DISCRETE SIGNAL PROCESSING ON GRAPHS: GRAPH FILTERS

Aliaksei Sandryhaila and José M. F. Moura

Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

We propose a novel discrete signal processing framework for structured datasets that arise from social, economic, biological, and physical networks. Our framework extends traditional discrete signal processing theory to datasets with complex structure that can be represented by graphs, so that data elements are indexed by graph nodes and relations between elements are represented by weighted graph edges. We interpret such datasets as signals on graphs, introduce the concept of graph filters for processing such signals, and discuss important properties of graph filters, including linearity, shift-invariance, and invertibility. We then demonstrate the application of graph filters to data classification by demonstrating that a classifier can be interpreted as an adaptive graph filter. Our experiments demonstrate that the proposed approach achieves high classification accuracy.

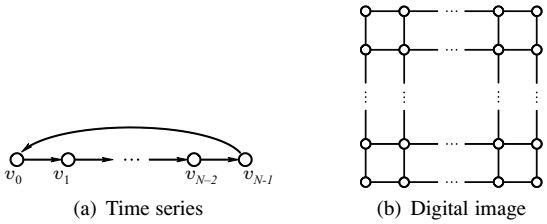
Index Terms— Graph signal processing, graph signal, graph filter, structured data, data classification, label propagation.

1. INTRODUCTION

Recent years have witnessed an enormous growth of interest in the analysis of large-scale datasets emerging in various fields and applications, such as social and economic networks, internet and world wide web, image and video databases, sensor and transportation networks. A common feature of these datasets is their complex relational structure represented, for example, by similarities or dependencies between data elements. A usual way to represent this structure is to use graphs, so that data elements are indexed by graph nodes, and the strength of relations between elements is represented via corresponding weighted graph edges.

The graph representation of structured data has been exploited in numerous works. In social and economic networks, graph properties, such as degree distributions, node centrality and betweenness, and clustering, have been used to infer the community structure and interaction [1, 2]. Graphical models [3, 4] study inference and learning from structured datasets by viewing data elements as random variables and reflecting their probabilistic dependencies between each other with graph edges. Spectral graph theory has been applied for data learning [5, 6]. These methods, however, analyze the representation graph, not the actual data. More recently, the Laplacian matrix and its eigenvectors have been used for the representation and spectral analysis of data [7, 8]. This approach is significantly more similar to existing signal processing techniques, and to our work in particular. Its major limitation, however, is the restriction to indexing graphs that are undirected and have real, non-negative edge weights.

In this paper, we propose a framework, which we call *discrete signal processing on graphs* (DSP_G), for the representation, analysis, and processing of data indexed by arbitrary graphs. Our framework extends the traditional discrete signal processing (DSP) theory that addresses signals with linear structure, such as time (speech, radar, econometric series) and space signals (images), to datasets with complex structure. We define the fundamental DSP concepts of signals and filters on graphs[1], and discuss important properties of graph filters. As a potential application of graph filters we consider data classification. In particular, we demonstrate that data labels can be viewed as a graph signal, and a data classifier can be represented by a graph filter that processes known labels (input signal) to predict unknown labels (output signal). We demonstrate that a properly designed graph filter achieves high classification accuracy even when a relatively small fraction of data labels are known initially.



(a) Time series (b) Digital image

Fig. 1. Graph representations for standard DSP signals.

2. GRAPH SIGNALS

Consider a dataset, for which we know how its elements relate to each other. A simple example is a set of images, where the information about them, such as content, represents the data, and the similarity between images represents the relation. Another example is a collection of documents, where their topics represent the data, and references (citations, hyperlinks) between documents represent the relations. Despite the different nature of each dataset, if we represent it in a numeric form, we can view it as a set of vectors

$$\mathcal{S} = \left\{ \mathbf{s} : \mathbf{s} = (s_0, \ldots, s_{N-1})^T, s_n \in \mathbb{C} \right\}, \tag{1}$$

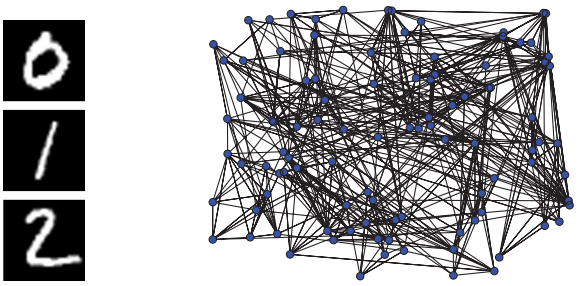
and represent the relation between coefficients s_n of $\mathbf{s}$ with a graph $G = (\mathcal{V}, \mathbf{A})$. Here, $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ is a set of N nodes, and $\mathbf{A}$ is a $N \times N$ weighted adjacency matrix. Each coefficient s_n is indexed by node v_n, and the weight $\mathbf{A}_{n,m}$ of the directed edge from v_m to v_n reflects the degree of relation of s_n to s_m. Note that edge weights $\mathbf{A}_{n,m}$ can take arbitrary real or complex values (for example, if data elements are negatively correlated). We denote the set of indices of nodes connected to v_n as $\mathcal{N}_n = \{m \mid \mathbf{A}_{n,m} \neq 0\}$. We call a signal $\mathbf{s}$ indexed by a graph G a *graph signal*.

Since, in general, signals can be complex-valued, and they can be added together and scaled by constant coefficients, they form a vector space. If we do not make additional restrictions, the set $\mathcal{S}$ of graph signals is the N-dimensional complex vector space $\mathcal{S} = \mathbb{C}^N$.

Let us illustrate graph signals with several examples. A finite periodic discrete time series (a standard signal in finite DSP), can be represented by the directed cyclic graph in Fig. 1(a) [10, 11]. The causality of a time series is represented by the graph edges that are

This work was supported in part by AFOSR grant FA95501210087.

[1]Other fundamental concepts of DSP_G, including the spectrum, Fourier transform, and frequency response, are discussed in [9].



(a) (b)

Fig. 2. Graph representation for images of handwritten digits: a) example images; b) similarity graph of 100 images.

all directed and have the same weight; and the periodicity is represented by the edge from v_{N-1} to v_0. A general digital image can be represented by the two-dimensional rectangular lattice in Fig. 1(b). Pixels correspond to graph nodes and are related to the four adjacent pixels. This relation is symmetric, hence all edges are undirected and have the same weight, with possible exceptions of boundary nodes, depending on boundary conditions [10, 12]. The graph in Fig. 2(b) represents images of handwritten digits, such as the ones shown in Fig. 2(a) [13]. Each node corresponds to an image and connects to several most similar images, where similarity of two images is measured as the ℓ_2-norm of their difference. We discuss this dataset in more detail in Section 5. Note that, unlike the graphs for discrete time series and images, this graph has no immediately apparent structure. Nevertheless, as we demonstrate in Section 5, it is just as useful as the previous graphs.

3. GRAPH FILTERS

Here, we introduce the concept of graph filters. Similarly to the traditional DSP, graph filters are systems that take a graph signal as an input and produce another signal indexed by the same graph as the output.

Graph shift. In DSP, the basic building block for filters is a time delay, or time shift, denoted by z^{-1} [14]. It represents the simplest non-trivial system that delays the input signal $\mathbf{s}$ by one sample. Using the graph representation of finite periodic time series in Fig. 1(a), for which the adjacency matrix $\mathbf{A}$ is the $N \times N$ circulant matrix with weights [10, 11]

$$\mathbf{A}_{n,m} = \begin{cases} 1, & \text{if } n-m=1 \mod N \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

we can write the output of the time shift as

$$\tilde{\mathbf{s}} = \mathbf{A}\mathbf{s}. \tag{3}$$

In DSP_G, we extend the notion of the shift (3) to graph signals indexed by a graph $G = (\mathcal{V}, \mathbf{A})$. We call the operation (3) the *graph shift*. It is realized by replacing the coefficient s_n at node v_n with the weighted linear combination of the coefficients at its neighbor nodes:

$$\tilde{s}_n = \sum_{m=0}^{N-1} \mathbf{A}_{n,m} s_m = \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m} s_m. \tag{4}$$

Graph filters. The representation (3) of a graph shift as a matrix-vector multiplication suggests that, in general, any matrix $\mathbf{H} \in \mathbb{C}^{N \times N}$ that for input signal $\mathbf{s} \in \mathcal{S}$ outputs another graph signal $\tilde{\mathbf{s}} = \mathbf{H}\mathbf{s}$ corresponds to a *graph filter*. This implies that graph filters are linear, since for any $\mathbf{H}, \mathbf{G} \in \mathbb{C}^{N \times N}$ the output of the linear combination of these systems is the linear combination of their outputs:

$$(\alpha\mathbf{H} + \beta\mathbf{G})\mathbf{s} = \alpha\mathbf{H}\mathbf{s} + \beta\mathbf{G}\mathbf{s}.$$

However, in this paper we would like to focus on *shift-invariant* graph filters, for which applying the graph shift to the output is equivalent to applying the graph shift to the input prior to filtering:

$$\mathbf{A}\big(\mathbf{H}\mathbf{s}\big) = \mathbf{H}\big(\mathbf{A}\mathbf{s}\big). \tag{5}$$

The next theorem establishes the structure of linear, shift-invariant graph filters $\mathbf{H}$ as *polynomials* in the shift $\mathbf{A}$.

Theorem 1 Let $\mathbf{A}$ be the graph adjacency matrix and assume that its characteristic and minimal polynomials are equal: $p_\mathbf{A}(x) = m_\mathbf{A}(x)$. Then, a graph filter $\mathbf{H}$ is linear and shift invariant if and only if (iff) $\mathbf{H}$ is a *polynomial* in the graph shift $\mathbf{A}$, i.e., iff there exists a polynomial

$$h(x) = h_0 + h_1 x + \ldots + h_L x^L \tag{6}$$

with possibly complex coefficients $h_\ell \in \mathbb{C}$, such that:

$$\mathbf{H} = h(\mathbf{A}) = h_0\,\mathbf{I} + h_1\mathbf{A} + \ldots + h_L\mathbf{A}^L. \tag{7}$$

Proof: Since the shift-invariance condition (5) holds for all graph signals $\mathbf{s} \in \mathcal{S} = \mathbb{C}^N$, the matrices $\mathbf{A}$ and $\mathbf{H}$ commute: $\mathbf{A}\mathbf{H} = \mathbf{H}\mathbf{A}$. As $p_\mathbf{A}(x) = m_\mathbf{A}(x)$, all eigenvalues of $\mathbf{A}$ have exactly one eigenvector associated with them [15]. Then, the graph matrix $\mathbf{H}$ commutes with the shift $\mathbf{A}$ iff it is a polynomial in $\mathbf{A}$ (see Proposition 12.4.1 in [15]). □

We call the coefficients h_ℓ of the polynomial $h(x)$ in (6) the graph filter *taps*. Obviously, any graph filter is precisely represented by its taps.

Properties of graph filters. The assumption on $p_\mathbf{A}(x)$ and $m_\mathbf{A}(x)$ in Theorem 1 does not hold for all adjacency matrices $\mathbf{A}$. Nevertheless, we can extend the result in Theorem 1 to all matrices using the concept of equivalent graph filters, as defined next.

Definition 1 Given any shift matrices $\mathbf{A}$ and $\widetilde{\mathbf{A}}$, filters $h(\mathbf{A})$ and $g(\widetilde{\mathbf{A}})$ are called *equivalent* if for all input signals $\mathbf{s} \in \mathcal{S}$ they produce equal outputs: $h(\mathbf{A})\mathbf{s} = g(\widetilde{\mathbf{A}})\mathbf{s}$.

Hence, given a signal $\mathbf{s}$ indexed by an arbitrary $G = (\mathcal{V}, \mathbf{A})$ with $p_\mathbf{A}(x) \neq m_\mathbf{A}(x)$, we can view it as being indexed by another graph $\widetilde{G} = (\mathcal{V}, \widetilde{\mathbf{A}})$ with the same set of nodes $\mathcal{V}$ but potentially different edges and edge weights, for which $p_{\widetilde{\mathbf{A}}}(x) = m_{\widetilde{\mathbf{A}}}(x)$ holds true. Then graph filters on G can be expressed as equivalent filters on $\widetilde{G}$, as described by the following theorem (the proof is available in [9]).

Theorem 2 For any matrix $\mathbf{A}$ there exists a matrix $\widetilde{\mathbf{A}}$ and polynomial $r(x)$, such that $\mathbf{A} = r(\widetilde{\mathbf{A}})$ and $p_{\widetilde{\mathbf{A}}}(x) = m_{\widetilde{\mathbf{A}}}(x)$.

As a consequence of Theorem 2, any filter on the graph $G = (\mathcal{V}, \mathbf{A})$ is equivalent to a filter on the graph $\widetilde{G} = (\mathcal{V}, \widetilde{\mathbf{A}})$, since $h(\mathbf{A}) = h(r(\widetilde{\mathbf{A}})) = (h \circ r)(\widetilde{\mathbf{A}})$, where $h \circ r$ denotes the polynomial composition of h and r. Thus, the condition $p_\mathbf{A}(x) = m_\mathbf{A}(x)$ in Theorem 1 can be assumed to hold for any graph $G = (\mathcal{V}, \mathbf{A})$. Otherwise, by Theorem 2, we can replace the graph by another $\widetilde{G} = (\mathcal{V}, \widetilde{\mathbf{A}})$ for which the condition holds, and assign $\widetilde{\mathbf{A}}$ to $\mathbf{A}$.

Furthermore, as the next theorem demonstrates, the number of taps required to represent any graph filter is limited.

Theorem 3 Any graph filter (7) has a unique equivalent filter on the same graph with at most $\deg m_{\mathbf{A}}(x) = N_{\mathbf{A}}$ taps.

Proof: Consider the polynomials $h(x)$ in (6). By polynomial division, there exist unique polynomials $q(x)$ and $r(x)$:

$$h(x) = q(x)m_{\mathbf{A}}(x) + r(x), \tag{8}$$

where $\deg r(x) < N_{\mathbf{A}}$. Hence, we can express (7) as

$$h(\mathbf{A}) = q(\mathbf{A})m_{\mathbf{A}}(\mathbf{A}) + r(\mathbf{A}) = q(\mathbf{A})\,\mathbf{0}_N + r(\mathbf{A}) = r(\mathbf{A}).$$

Thus, $h(\mathbf{A}) = r(\mathbf{A})$ and $\deg r(x) < \deg m_{\mathbf{A}}(x) = N_{\mathbf{A}}$. □

As a consequence of Theorem 3, all graph filters (7) on the indexing graph $G = (\mathcal{V}, \mathbf{A})$ form a set

$$\mathcal{F} = \left\{ \mathbf{H} : \;\; \mathbf{H} = \sum_{\ell=0}^{N_{\mathbf{A}}-1} h_\ell \mathbf{A}^\ell \,\middle|\, h_\ell \in \mathbb{C} \right\}. \tag{9}$$

Due to the linearity of graph filters, this set is a vector space. Moreover, the shift-invariance of filters in $\mathcal{F}$ implies that multiplication of graph filters produces filters from $\mathcal{F}$. Furthermore, if a graph filter $h(\mathbf{A})$ is invertible, its inverse $h(\mathbf{A})^{-1}$ is also a graph filter from $\mathcal{F}$ (the proof is available in [9]):

Theorem 4 A graph filter $\mathbf{H} = h(\mathbf{A}) \in \mathcal{F}$ is invertible iff polynomial $h(x)$ satisfies $h(\lambda_m) \neq 0$ for all distinct eigenvalues λ_m, of $\mathbf{A}$. Then, there is a unique polynomial $g(x)$ of degree $\deg g(x) < N_{\mathbf{A}}$ that satisfies

$$h(\mathbf{A})^{-1} = g(\mathbf{A}) \in \mathcal{F}. \tag{10}$$

Theorem 4 implies that instead of inverting the $N \times N$ matrix $h(\mathbf{A})$ directly, we only need to construct a polynomial $g(x)$ specified by at most $N_{\mathbf{A}}$ taps. Overall, the above properties of graph filters imply that $\mathcal{F}$ has the structure of an algebra [10].

Example 1 Observe that the proposed definition of graph filters in DSP_G is consistent with the traditional DSP theory. As we discussed in Section 2, finite discrete periodic time series are represented by the directed graph in Fig. 1(a). Its adjacency matrix is the $N \times N$ circulant matrix (2). Hence, for any graph filter $h(\mathbf{A}) = h(\mathbf{C}_N) = \sum_{\ell=0}^{N-1} h_\ell\, \mathbf{C}_N^\ell$, the coefficients of its output $\widehat{\mathbf{s}} = h(\mathbf{C}_N)\mathbf{s}$ are calculated as

$$\begin{aligned} \widehat{s}_n &= h_n s_0 + \ldots + h_0 s_n + h_{N-1} s_{n+1} + \ldots + h_{n+1} s_{N-1} \\ &= \sum_{k=0}^{N-1} s_k h_{(n-k \mod N)}. \end{aligned}$$

Obviously, this is the standard circular convolution.

4. DATA CLASSIFICATION WITH GRAPH FILTERS

In this section we discuss a potential application of graph filters to data classification. This is an important task in data learning and analysis. Traditionally, the problem of data classification has been studied in machine learning, where multiple approaches using support vector machines and neural networks have been used [16]. More recently, methods utilizing the graph representation of data were proposed, such as minimum cut clustering [17], spectral clustering [6], and label propagation [18, 19].

We propose a novel approach to data classification that interprets a classifier system as a graph filter. The construction of an optimal classifier is viewed and studied as the design of an adaptive graph filter. Our approach is similar to the label propagation, which is a simple, yet efficient technique for two-class classification. It is based on advancing known labels from labeled graph nodes along edges to unlabeled nodes. Usually this propagation of labels is modeled as a stationary discrete-time Markov process, and the graph adjacency matrix is constructed as a probability transition matrix, i.e., $\mathbf{A}_{n,m} \geq 0$ for all n, m, and $\mathbf{A}\,\mathbf{1}_N = \mathbf{1}_N$, where $\mathbf{1}_N$ is a column vector of N ones. Initially known labels determine the initial probability distribution $\mathbf{s}^{(known)}$. For a binary classification problem with only two labels, the resulting labels are determined by the predicted distribution $\mathbf{s}^{(pred)} = \mathbf{A}^P \mathbf{s}^{(known)}$. If $s_n^{(pred)} \leq 1/2$, node v_n is assigned one label, and otherwise the other. The number P of propagations is determined heuristically.

Our DSP_G-based approach has two major distinctions from the label propagation technique. First, we do not require $\mathbf{A}$ to be a stochastic matrix. We only assume that edge weights $\mathbf{A}_{k,m} \geq 0$ are non-negative and indicate similarity or dependency between nodes. In this case, nodes with positive predicted labels $s_n^{(pred)} > 0$ are assigned to one class, and with negative labels to another. Second, instead of propagating labels as in a Markov chain, we construct a filter $h(\mathbf{A}) = h_0\,\mathbf{I} + \ldots + h_L \mathbf{A}^L$ that produces predicted labels

$$\mathbf{s}^{(pred)} = h(\mathbf{A})\mathbf{s}^{(known)}. \tag{11}$$

Under these assumptions, we determine the optimal taps by adaptively constructing the filter using the initially known labels $\mathbf{s}^{(known)}$. This process corresponds to the "training" of the classifier and proceeds as follows. Denote the subset of nodes with initially known labels as $\mathcal{V}^{(known)} \subset \mathcal{V}$; here, the label $s_n^{(known)}$ is set to ± 1 if $v_n \in \mathcal{V}^{(known)}$ and to 0 otherwise. Let us select a smaller subset of training nodes $\mathcal{V}^{(train)} \subsetneq \mathcal{V}^{(known)}$, and construct the training signal $\mathbf{s}^{(train)}$, for which labels $s_n^{(train)}$ are known and set to ± 1 only if $v_n \in \mathcal{V}^{(train)}$; otherwise they are set to 0. Then we assume that a classifier $h(\mathbf{A})$ correctly classifies all nodes in $\mathcal{V}^{(known)}$ using the information about nodes in $\mathcal{V}^{(train)}$, if for each $v_n \in \mathcal{V}^{(known)}$, the nth coefficient of the output $h(\mathbf{A})\mathbf{s}^{(train)}$ has the same sign as the coefficient $s_n^{(known)}$. This condition can be written as

$$\mathbf{D}\, h(\mathbf{A})\mathbf{s}^{(train)} \geq 0, \tag{12}$$

where $\mathbf{D} = \mathbf{diag}\left(\mathbf{s}^{(known)}\right)$ is the diagonal matrix with initially known labels on its main diagonal.

Let us denote the vector of taps in $h(\mathbf{A})$ as $\mathbf{h} = \begin{pmatrix} h_0 & \ldots & h_L \end{pmatrix}^T$. Assuming that L is fixed, we find the taps $\mathbf{h}$ of the optimally trained graph filter satisfying (12) by solving the least-squares minimization problem

$$\begin{aligned} & \mathbf{argmin}\, || \,\mathbf{D}\, h(\mathbf{A})\mathbf{s}^{(train)} - \mathbf{1}_N \,||_2 \\ = \;\; & \mathbf{argmin} || \begin{pmatrix} \mathbf{D}\,\mathbf{A}^0\mathbf{s}^{(train)} & \ldots & \mathbf{D}\,\mathbf{A}^L\mathbf{s}^{(train)} \end{pmatrix} \mathbf{h} - \mathbf{1}_N \,||_2. \end{aligned}$$

After the "training" stage, we use the constructed graph filter $h(\mathbf{A})$ to classify all nodes $v_n \in \mathcal{V}$ using the initially known labels ± 1 of nodes $v_n \in \mathcal{V}^{(known)}$ by analysing the output signal $h(\mathbf{A})\mathbf{s}^{(known)}$. A node v_n is assigned to the class corresponding to labels $+1$ if the nth coefficient of the output signal is positive; and to the class with label -1, if the nth coefficient of the output is negative.

Total number	Fraction of initially labeled images				
of images	2%	5%	10%	20%	30%
1000	26%	59%	72%	83%	87%
5000	79%	84%	89%	93%	95%
10000	87%	89%	92%	93%	96%

Table 1. Accuracy of image classification using adaptive filters.

5. EXPERIMENTS

In this section we illustrate the use of graph filters in data classification, as discussed in Section 4, on the MNIST dataset of images of handwritten digits from 0 to 9 [13]. Examples of these images are shown in Fig. 2(a). All images are grayscale, centered and resized to 28×28 pixels.

We use an 8-nearest-neighbor graph that represents the similarity between images. Each graph node indices an image. It is connected by directed edges to 8 nodes representing 8 most similar images. We measure the similarity between two images as the Frobenius norm of their difference. Fig. 2(b) shows an example graph for 100 randomly selected images, 10 images per each digit.

The classifier system, discussed in Section 4, only performs a binary classification, i.e. it assigns each node to one of two classes. In this experiment we have ten classes corresponding to digits 0 through 9. To extend our binary classifier to this ten-class problem, we perform a one-against-all classification for each class. We use one digit as one class and group other digits into another class, run the binary classifier on this two-class problem, and record the results for each node. After repeating the procedure ten times, each time selecting a new class and grouping others together, we choose the most likely class predicted for each image.

In our experiments, we used datasets with $N = 1000, 5000$, and 10000 images, where each digit was equally represented by $N/10$ images. We randomly selected between 2% and 30% of images as pre-classified ones with known labels. The prediction accuracy was estimated over the remaining, unlabeled images, and calculated as an average of 30 runs. In all experiments, the prediction filters were limited to $L = 15$ taps.

The average prediction accuracy is shown in Table 1. Each result shows the portion of unlabeled images that were classified correctly. As can be observed from the results, even when relatively few images are pre-classified, the adaptively constructed graph filter predicts unknown labels with high accuracy. Furthermore, the larger the overall number of images, the smaller fraction of their labels needs to be known initially to achieve a desired level of classification accuracy.

6. CONCLUSIONS

We have proposed a framework for discrete signal processing on graphs that studies the datasets directly. We defined the fundamental DSP concepts of signals and filters on graphs, studied their important properties, and identified signal and filter spaces. As we showed, our framework extends the traditional discrete signal processing theory to datasets with complex structure. We have also studied the application of graph filters to important problems in data learning and analysis, such as data classification. While this task is traditionally addressed by machine learning, and not DSP, we have demonstrated that it can be interpreted as an adaptive filter design problem, which is a common problem in DSP theory. In our experiments with multi-class classification, we obtained high prediction accuracy even when a relatively small fraction of labels were known initially.

7. REFERENCES

[1] M. Jackson, *Social and Economic Networks*, Princeton Univ., 2008.

[2] M. Newman, *Networks: An Introduction*, Oxford Univ. Press, 2010.

[3] S. L. Lauritzen, *Graphical Models*, Oxford Univ. Press, 1996.

[4] M. I. Jordan, "Graphical models," *Statistical Science (Special Issue on Bayesian Statistics)*, vol. 19, no. 1, pp. 140–155, 2004.

[5] M. Belkin and P. Niyogi, "Using manifold structure for partially labeled classification," 2002.

[6] U. Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[7] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *J. Appl. Comp. Harm. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[8] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2786–2799, 2012.

[9] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Proc.*, accepted.

[10] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory," `http://arxiv.org/abs/cs.IT/0612077`.

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

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

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, 2nd edition, 1999.

[15] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 2nd edition, 1985.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, 2nd edition, 2000.

[17] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *ICML*, 2001, pp. 19–26.

[18] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning ising Gaussian fields and harmonic functions," in *Proc. ICML*, 2003, pp. 58–65.

[19] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *Proc. ICML*, 2006, pp. 985–992.