i.e. it satisfies the equation $\boldsymbol{F}_1\boldsymbol{s}^1 = \boldsymbol{s}^1$, where $\boldsymbol{F}_1 := \boldsymbol{U}_K\boldsymbol{U}_K^T$ and $\boldsymbol{U}_K$ is an $E \times K$ matrix, with E denoting the number of edges and $K < E$, whose columns are K eigenvectors of $\boldsymbol{L}_1$, then it is possible to recover $\boldsymbol{s}^1$ from its samples if and only if the following condition is satisfied: $\|(\boldsymbol{I} - \boldsymbol{D})\boldsymbol{F}_1\| < 1$, where $\|\boldsymbol{A}\|$ denotes the spectral norm of $\boldsymbol{A}$ [13].

4. INFERENCE OF SIMPLICIAL COMPLEX STRUCTURE FROM OBSERVATIONS

In this section, we propose two algorithms to infer the structure of the simplicial complex from the observed data. This amounts to inferring $\boldsymbol{L}_1$ using the available observations. We assume to observe $\boldsymbol{s}^1$ and that the 0-order Laplacian $\boldsymbol{L}_0$ is known (or it has been estimated using one graph inference method). From $\boldsymbol{L}_0$, choosing an orientation for each edge, we can get $\boldsymbol{B}_1$. Since $\mathbf{L}_1 = \mathbf{B}_1^T\mathbf{B}_1 + \mathbf{B}_2\mathbf{B}_2^T$, the question is then to recover the second term $\mathbf{B}_2\mathbf{B}_2^T$. First of all, we need to check, from the data, if this term is necessary. Since every flow signal can be written as $\boldsymbol{s}^1 = \boldsymbol{B}_1^T\boldsymbol{s}^0 + \boldsymbol{B}_2\boldsymbol{s}^2$, where $\boldsymbol{B}_2\boldsymbol{s}^2$ represents the solenoidal part, to check if $\boldsymbol{B}_2$ is different from zero, we can first extract the solenoidal signal $\boldsymbol{s}^1_{\text{sol}}$ from the observed signal $\boldsymbol{s}^1$. This operation is performed using the projection

$$\boldsymbol{s}^1_{\text{sol}} = \left(\boldsymbol{I} - \boldsymbol{U}_{\text{irrot}}\boldsymbol{U}_{\text{irrot}}^T\right)\boldsymbol{s}^1, \tag{12}$$

where $\boldsymbol{U}_{\text{irrot}}$ is the matrix whose columns are the eigenvectors associated with the nonnull eigenvalues of $\boldsymbol{B}_1^T\boldsymbol{B}_1$. Then we check if the norm of $\boldsymbol{s}^1_{\text{sol}}$ is larger than zero [1]. If the check is positive, start the following algorithm to derive the entries of $\boldsymbol{B}_2$. Let us denote by $\mathbf{T} = \{T_n\}_{n=1}^{|\mathcal{T}|}$ the vector of the possible 2-simplices T_n, $\mathbf{b}_n$ the n-th column of $\mathbf{B}_2$ corresponding to the triangle T_n and $|\mathcal{T}|$ the number of possible triangles. We can formulate our optimization strategy to find the number of 2-simplices from the observed edge signal $\boldsymbol{s}^1$ as

$$\begin{aligned} \min_{\mathbf{T}} \quad & \sum_{n=1}^{|\mathcal{T}|} T_n \boldsymbol{s}^{1\,T}_{\text{sol}}\mathbf{b}_n\mathbf{b}_n^T\boldsymbol{s}^1_{\text{sol}} \\ \text{s.t.} \quad & \sum_{n=1}^{|\mathcal{T}|} T_n \geq t, \quad T_n \in \{0,1\}, \forall n, \end{aligned} \tag{13}$$

[1] In practice, because of noise, we will compare this norm with a nonnegative threshold.

where t is the minimum number of triangles that we may detect. Notice that the triangle signals are representative of local, triangular consistence of the observed edge signals. More specifically, a non-zero triangle implies a curl-free flow and, then, a linear dependence among the edge signals circulating around the triangle. Since problem in (13) is combinatorial, we can solve it efficiently, albeit in approximate sense, by relaxing the binary variables $T_n \in [0, 1]$.

We also propose an alternative algorithm to recover $\boldsymbol{B}_2$, which does not involve the solution of a combinatorial problem. From the observation of $\boldsymbol{s}^0$ and $\boldsymbol{s}^1$, we build the signal $\boldsymbol{z} := \boldsymbol{s}^1 - \boldsymbol{B}_1^T\boldsymbol{s}^0$. Then we construct an augmented matrix $\bar{\boldsymbol{B}}_2$, built assuming that every time the graph has a triplet of edges forming a triangle, we assign a triangle to that triplet. Clearly, not necessarily all these triangles should be filled. The decision about which ones are to discard depends on the observed signal. If the maximum number of triangles is T_{max} and $T_{max} \leq E$, we estimate $\boldsymbol{s}^2$ from $\boldsymbol{z}$ as

$$\hat{\boldsymbol{s}}^2 = \bar{\boldsymbol{B}}_2^{\dagger}\boldsymbol{z} \tag{14}$$

where $\dagger$ denotes pseudo-inverse. We can then compare the entries of $\hat{\boldsymbol{s}}^2$ with a positive threshold: if some coefficients fall below the threshold, we remove the corresponding columns of $\bar{\boldsymbol{B}}_2$ and then the corresponding triangles. The presence of a triangle corresponds to triangular flows with minimum curl. As a consequence the void triangles are those where we expect inconsistent observed flows, such as non regular traffic circulation. The performance of this algorithm are reported in Fig. 3 where we show the percentage of correct detection of the triangles vs. the signal-to-noise ratio (SNR). We can see that, as the SNR increases, the percentage of correct detection tends to be very close to one.

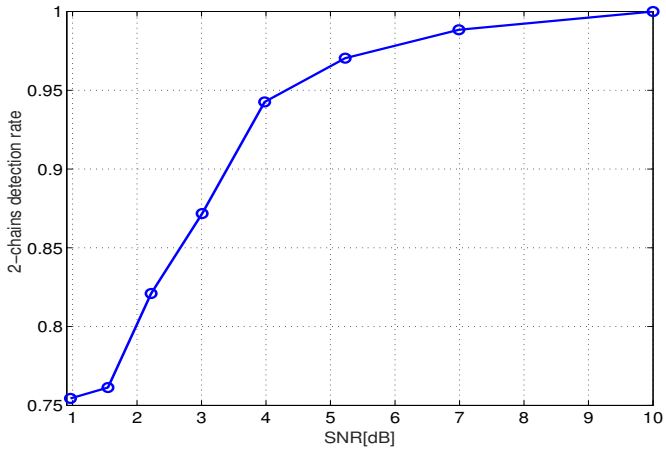

Fig. 3: Triangle correct detection rate vs. SNR.

5. REFERENCES

[1] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] J.R. Munkres, *Topology*, Prentice Hall, 2000.

[3] G. Carlsson, "Topology and data," *Bulletin Amer. Math. Soc.*, vol. 46, no. 2, pp. 255–308, 2009.

[4] A. Muhammad and M. Egerstedt, "Control using higher order Laplacians in network topologies," in *Proc. of 17th Int. Symp. Math. Theory Netw. Syst.*, 2006, pp. 1024–1038.

[5] X. Jiang, L.H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial Hodge theory," *Math. Programm.*, vol. 127, no. 1, pp. 203–244, 2011.

[6] A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification in sensor networks without location information," *IEEE Trans. Autom. Contr.*, vol. 55, no. 8, pp. 1837–1849, 2010.

[7] H. Chintakunta and H. Krim, "Distributed localization of coverage holes using topological persistence," *IEEE Trans. Signal Process.*, vol. 62, no. 10, pp. 2531–2541, 2014.

[8] V. De Silva, R. Ghrist, et al., "Coverage in sensor networks via persistent homology," *Algebraic & Geometric Topology*, vol. 7, no. 1, pp. 339–358, 2007.

[9] S. Emrani, T. Gentimis, and H. Krim, "Persistent homology of delay embeddings and its application to wheeze detection," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 459–463, 2014.

[10] M. Robinson, *Topological signal processing*, Springer, 2016.

[11] L.J. Grady and J.R. Polimeni, *Discrete calculus: Applied analysis on graphs for computational science*, Springer Science & Business Media, 2010.

[12] B. Eckmann, "Harmonische funktionen und randwertaufgaben in einem komplex," *Comment. Math. Helv.*, vol. 17, no. 1, pp. 240–255, 1944.

[13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, 2016.

[14] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 6425–6429.