A DISCRETE SIGNAL PROCESSING FRAMEWORK FOR MEET/JOIN LATTICES WITH APPLICATIONS TO HYPERGRAPHS AND TREES

Markus Püschel

Department of Computer Science
ETH Zurich, Switzerland

ABSTRACT

We introduce a novel discrete signal processing framework, called discrete-lattice SP, for signals indexed by a finite lattice. A lattice is a partially ordered set that supports a meet (or join) operation that returns the greatest element below two given elements. Discrete-lattice SP chooses the meet as shift operation and derives associated notion of (meet-invariant) convolution, Fourier transform, frequency response, and a convolution theorem. Examples of lattices include sets of sets that are closed under intersection and trees. Thus our framework is applicable to certain sparse set functions, signals on sparse hypergraphs, and signals on trees. Another view on discrete-lattice SP is as an SP framework for a certain class of directed graphs. However, it is fundamentally different from the prior graph SP as it is based on more than one basic shift and all shifts are always simultaneously diagonalizable.

Index Terms— Lattice Fourier transform, lattice shift, graph signal processing, hypergraph, algebraic signal processing

1. INTRODUCTION

Signal processing (SP) is a classical data science, founded on a well-developed theory of signals (or data) indexed with time and linear time-invariant systems. The theory provides the key basic concepts of convolution, Fourier transform, sampling, and others. The advent of big data has dramatically increased not only the size but also the variety of data to be processed or analyzed, thus it is of great interest to port basic SP concepts, and hence the large SP tool set that builds on it, to data with other index domains. As one prominent example, many very successful neural nets are built from convolutions and new types of convolutions for graphs thus have found direct applications in this domain [1].

A general platform to derive new linear SP frameworks is provided by the algebraic signal processing theory (ASP) [2, 3]. It provides the basic axioms and theory and identifies the definition of the shift operation as the key concept: the shift captures the structure of the index domain and from its definition a shift-invariant SP framework including all basic SP concepts can be derived. Graph SP as introduced in [4, 5] builds on ASP by stipulating the adjacency matrix as shift (see also [3, Sec.XV-B]). An alternative framework can be built using the Laplacian as shift as done in [6].

Contributions. In this paper we build on classical lattice theory [7, 8] to introduce discrete-lattice SP, a novel linear SP framework for signals indexed by semilattices or lattices. A semilattice is a partially ordered set that supports a meet (or join) operation that returns for every two elements their largest lower bound (or smallest upper bound). In a lattice both meet and join are available. We define the shift through the meet operation and, following ASP, derive associated notions of shift-invariant systems, convolution or filtering, Fourier transform, frequency response, and convolution theorem. In contrast to discrete-time SP or graph SP, filters are not generated by one basic shift but by several, their number depending on the structure of the lattice. Lattice theory guarantees their simultaneous diagonalization and thus a Fourier transform. We illustrate the theory with a small example.

Finally, we will discuss possible application domains that include certain edge- or node-weighted sparse hypergraphs, sparse set functions, and signals on trees. Discrete-lattice SP can also be viewed as an SP framework (different from graph SP) for a certain class of directed graphs whose adjacency matrices have only the eigenvalue zero and are never diagonalizable.

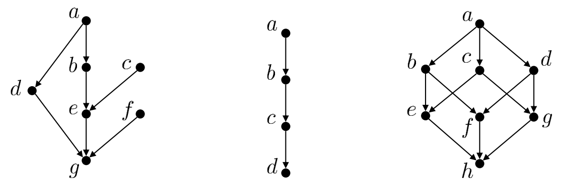
Related work. Just as graph SP builds on concepts from algebraic graph theory, our work builds on the algebraic aspects of lattice theory [7, 8], reinterpreted and expanded for use in signal processing. In particular, [9] allows us to derive the spectral concepts for lattices. We also took inspiration from [10], which provides fast algorithms for the the discrete lattice transform (called Moebius transform) and its inverse (called zeta transform).

There are various ways of defining a form of spectral analysis for hypergraphs, most of which approximate it by a graph to then use spectral graph theory based on adjacency matrix or Laplacian. A good brief overview is given in [11]. A different approach is taken in [12] using simplicial complexes. Our approach is different due to a different notion of shift, and applicable to edge-weighted hypergraphs whose edge set is closed (or almost closed) under intersection. Applications of hypergraphs include [13, 14, 15]. Set functions are equivalent to hypergraphs and have various applications (e.g., [16, 17, 18]) including in machine learning [19]. In [20] we introduced an SP framework for (non-sparse) set functions, which is a particularly well-structured special case of this paper as we will explain in detail later.

A number of SP frameworks based on a shift other than the standard time shift (or translation) have been defined. As said above, two prominent examples are graph SP based on the Laplacian [6] or the adjacency matrix [4] as shift; the shift-invariant filters then become polynomials in the shift. Reference [1] discusses also convolutions on manifolds. A special case of graph SP is discrete-space SP, which assumes a symmetric shift and underlies the discrete cosine and sine transforms [21]. A generalization considers arbitrary nearest-neighbour relations [22]. SP frameworks with more than one, but commuting, basic shifts have been defined in [23, 24, 25] for signals on a hexagonal or quincunx grid, respectively.

2. BACKGROUND: LATTICES

We provide basic background on semilattices and lattices. Good reference books for lattice theory are [7, 8].



(a) Random semilattice (b) Total order lattice (c) Subset lattice

Fig. 1: Examples of semilattices L. The labels are the names of the nodes, i.e., semilattice elements. In this papers we consider signals on L, i.e., that associate values which each node.

We consider finite sets L with a *partial order* $\leq$, also called *posets*. We denote elements of L with lower case letters $a, b, x, \dots$. Formally, a poset satisfies for all $a, b, c \in L$: (a) $a \leq a$; (b) $a \leq b$ and $b \leq a$ implies $a = b$; and (c) $a \leq b$ and $b \leq c$ implies $a \leq c$. We use $a < b$ if $a \leq b$ but $a \neq b$.

We say that b covers a, written as $a \prec b$, if $a < b$ and there is no $x \in L$ with $a < x < b$. So a is a largest element (there could be several) below b.

A meet-semilattice is a poset L that also permits a meet operation $a \wedge b$, which returns the greatest lower bound of a and b. Formally, it satisfies for all $a, b, c \in L$: (a) $a \wedge a = a$; (b) $a \wedge b = b \wedge a$; and (c) $(a \wedge b) \wedge c = a \wedge (b \wedge c)$. The latter two show that $\wedge$ is commutative and associative.

Using the notion of cover, semilattices can be visualized through a directed graph (digraph) (L, E), i.e., the nodes are the elements of L and $(b, a) \in E$ if b covers a. The elements of E are called covering pairs. The graph is typically drawn such that if $a \leq b$, then b is drawn higher than a.

Fig. 1a shows an example of a meet semilattice. The digraph shows, e.g., $e \leq a, g \leq f, b \wedge f = g, a \wedge e = e$, etc. Note that due to the meet it must have a unique minimal element: removing g yields a digraph that is not a semilattice. So not all digraphs are semilattices.

Dually to the meet-semilattice, one can define a *join-semilattice* as a poset with an operation $\vee$ that returns the least upper bound and satisfies analogous properties and necessarily has a unique maximal element. E.g., Fig. 1a is not a join-semilattice since $a \vee d$ does not exist. A poset which is both a meet- and join-semilattice (and with proper interaction between the two operations) is called a lattice. Fig. 1b shows the lattice arising from the total order on a set with four elements.

A very structured example of a lattice is the power set (set of all subsets) of a finite set S with $\cap$ as meet and $\cup$ as join, which yields a directed hypercube as associated digraph. Fig. 1c shows the example for $S = \{\alpha_1, \alpha_2, \alpha_3\}$: here, $a = S$, b, c, d are the two-element subsets, e, f, g the one-element subsets, and $h = \emptyset$. Signals indexed by power sets are called set functions.

In this paper we focus on meet-semilattices, but all theoretical results and concepts can be analogously derived for join-semilattices. For lattices there is thus a choice whether to use the meet or join in our framework.

Finally, we note that in the finite case here, every meet-semilattice can be easily extended to a lattice by adding one auxiliary maximal element that is larger than all $x \in L$. This guarantees the existence of the join.

3. DISCRETE-LATTICE SP

In this section we define discrete-lattice SP[1] (DLSP), a signal processing framework for signals, or data, associated with the elements of a given lattice. A different viewpoint of discrete-lattice SP is that it is an alternative, and novel, form of graph SP for the special digraphs (e.g., those in Fig. 1) associated with lattices. We discuss the differences to graph SP later.

We define DLSP through the shift operation and then derive the associated notions of convolution or filtering, Fourier transform, frequency response, and convolution theorems. A concrete example and the application to hypergraphs is then explained in the next section.

Lattice signals. We consider real signals indexed with the elements of a given meet-semilattice L of size n:

$$\mathbf{s}: \; L \to \mathbb{R}, \quad x \mapsto s_x. \tag{1}$$

We will also write $\mathbf{s} = (s_x)_{x \in L}$, but assume a specific order to obtain a coordinate vector only if stated. The set of signals is an n-dimensional vector space.

Shifts. The central concept in the derivation of a signal processing framework is the definition of the shift operation, as all other concepts can be derived from it [2]. Our construction uses the semilattice operation $\wedge$ as shift. Formally, for every $a \in L$ we define the

$$\text{shift by } a: \quad (s_{x \wedge a})_{x \in L},$$

which captures the semilattice structure. Obviously, this shift is a linear mapping on the signal space since shifting $\alpha\mathbf{s} + \beta\mathbf{s}'$ ($\alpha, \beta \in \mathbb{R}$) by a yields $\alpha(s_{x \wedge a})_{x \in L} + \beta(s'_{x \wedge a})_{x \in L}$.

In discrete-time SP a signal can be shifted by any integer value, but each such shift can be expressed as a repeated shift by 1, i.e., the shift by 1 *generates* all others. The question is which shifts by a in our lattice are needed to generate all others. Intuitively, it should be those that move the elements of the lattice the least, i.e., given by the largest a. Lattice theory explains that the generators of the meet-semilattice are precisely all meet-irreducible elements [8], i.e., all $a \in L$ that cannot be written as $a = b \wedge c$ with $b, c \neq a$.

In Fig. 1a the basic shifts that generate all others are thus given by a, c, b, d, f. In Fig. 1b these are all elements, and in Fig. 1c they are a, b, c, d.

Convolution. Since we can now shift by any $a \in L$ we can linearly extend to a notion of convolution. Namely, if $\mathbf{h} = (h_q)_{q \in L}$ is a filter we obtain

$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q} \right)_{x \in L}.$$

Shift-invariance. Since the meet operation is commutative, all shifts commute with each other and thus also with all filters, in other words, our SP framework is shift-invariant.

Pure frequencies and frequency response. To derive the associate Fourier transform we first determine the pure frequencies, i.e., the signals that are simultaneous eigenfunctions for all shifts and thus for all filters. Note that their existence is possible since the shifts commute.

Lattice theory provides the frequencies through the Zeta transform [9]. Translated to our SP setting, there is a pure frequency $\mathbf{f}^y$

[1] More correct would be discrete-semilattice SP, but we opted for simplicity.

for every $y \in L$, defined as

$$\mathbf{f}^y = (\iota_{y\le x})_{x\in L}. \qquad (2)$$

Here, $\iota_{y\le x} = \iota_{y\le x}(x,y)$ is the characteristic function of $y \le x$: $\iota_{y\le x} = 1$ if $y \le x$ holds, and $= 0$ else. To show the assertion, we shift $\mathbf{f}^y$ by some $q \in L$ and get

$$(\iota_{y\le x\wedge q})_{x\in L}.$$

If $y \le q$, then $y \le x \wedge q \Leftrightarrow y \le x$ and thus the result is $\mathbf{f}^y$. This means the frequency response of a shift by such a q is 1. If $y \not\le q$, then $y \le x \wedge q$ never holds and the result is the zero vector. This means the frequency response of a shift by such a q is 0.

By linear extension, the frequency response of a filter $\mathbf{h} = (h_q)_{q\in L}$ at frequency y is computed as

$$\overline{h}_y = \sum_{q\in L,\ y\le q} h_q. \qquad (3)$$

Note that the frequencies are indexed by $y \in L$, this suggests that they inherit the partial order of L that then stipulates which frequencies are high and which low. In our example below we provide more evidence that this interpretation is meaningful.

Fourier transform. We denote the Fourier coefficients of a signal $\mathbf{s}$ as $\widehat{\mathbf{s}} = (\widehat{s}_y)_{y\in L}$. Equation (2) shows that the inverse Fourier transform is given by

$$s_x = \sum_{y\in L} (\iota_{y\le x})\widehat{s}_y = \sum_{y\le x} \widehat{s}_y.$$

This equation is inverted using the classical Moebius inversion formula [9] and yields the associated Fourier transform that we call *discrete lattice transform (*DLT *or* DLT_L*)*:

$$\widehat{s}_y = \sum_{x\le y} \mu(x,y) s_x. \qquad (4)$$

Here, μ is the Moebius function, defined recursively as

$$\mu(x,x) = 1, \qquad \text{for } x \in L,$$
$$\mu(x,y) = -\sum_{x\le z<y} \mu(x,z), \qquad x \ne y.$$

Convolution theorem. The preceding results yield the following convolution theorem:

$$\widehat{\mathbf{h} * \mathbf{s}} = \overline{\mathbf{h}} \odot \widehat{\mathbf{s}}.$$

Note that frequency response and Fourier transform are computed differently. The frequency response can be inverted using an analogous Moebius inversion formula [7, p. 304].

Fast algorithms. Fourier transform and frequency response and their inverses can be computed in $O(\mu n)$ many operation where μ is the number of meet-irreducible elements of L [10]. In some cases this can be further to $O(\eta)$, where η is the number of covering pairs.

Comparison with discrete-time SP. We compare the derived main concepts to their counterparts in finite discrete-time SP. By finite we mean signals with finite support, assumed to be periodically extended. The comparison is summarized in Table 1. We note that finite discrete-time SP is not a special case of DLSP. The former assumes that the signals resides on a circle, which is not a lattice or semilattice since it lacks minimum and maximum.

Table 1: Comparison of DLSP and finite DTSP. For the latter, the signals are assumed periodically extended, i.e., the indices are all considered mod n as usual. For simplicity we denote the index range with $[n] = (0,\dots,n-1)$ and set $\omega_n = \exp(-2\pi j/n)$.

Concept	DLSP	DTSP
Signal	$(s_x)_{x\in L}$	$(s_k)_{k\in[n]}$
Filter	$(h_q)_{q\in L}$	$(h_m)_{m\in[n]}$
Basic shifts	$(s_{x\wedge b})_{x\in L}$, b meet irred.	$(s_{k-1})_{k\in[n]}$
Convolution	$\sum_{q\in L} h_q s_{x\wedge q}$	$\sum_{0\le m<n} h_m s_{k-m}$
Pure frequency	$(\iota_{y\le x})_{x\in L},\ y \in L$	$\frac{1}{n}(\omega_n^{-k\ell})_{k\in[n]},\ \ell\in[n]$
Fourier transform	$\widehat{s}_y = \sum_{x\le y} \mu(x,y)s_x$	$\widehat{s}_\ell = \sum_{k\in[n]} \omega_n^{k\ell} s_k$

4. EXAMPLE

As an example we consider the lattice in Fig. 1a with five basic shifts a, b, c, d, f. To allow for a representation of shifts, filters, and Fourier transform as matrices, we order the elements of L as suggested by the names from a to g. Signals are viewed as column vectors. Since the chosen order is a topological sort of the lattice (i.e., compatible with its partial order), all matrices will become upper triangular.

Every shift by $x \in L$ can be represented by a matrix S_x; we consider $x = c$ as example. To obtain S_c, we let c operate on the canonical basis vectors $(\iota_{x=k})_{x\in L}$ for $k \in L$. For example, for $k = e$, the shift by c maps

$$(\iota_{x=e})_{x\in L} \mapsto (\iota_{x\wedge c=e})_{x\in L}.$$

$x\wedge c = e$ holds for $x = a, b, e$, thus the result is $(1,1,0,0,1,0,0)^T$, which becomes the eth column of S_e. The overall result is shown in Fig. 2a. Note that S_e and the other shift matrices are sparse, but filter matrices are (triangular) dense in general.

Next, we compute the pure frequencies using (2) which are the columns of DLT_L^{-1}. The result is shown in Fig. 2b. We observe the constant all-one signal as lowest frequency, just as in discrete-time SP. It is the pure frequency associated with the minimal element in L.

Finally, the DLT_L in matrix form is computed using (4) or by inverting Fig. 2b and shown in Fig. 2c. We note that in general, entries other than $1, 0, -1$ do occur. The spectrum of a signal $\mathbf{s}$ now can be computed as the matrix vector product

$$\widehat{\mathbf{s}} = \mathrm{DLT}_L\,\mathbf{s}.$$

All shift and filter matrices are diagonalized by the DLT. Indeed

$$\mathrm{DLT}_L\, S_e\, \mathrm{DLT}_L^{-1} = \mathrm{diag}(0,0,1,0,1,0,1,0).$$

As expected, all eigenvalues (i.e., the frequency response of the shift by e) are zero or one.

In discrete-time SP, a basic low pass filters adds a signal to its shifted version: $(s_n + s_{n-1})_n$. Translated to DLSP, we add all basic-shifted versions, i.e., the low pass filter becomes $\mathbf{h} = (1,1,1,1,0,1,0)^T$. The frequency response, computed with (3), thus becomes $\overline{\mathbf{h}} = (1,2,1,2,3,1,5)$.[2] Above we suggested that the frequencies inherit the partial order of L, i.e., we can associate its values with the elements of the lattice. The visualization of the

[2]A detail here is that the identity mapping is a filter if and only if L contains a maximal element m (shifting by m is then the identity). If this is not available we can only add the basic-shifted versions and not the signal itself.

$$\begin{bmatrix} 0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1 \end{bmatrix}$$

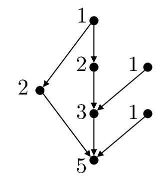
(a) Shift matrix S_e

$$\begin{bmatrix} 1&1&0&1&1&0&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&1&0&1\\ 0&0&0&1&0&0&1\\ 0&0&0&0&1&0&1\\ 0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1 \end{bmatrix}$$

(b) DLT_L^{-1} containing the pure frequencies as columns from high (left) to low (right)

$$\begin{bmatrix} 1&-1&0&-1&0&0&1\\ 0&1&0&0&-1&0&0\\ 0&0&1&0&-1&0&0\\ 0&0&0&1&0&0&-1\\ 0&0&0&0&1&0&-1\\ 0&0&0&0&0&1&-1\\ 0&0&0&0&0&0&1 \end{bmatrix}$$

(c) DLT_L in matrix form



(d) Frequency response of a basic low pass filter

Fig. 2: Basic DLSP concepts instantiated for the meet-semilattice in Fig. 1a.

response in Fig. 2d shows that indeed lower frequencies are amplified. In particular, the lowest values are associated with the maximal elements. The definition in (3) shows that this holds in general.

Discussion. We already pointed out that DLSP does not include finite DTSP as special case since a circle is not a semilattice.

DLSP provides an SP framework for a special class of digraphs, but is fundamentally different from the graph SP in [4, 5] (see also [3, Sec. XV]). In graph SP only one basic shift (the adjacency matrix) is available and diagonalization is not guaranteed. In fact, for all digraphs (such as Fig. 1) associated with semilattices, the associated adjacency matrix has the characteristic polynomial x^n ($n = |L|$) and is thus not diagonalizable. To see this, order L topologically, which makes the adjacency matrix upper triangular with zeros on the diagonal. In contrast our framework, based on lattice theory, guarantees simultaneous diagonalization of all shifts and hence all filters and hence the existence of a Fourier transform. One could argue that the notion of filtering also better captures the details of the lattice structure.

A special case of DLSP is the discrete-set SP that we introduced in [20]. It arises from setting $L = 2^U$, the power set of a finite set $U = \{\alpha_1, \dots, \alpha_t\}$, with $\wedge = \cap$. Thus $n = 2^t$. The meet-irreducible elements are the maximal subsets $U \setminus \{\alpha_i\}$, which yields the t basic shifts $(s_V)_{V\subset U} \mapsto (s_{V\cap U\setminus\{\alpha_i\}})_{V\subset U} = (s_{V\setminus\{\alpha_i\}})_{V\subseteq U}$. We called these natural delays in [20] and derived all associated SP concepts. In essence, [20] considers the special case of subset lattices such as the one in Fig. 1c.

Finally, the basic shifts in DLSP are not invertible but this can happen also in graph SP and even in the space SP underlying the discrete cosine transform [21].

5. POSSIBLE APPLICATIONS

We discuss possible application scenarios in which signals are naturally indexed by lattices.

Signals on hypergraphs. Given a finite set of nodes V, an *edge-weighted* hypergraph [26, 27] given by $H = (V, E, \mathbf{w})$, where $E \subseteq 2^V$ is the set of hyperedges (2^V is the power set or set of all subsets of V) and $\mathbf{w} : E \mapsto \mathbb{R}$ are their weights. Typically, hypergraphs are very sparse, i.e., E is only a small subset of 2^V. We can view $\mathbf{w}$ as lattice signal with $\wedge = \cap$ if we extend E to its meet-closure E', i.e., E' contains all subsets of V that can be expressed as intersections, or, equivalently, E' is the smallest meet-subsemilattice of 2^V that contains E. The weights of the added hyperedges in $E' \setminus E$ are set to 0. This construction makes particular sense if E is already almost closed under the meet.

As a simple, somewhat artificial example consider a book written (for example) in English. We set V as the alphabet and add an edge $A \subseteq 2^V$ to E of there is a word in the book whose character set is exactly A. We set the weight $w_A = \mathbf{w}(A)$ to the number of such words making $\mathbf{w}$ our lattice signal. In many cases $A, B \in E$ will imply $A \cap B \in E$. If not, $A \cap B$ is added to E with weight 0. Now E is meet-closed (i.e., intersection-closed) and our framework can be applied. The basic shifts correspond to computing the intersection with the maximal occurring sets $M \in E$: $w_A \mapsto w_{A\cap M}$.

Finally, we note that in a hypergraph the roles of nodes and edges can be exchanged to convert between weights on edges and weights on nodes. As an example, consider now a *node-weighted* hypergraph $H = (V, E, \mathbf{w})$, where $\mathbf{w} : V \mapsto \mathbb{R}$, i.e., $\mathbf{w}$ is a signal indexed by the nodes of H. The dual hypergraph is given by $H' = (E, V', \mathbf{w'})$, where $V' = \{F_v \subseteq E, v \in F \mid v \in V\} \subseteq 2^E$ and $\mathbf{w'}(F_v) = \mathbf{w}(v)$. Now the prior discussion applies to H'.

Set functions. The prior discussion shows that edge- or node-weighted hypergraphs are equivalent to set functions, i.e., function on the power set of a finite set. Thus our framework is applicable to sparse set functions that are closed under meet. Set functions have numerous applications as discussed in the introduction.

Signals on trees. Another example of meet-semilattices are trees. Specifically, we view a tree upside down as digraph, as in Fig. 1. The root is the smallest element and every node is covered by its children. The meet-irreducible elements are then the leaves of the tree. Our work thus offers a DSP framework for signals on trees, different from graph signal processing as mentioned before. An application example of tree signals is given in [28].

6. CONCLUSIONS

We introduced discrete-lattice SP for signals indexed by semilattices that support a meet operation. The derivation for an available join was not shown but is analogous and will yield a dual result. The first key component of discrete-lattice SP is a suitable notion of shift-invariant convolution, where the shifts correspond to the meet operation on the signal indices. The second is the associated Fourier transform that diagonalizes all shifts and convolutions. While lattices can be visualized as graphs, discrete-lattice SP is not comparable to graph SP in which filters are generated from one basic shift, whereas our framework used several, depending on the lattice structure. Interestingly, simultaneous diagonalization, and thus a Fourier transform, is still guaranteed by lattice theory, whereas the adjacency matrices of lattice graphs are never diagonalizable.

Our contribution is theoretical but we did point out possible application domains including hypergraphs, set functions, and signals on trees. More work is needed to explore such applications, expand the theory, and to also develop a better intuition for what lattice convolutions and lattice Fourier transforms achieve in practice.

7. REFERENCES

[1] M. M. Bronstein, J. Bruna, Y. LeCun, Szlam A., and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, pp. 18–42, 2017.

[2] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. on Signal Processing*, vol. 56, no. 8, pp. 3572–3585, 2008.

[3] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory," *CoRR*, vol. abs/cs/0612077, 2006.

[4] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

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

[7] R. P. Stanley, *Enumerative Combinatorics: Volume 1*, Cambridge University Press, 2nd edition, 2011.

[8] G. Grätzer, *Lattice Theory: Foundation*, Birkhäuser, 2011.

[9] G.-C. Rota, "On the foundations of combinatorial theory. I. theory of Möbius functions," *Z. Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 2, no. 4, pp. 340–368, 1964.

[10] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, J. Nederlof, and P. Parviainen, "Fast zeta transforms for lattices with few irreducibles," *ACM Trans. on Algorithms*, vol. 12, no. 1, pp. 4:1–4:19, 2015.

[11] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. International Conference on Machine Learning (ICML)*, 2006, pp. 17–24.

[12] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[13] B. Oselio, A. Kulesza, and A. O. Hero, "Multi-layer graph analysis for dynamic social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, 2014.

[14] B. M. Jones, M. Campbell, and L. Tong, "Maximum likelihood fusion of stochastic maps," *IEEE Trans. on Signal Processing*, vol. 62, no. 8, pp. 2090–2099, 2014.

[15] W. Huang and A. Ribeiro, "Persistent homology lower bounds on high-order network distances," *IEEE Trans. on Signal Processing*, vol. 65, no. 2, pp. 319–334, 2017.

[16] H. Zhu, N. Prasad, and S. Rangarajan, "Precoder design for physical layer multicasting," *IEEE Trans. on Signal Processing*, vol. 60, no. 11, pp. 5932–5947, 2012.

[17] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE J. Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 809–822, 2016.

[18] M. Coutino, S. P. Chepuri, and G. Leus, "Submodular sparse sensing for gaussian detection with correlated observations," *IEEE Trans. on Signal Processing*, vol. 66, no. 15, pp. 4025–4039, 2018.

[19] A. Krause and D. Golovin, *Tractability: Practical Approaches to Hard Problems*, chapter Submodular function maximization, pp. 71–104, Cambridge University Press, 2014.

[20] M. Püschel, "A discrete signal processing framework for set functions," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[21] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Trans. on Signal Processing*, vol. 56, no. 8, pp. 3586–3599, 2008.

[22] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Algebraic signal processing theory: 1-D nearest-neighbor models," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2247–2259, 2012.

[23] M. Püschel and M. Rötteler, "Algebraic signal processing theory: 2-D hexagonal spatial lattice," *IEEE Trans. on Image Processing*, vol. 16, no. 6, pp. 1506–1521, 2007.

[24] M. Püschel and M. Rötteler, "Fourier transform for the directed quincunx lattice," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, vol. 4, pp. 401–404.

[25] M. Püschel and M. Rötteler, "Fourier transform for the spatial quincunx lattice," in *Proc. International Conference on Image Processing (ICIP)*, 2005, vol. 2, pp. 494–497.

[26] A. Bretto, *Hypergraph Theory: An Introduction*, Springer, 2013.

[27] D. Zhou, J. Huang, and B Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, vol. 19, pp. 1601–1608.

[28] G. Mishne, R. Talmon, I. Cohen, R. R. Coifman, and Y. Kluger, "Data-driven tree transforms and metrics," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 451–466, 2018.