Robust Semisupervised Graph Classifier Learning With Negative Edge Weights

Gene Cheung, *Senior Member, IEEE*, Weng-Tai Su, *Student Member, IEEE*, Yu Mao, *Student Member, IEEE*, and Chia-Wen Lin, *Fellow, IEEE*

***Abstract*—In a semisupervised learning scenario, (possibly noisy) partially observed labels are used as input to train a classifier in order to assign labels to unclassified samples. In this paper, we construct a complete graph-based binary classifier given only samples' feature vectors and partial labels. Specifically, we first build appropriate similarity graphs with positive and negative edge weights connecting all samples based on internode feature distances. By viewing a binary classifier as a piecewise constant graph signal, we cast classifier learning as a signal restoration problem via a classical maximum *a posteriori* (MAP) formulation. One unfortunate consequence of negative edge weights is that the graph Laplacian matrix $\mathbf{L}$ can be indefinite, and previously proposed graph-signal smoothness prior $\mathbf{x}^T\mathbf{L}\mathbf{x}$ for candidate signal $\mathbf{x}$ can lead to pathological solutions. In response, we derive a minimum-norm perturbation matrix $\mathbf{\Delta}$ that preserves $\mathbf{L}$'s eigenstructure—based on a fast lower-bound computation of $\mathbf{L}$'s smallest negative eigenvalue via a novel application of the Haynsworth inertia additivity formula—so that $\mathbf{L}+\mathbf{\Delta}$ is positive semidefinite, resulting in a stable signal prior. Further, instead of forcing a hard binary decision for each sample, we define the notion of generalized smoothness on graphs that promotes ambiguity in the classifier signal. Finally, we propose an algorithm based on iterative reweighted least squares that solves the posed MAP problem efficiently. Extensive simulation results show that our proposed algorithm outperforms both SVM variants and previous graph-based classifiers using positive-edge graphs noticeably.**

***Index Terms*—Graph signal processing, signal restoration, classifier learning.**

I. Introduction

A FUNDAMENTAL problem in machine learning is *semi-supervised learning* [1]: given partially observed labels (possibly corrupted by noise) as input, train a classifier so that unclassified samples can also be appropriately assigned labels. Among many approaches to the problem is a class of graph-based methods [2]–[10] that model each sample as a node in a graph, connected to other nodes via undirected edges, with weights that reflect pairwise distances in a high-dimensional feature space. See Fig. 1 for an example of a graph with eight nodes (samples) in a two-dimensional feature space. Establishing a graph representation of the data means that intrinsic properties of the graph spectrum (e.g., low graph frequencies that are eigenvectors of the graph Laplacian matrix) can be exploited for label assignment via spectral graph theory [11].

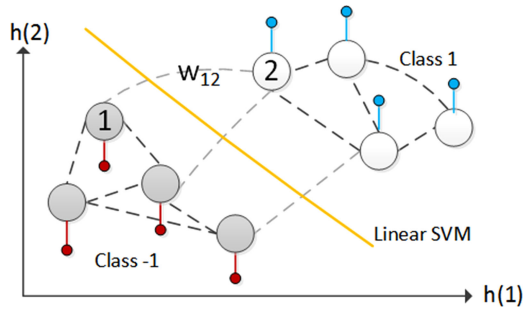


Fig. 1. Example of a graph classifier and linear SVM in 2-dimensional feature space. The graph $\mathcal{G}$ contains nodes $\mathcal{N}$ representing samples, and edges $\mathcal{E}$ with weights $w_{i,j}$ that reflect feature space distance between nodes. The classifier graph signal takes on binary values: 1 (blue spikes) and -1 (red spikes).

In this paper, extending previous studies we construct a complete graph-based binary classifier given only samples' feature vectors and partial labels, considering in addition *negative edge weights*. Conventional formulations in graph signal processing (GSP) [12] use positive edge weights to signify inter-node *similarity*. However, negative edge weights can signify *dissimilarity*: $w_{i,j}=-1$ means samples x_i and x_j are expected to take on different values, i.e., $|x_i-x_j|$ should be large. Incorporating pairwise dissimilarity into the graph should intuitively be beneficial during classifier learning. For example, if edge weight $w_{1,2}$ is assigned -1 in Fig. 1, then from the graph $\mathcal{G}$ itself without any label information, one already expects x_1 and x_2 to be assigned opposite labels in a binary classifier.[1]

Specifically, we first build appropriate similarity graphs with positive and negative edge weights connecting all samples based on inter-node feature distances. Interpreting a binary classifier as a piecewise constant (PWC) graph signal, we cast classifier learning as a signal restoration problem via a classical *maximum a posteriori* (MAP) formulation [14]. We show that a graph Laplacian matrix $\mathbf{L}$ with negative edge weights can be indefinite, and a common graph signal prior called *graph Laplacian regularizer* [15]–[22] $\mathbf{x}^T\mathbf{L}\mathbf{x}$ for candidate signal $\mathbf{x}$—measuring

Manuscript received June 12, 2017; revised July 20, 2017, November 26, 2017, and March 15, 2018; accepted March 16, 2018. Date of publication March 22, 2018; date of current version September 7, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael Rabbat. *(Corresponding author: Gene Cheung.)*

G. Cheung and Y. Mao are with the National Institute of Informatics, Graduate University for Advanced Studies, Tokyo 101-8430, Japan (e-mail: cheung@nii.ac.jp; vimystic@gmail.com).

W.-T. Su and C.-W. Lin are with the National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: wengtai2008@hotmail.com; cwlin@ee.nthu.edu.tw).

Digital Object Identifier 10.1109/TSIPN.2018.2819018

[1] [13] provides a physical interpretation of a signed graph with both positive and negative edge weights, where a negatively weighted edge is interpreted as a repulsive spring in a mass-spring system.

signal smoothness with respect to the underlying graph—can lead to pathological solutions. In response, we derive a minimum-norm perturbation matrix $\mathbf{\Delta}$ that preserves $\mathbf{L}$'s eigen-structure, so that $\mathbf{L}+\mathbf{\Delta}$ is positive semi-definite (PSD), resulting in a stable signal prior. To efficiently compute an approximate $\mathbf{\Delta}$, we propose a fast recursive algorithm that identifies a lower bound for the smallest negative eigenvalue of $\mathbf{L}$ via a novel application of the *Haynsworth Inertia Additivity formula* [23].

Second, instead of forcing a hard binary decision for each sample, we define the notion of *generalized smoothness* on graph—an extension of *total generalized variation* (TGV) [24] to the graph signal domain—that promotes the right amount of ambiguity in the classifier signal. Estimated labels with low confidence (signal values close to zero) can be removed thereafter, thus improving the overall classification performance.

Finally, we propose an algorithm based on *iterative reweighted least squares* (IRLS) [25] that efficiently solves the posed MAP problem for the noisy label scenario. Extensive simulation results show that our proposed algorithm outperforms SVM variants, a well-known robust classifier in the machine learning literature called *RobustBoost* [26], and graph-based classifiers using positive-edge graphs noticeably for both noiseless and noisy label scenarios.

The outline of the paper is as follows. We first overview related works in Section II. We define a graph signal smoothness prior and formulate a MAP optimization objective in Section III. Adding a generalized smoothness prior, we formulate a second objective in Section IV, which is useful for applications that can tolerate a small portion of rejected data classified with low confidence. We discuss graph construction using positive and negative edges in Section V-A. In Section VI, we derive an appropriate perturbation matrix $\mathbf{\Delta}$ such that $\mathbf{L}+\mathbf{\Delta}$ is PSD. In Section VII, we describe a fast algorithm to approximate the best $\mathbf{\Delta}$. In Section VIII, we present our IRLS-based algorithm to solve our two objectives. Finally, we present experimental results and conclusions in Section IX and X, respectively.

II. Related Works

A. Robust Graph-based Classifier Learning

There exists a wide range of approaches to noisy label classifier learning, including theoretical (e.g., label propagation in [27]) and application-specific (e.g., emotion detection using inference algorithm based on multiplicative update rule [28]). In this paper, we focus specifically on graph-based classifiers, which has been studied extensively in the past decade [2]–[10]. An early work [2] used the graph Laplacian matrix to construct an interpolation filter for the input partial labels to compute missing labels. Another seminal work [3] proposed two graph-based formulations for the noisy and noiseless label learning scenarios using the graph smoothness prior $\mathbf{x}^\top\mathbf{L}\mathbf{x}$. While the origin of our MAP formulation can be traced back to [3] and enjoys similar computation benefit of solving simple linear systems at each iteration, neither [2] nor [3] handled the case when negative edges are present to denote inter-node dissimilarity, which is one main focus of this paper.

Recent advent in graph signal processing (GSP) [12] has led to the development of transforms [29] and wavelets for signals that live on irregular data kernels described by graphs. Using these developed tools, one can design learning algorithms via assumptions in the transform domain [4]–[6]. For example, [6] assumed that the l_1-norm of the graph wavelet coefficients is small as a signal prior for graph signal (classifier) reconstruction. However, to-date critically sampled, compact support, orthogonal perfect reconstruction graph wavelets only exist for very special graphs like bipartite or k-colorable graphs.[2] Thus, for signals on a general graph, to use these wavelets one must first approximate the original graph with a series of bipartite graphs [31], [32]. This means that the l_1-norm is difficult to apply across different stages of bipartite graph approximation. As a representative graph wavelet scheme, we will show in our experiments that our proposed smoothness prior leads to better performance than an over-complete wavelet in [6].

One can also approach the semi-supervised learning problem from a sampling perspective: available labels are observed signal samples, and missing samples are interpolated using a bandlimited signal assumption [17], [33], [34]. There are two problems to this approach. First, practical graph signals are often not strictly bandlimited. Second, observed labels are often corrupted by noise, and straightforward interpolation schemes would lead to error propagation. We will show in our experiments that our proposed classifier scheme outperforms [17] noticeably.

Compared to previous graph-based classifiers, we make the following three key technical contributions. First, we construct a similarity graph with positive and negative edge weights, latter of which signify inter-node dissimilarities, given only the samples' feature vectors. Second, for the graph signal smoothness prior to be numerically stable in a classical MAP formulation, we derive a minimum-norm perturbation matrix $\mathbf{\Delta}$ that preserves the eigen-structure of the original Laplacian $\mathbf{L}$, so that $\mathbf{L}+\mathbf{\Delta}$ is PSD via a novel application of the Haynsworth inertia additivity formula [23]. Third, we extend the generalized smoothness notion in TGV [24] to the graph signal domain, in order to promote appropriate degree of ambiguity in the classifier solution to lower overall classification error rate.

B. Graph Signal Image Restoration

More generally, graph signal priors have been used for image restoration problems such as denoising [15], [22], interpolation [16]–[18], bit-depth enhancement [19] and JPEG de-quantization [20], [21]. The common assumption is that the desired graph signal is smooth or bandlimited with respect to an appropriate graph with positive edge weights that reflect inter-pixel similarity. Instead of posing an optimization, graph filters can also be designed directly for image denoising [35], edge-enhancing [36] and image magnification [37]. In contrast, by introducing negative edges into the graph, we incorporate dissimilarity information into a classical MAP formulation like [3]

[2]Recently, [30] proposed a M-channel critically sampled filter bank for graph signals, where the synthesis filters are replaced by interpolation operators.

and study methods to resolve the graph Laplacian's indefiniteness. Further, we define a generalized notion of graph smoothness for signal restoration specifically for classifier learning.

C. Negative Edge Weights in Graphs

Recent studies in the control community have examined the conditions where one or more negative edge weights would induce a graph Laplacian to be indefinite [38], [39]. The analysis, however, rests on an assumption that there are no cycles in the graph with more than one negative edge, which is too restrictive for binary classifier graphs.

[40] considered a signed social network where each edge denotes either a cohesive (positive edge weight) or oppositive (negative edge weight) relationship between two vertices. The goal is to identify similar groups within the graph, and thus is akin to a distributed clustering problem, which is unsupervised by definition. In contrast, our goal is to restore a classifier graph signal from partially observed labels, which is a semi-supervised learning problem.

A notable recent work [13] argued that the eigenvectors of the original indefinite graph Laplacian with negative edges are more intuitive and useful than the eigenvectors of the signed graph Laplacian [41] for spectral clustering. The key argument is that the shapes of the first eigenvectors of the original indefinite graph Laplacian are more pronounced at the negative edge endpoints, and the condition numbers are more favorable. In Section VIII, we also stress the usefulness of the eigenvectors of the original indefinite graph Laplacian, but are using them for classifier learning rather than clustering.

III. Problem Formulation I: Graph Smoothness

A. Graph Definition

We first introduce definitions in GSP needed to formulate our problem. A graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ has a set $\mathcal{V}$ of N nodes and a set $\mathcal{E}$ of M edges, where we assume $M \ll N^2$, i.e., the graph is sparse. Each edge $(i,j) \in \mathcal{E}$ connecting nodes i and j is undirected and has an associated scalar weight $w_{i,j}$. In this paper, we assume that $w_{i,j}$ can be positive *or* negative; a negative $w_{i,j}$ means that the samples in nodes i and j are *dissimilar*—the samples are expected to have very different values.

A graph signal $\mathbf{x} \in \mathbb{R}^N$ on $\mathcal{G}$ is a discrete signal of dimension N—one value x_i for each node (sample) i in $\mathcal{V}$. If we restrict $\mathbf{x}$ to be a binary classifier, then x_i can only take on one of two values specifying the class to which sample i belongs, i.e., $\mathbf{x} \in \{-1, 1\}^N$. However, letting the reconstructed signal $\hat{\mathbf{x}}$ take on real values $\mathbb{R}^N$ allows us to introduce ambiguity in the reconstruction instead of forcing hard binary decisions; this is discussed in Section IV.

B. Graph Spectrum

Given edge weight (adjacency) matrix $\mathbf{W}$, we define a diagonal *degree matrix* $\mathbf{D}$, where $d_{i,i} = \sum_j w_{i,j}$. A *combinatorial graph Laplacian matrix* $\mathbf{L}$ is simply $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [12]. $\mathbf{L}$ is symmetric, which means that it can be eigen-decomposed into:

$$\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{1}$$

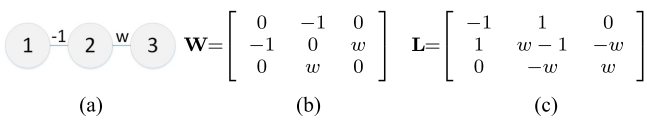


Fig. 2. Example of a 3-node graph with negative edges. (a) 3-node graph. (b) adjacency $\mathbf{W}$. (c) graph Laplacian $\mathbf{L}$.

where $\mathbf{\Lambda}$ is a diagonal matrix containing real eigenvalues λ_k (not necessarily unique), and $\mathbf{V}$ is an eigen-matrix composed of orthogonal eigenvectors $\mathbf{v}_i$ as columns. If edge weights $w_{i,j}$ are strictly positive, then one can show that $\mathbf{L}$ is PSD, meaning that $\lambda_k \geq 0, \forall k$ and $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$, $\forall \mathbf{x}$. Non-negative eigenvalues λ_k can be interpreted as *graph frequencies*, and eigenvectors $\mathbf{v}_k$ interpreted as corresponding graph frequency components. Together they define the *graph spectrum* for graph $\mathcal{G}$. To avoid confusion, we denote a graph Laplacian for a positive-edge-only graph as $\mathbf{L}^+$.

In this paper, we consider also negative edge weights $w_{i,j} < 0$, and thus eigenvalues λ_k can be negative and $\mathbf{L}$ can be indefinite. It is then hard to interpret $\mathbf{L}$'s eigenvalues λ_k as frequencies, and in general, it is desirable to have a variational operator that is PSD. Thus, we seek to add a *perturbation matrix* $\mathbf{\Delta}$ to $\mathbf{L}$ such that the resultant *generalized graph Laplacian* $\mathbf{L}_g = \mathbf{L} + \mathbf{\Delta}$ is PSD. We address this problem of finding an "optimal" $\mathbf{\Delta}$ in Section VIII.

C. Graph Signal Smoothness Prior for Positive Graphs

For graph $\mathcal{G}$ with positive edge weights, signal $\mathbf{x}$ is *smooth* if each sample x_i on node i is similar to x_j on neighboring nodes j with large $w_{i,j}$. In the graph frequency domain, smoothness means that $\mathbf{x}$ contains mostly low graph frequency components; i.e., coefficients $\boldsymbol{\alpha} = \mathbf{V}^T\mathbf{x}$ are very small for high frequencies. The smoothest signal is the constant vector $\mathbf{1}$—the first eigenvector $\mathbf{v}_1$ for $\mathbf{L}^+$ corresponding to the smallest eigenvalue $\lambda_1 = 0$.

Mathematically, we can write that a signal $\mathbf{x}$ is smooth if its *graph Laplacian regularizer* $\mathbf{x}^T\mathbf{L}^+\mathbf{x}$ is small [12], [22]. Graph Laplacian regularizer can be expressed as:

$$\mathbf{x}^T\mathbf{L}^+\mathbf{x} = \sum_{(i,j)\in\mathcal{E}} w_{i,j}\,(x_i - x_j)^2 = \sum_k \lambda_k\,\alpha_k^2 \tag{2}$$

Because $\mathbf{L}^+$ is PSD, $\mathbf{x}^T\mathbf{L}^+\mathbf{x}$ is lower-bounded by 0.

We can also interpret the graph Laplacian regularizer as a *signal prior* in a Bayesian formulation; i.e., the probability $Pr(\mathbf{x})$ of observing a signal $\mathbf{x}$ is:

$$Pr(\mathbf{x}) \propto \exp\left(-\frac{\mathbf{x}^T\mathbf{L}^+\mathbf{x}}{\sigma^2}\right) \tag{3}$$

where σ is a parameter.

D. Graph Signal Smoothness Prior for Signed Graphs

When considering a more general *signed graph* with positive *and* negative edge weights, graph signal smoothness priors proposed in the GSP literature may become problematic. Consider the 3-node line graph in Fig. 2 for $w = 1$ or -1. Using

smoothness prior $\mathbf{x}^T\mathbf{L}\mathbf{x}$ in (2), we get:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = -1(x_1 - x_2)^2 + w(x_2 - x_3)^2 \tag{4}$$

which promotes a *large* difference between nodes 1 and 2 and a large/small difference between nodes 2 and 3 depending if $w = -1$ or $w = 1$. This prior thus agrees with our intuitive notions of inter-node (dis)similarity embedded in edge weights. However, direct use of $\mathbf{x}^T\mathbf{L}\mathbf{x}$ can lead to numerical problems. For example, $x_1 = \infty$ and $x_2 = -\infty$ would result in $-\infty$, which is a pathological optimal solution for a minimization problem.

Alternatively, one separate GSP approach—based on algebraic theory in traditional digital signal processing [9], [42]–[44]—interprets the adjacency matrix $\mathbf{W}$ as a shift operator. A graph signal smoothness prior can thus be defined as the difference between the signal $\mathbf{x}$ and its shifted version $\mathbf{W}\mathbf{x}$; specifically, $\|\mathbf{x} - \frac{1}{|\lambda_{\max}(\mathbf{W})|}\mathbf{W}\mathbf{x}\|_p^p$ given a positive integer p was proposed in [9]. However, when edge weights are negative, this smoothness prior can be insensible. For the same 3-node graph in Fig. 2, assuming $p = 2$ and $\lambda_{\max}(\mathbf{W}) = 1$ as done in [44] for simplicity of illustration, the smoothness prior when $w = -1$ is:

$$\|\mathbf{x} - \mathbf{W}\mathbf{x}\|_2^2 = \|(\mathbf{I} - \mathbf{W})\mathbf{x}\|_2^2 = \left\| \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right\|_2^2$$
$$= (x_1 + x_2)^2 + (x_1 + x_2 + x_3)^2 + (x_2 + x_3)^2$$

Given two negative edges that explicitly specify inter-node dissimilarity, it is hard to interpret why the prior should promote three small sums of signal values. For example, $\mathbf{x} = (\rho, \rho + 100, \rho)$ for a large $\rho > 0$ is a signal with a large difference (100) among each connected pair—agreeing with the dissimilarity notion specified by the two negative edges, but would compute to a large prior.

Suppose a *total variation* (TV) approach [45] is taken instead, so that a smoothness prior using $\mathbf{L}$ but based on l_1-norm is used instead; i.e., $|\mathbf{L}\mathbf{x}|$. Using the same 3-node graph in Fig. 2 with $w = 1$, $|\mathbf{L}\mathbf{x}|$ is:

$$|\mathbf{L}\mathbf{x}| = \left| \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right| = \begin{vmatrix} x_2 - x_1 \\ x_1 - x_3 \\ x_3 - x_2 \end{vmatrix}$$

In other words, the prior tries to minimize the difference between every node pair. For example, a signal $\mathbf{x} = (\rho, \rho, \rho)$ for some $\rho > 0$ results in $|\mathbf{L}\mathbf{x}| = 0$, but the negative edge $(1, 2)$ actually specifies a large difference between x_1 and x_2. Thus this prior is also not sensible.

One final alternative we consider is to adopt a *signed graph Laplacian* definition in [41], where $\mathbf{L}^s = \mathbf{D}^s - \mathbf{W}$ and $D^s_{i,i} = \sum_j |w_{i,j}|$. Using $\mathbf{x}^\top\mathbf{L}^s\mathbf{x}$ as a smoothness prior, for $w = 1$ we get:

$$\mathbf{x}^T\mathbf{L}^s\mathbf{x} = (x_1 + x_2)^2 + (x_2 - x_3)^2 \tag{5}$$

$w_{1,2} = -1$ means x_1 and x_2 are expected to be very different, but a small $(x_1 + x_2)^2$ only means that x_1 and x_2 have similar magnitude but opposite signs. For example, $x_1 = \rho$ and $x_2 = -\rho$ for very small $\rho > 0$ will also compute to $(x_1 + x_2)^2 = 0$. Thus this prior is also not sensible in the general case.

Having demonstrated the shortcomings in alternative smoothness priors in the literature, in this paper we choose to use the graph Laplacian regularizer $\mathbf{x}^T\mathbf{L}\mathbf{x}$ (2) that agrees with our intuitive notions of inter-node (dis)similarity specified by edge weights, but perturb $\mathbf{L}$ with $\mathbf{\Delta}$ so that $\mathbf{L}_g = \mathbf{L} + \mathbf{\Delta}$ is PSD. We discuss this in details in Section VIII.

E. Binary Classifier Graph Signal Restoration

Given defined Bayesian graph signal smoothness prior (3), we can now formally define a restoration problem for a binary classifier via a MAP formulation. First, to model noise in binary labels, we adopt a uniform noise model [46], where the probability of observing $y_i = x_i$, $1 \le i \le K$, is $1 - p$, and p otherwise; i.e.,

$$Pr(y_i|x_i) = \begin{cases} 1 - p & \text{if } y_i = x_i \\ p & \text{o.w.} \end{cases} \tag{6}$$

This noise model is motivated by an observation in social media analysis, when labels are assigned manually by non-experts via *crowd-sourcing* [46]—i.e., employ non-experts online to assign labels to data at a very low cost. Because non-experts are often unreliable, observations $\mathbf{y}$ may result in label errors that are uniform and independent.

The probability of observing a noise-corrupted $\mathbf{y}$, $\mathbf{y} \in \{-1, 1\}^K$, given ground truth $\mathbf{x}$, $\mathbf{x} \in \{-1, 1\}^N$, where $K < N$, is:

$$Pr(\mathbf{y}|\mathbf{x}) = p^k(1 - p)^{K-k}$$
$$k = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 \tag{7}$$

where $\mathbf{H} \in \{0, 1\}^{K \times N}$ is a *sampling matrix* that picks K observations from N total samples. (7) serves as the likelihood term given label noise model in (6). The negative log of this likelihood $Pr(\mathbf{y}|\mathbf{x})$ can be rewritten as:

$$-\log Pr(\mathbf{y}|\mathbf{x}) = k \underbrace{(\log(1 - p) - \log(p))}_{\gamma} - K \log(1 - p) \tag{8}$$

Because the second term is a constant for fixed K and p, we can ignore it during minimization.

Given prior (3) and likelihood (7), we can formulate a MAP problem as follows:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 + \mu_1 \, \mathbf{x}^T (\mathbf{L} + \mathbf{\Delta})\mathbf{x} \tag{9}$$

where μ_1 is a parameter that trades off the importance between the likelihood term and the signal prior, and $\mathbf{\Delta}$ is the perturbation matrix to be discussed in Section VI and VII. We discuss how (9) can be solved efficiently in Section VIII.

IV. Problem Formulation II: Generalized Smoothness

We next describe a generalized version of the graph signal smoothness prior (2) for classifier signal reconstruction, when an application can tolerate rejection of a small portion of samples that are estimated with low confidence. We then formulate a second objective that considers in addition this new prior.

A. Positive Edge Weights for Generalized Smoothness

It is well known that using TV to restore a 2D image would often result in unpleasant "staircase" effect in the recovery, if the ground truth image has a linear slope. To alleviate the staircase effect, TGV is proposed [24], which defines a higher-order notion of smoothness. This generalized notion was used in [14] for graph signals using *positive* edge weights. Specifically, positive edge graph Laplacian $\mathbf{L}^+$ is related to the second derivative of continuous functions [12], and so $\mathbf{L}^+\mathbf{x}$ computes the second-order difference on graph signal $\mathbf{x}$.

As an example, the 3-node line graph in Fig. 2 with $w = 1$ has the following $\mathbf{L}^+$:

$$\mathbf{L}^+ = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \tag{10}$$

Using the second row $\mathbf{L}^+_{2,:}$ of $\mathbf{L}^+$, we can compute the second-order difference at node x_2:

$$\mathbf{L}^+_{2,:}\mathbf{x} = -x_1 + 2x_2 - x_3 \tag{11}$$

On the other hand, the definition of second derivative of a function $f(x)$ is:

$$f''(x) = \lim_{h\to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{12}$$

We see that (11) and (12) are computing the same quantity (with a sign change) in the limit.

Hence if $|\mathbf{L}^+\mathbf{x}|$ is small, then the second-order difference of $\mathbf{x}$ is small, or the first-order difference of $\mathbf{x}$ is smooth or changing slowly. In other words, the *gradient* of the signal is smooth with respect to the graph. We express this notion by stating that the square of the l_2-norm of $\mathbf{L}^+\mathbf{x}$ is small:

$$\|\mathbf{L}^+\mathbf{x}\|_2^2 = \mathbf{x}^T(\mathbf{L}^+)^T\mathbf{L}^+\mathbf{x} = \mathbf{x}^T(\mathbf{L}^+)^2\mathbf{x} = \sum_i (\lambda_i^+)^2\alpha_i^2 \tag{13}$$

where (13) is true since $\mathbf{L}^+$ is symmetric by definition.[3]

B. Negative Edges for Generalized Smoothness

We show that using an indefinite graph Laplacian $\mathbf{L}$ with negative edges to define generalized smoothness $\mathbf{x}^T\mathbf{L}^2\mathbf{x}$ is problematic. One reason is that while the frequency components are preserved,

$$\mathbf{L}^2 = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T \tag{14}$$

[3]Note that powers of the graph Laplacian $\mathbf{L}$ have been used previously to achieve signal smoothness within a local neighborhood [18], [47].

frequency preferences are reordered in $\mathbf{L}^2$; i.e., negative eigenvalues λ_k in $\mathbf{L}$ are now sorted with positive eigenvalues in magnitude as λ_k^2 in $\mathbf{L}^2$. It is hard to explain how this reordering of frequency components according to magnitude λ_k^2 is beneficial for signal restoration.

To illustrate the potential problem of negative edges in generalized smoothness in the nodal domain, consider again the three-node line graph in Fig. 2, where $w = 1$. The corresponding second row of the graph Laplacian $\mathbf{L}$ is:

$$\mathbf{L}_{2,:} = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{15}$$

This means that when we compute the generalized smoothness $|\mathbf{L}\mathbf{x}|$ at x_2, we get $|\mathbf{L}_{2,:}\mathbf{x}| = |x_1 - x_3|$; i.e., the generalized smoothness at x_2 does not actually depend on the value of x_2, which is not sensible.

C. Objective Function

If we choose to include the new generalized smoothness prior to our previous objective (9) to promote ambiguity in the solution, the new objective now has two priors:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0 + \mu_1\, \mathbf{x}^T\, \mathbf{L}_g\, \mathbf{x} + \mu_2\, \mathbf{x}^T\, (\mathbf{L}^+)^2\, \mathbf{x} \tag{16}$$

where $\mathbf{L}_g = \mathbf{L} + \mathbf{\Delta}$.

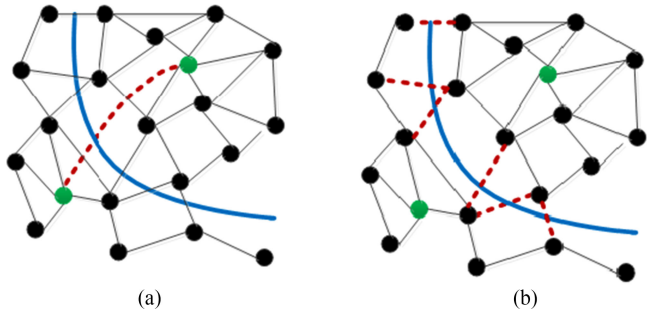
We interpret the two smoothness terms in the context of binary classification. We know that the true signal $\mathbf{x}$ is indeed *piecewise constant* (PWC); each true label x_i is binary, and labels of the same class cluster together in the same feature subspace. The graph signal smoothness term in (3), analogous to the TV prior [45] in image restoration [22], promotes a PWC signal $\hat{\mathbf{x}}$ during reconstruction, as empirically demonstrated in previous graph signal restoration works [20]–[22]. Hence the smoothness prior is appropriate.

Recall that the purpose of TGV [24] is to avoid over-smoothing a linear slope (ramp) in an image, when a TV prior is used. A ramp in the reconstructed signal $\hat{\mathbf{x}}$ in our classification context would mean an assignment of label other than -1 and 1, which can reflect the *confidence level* in the estimated label; e.g., a computed label $\hat{x}_i = 0.2$ would mean the classifier has determined that event i is more likely to be 1 than -1, but the confidence level is not high. We can thus conclude that *the generalized smoothness prior can promote an appropriate amount of ambiguity in the classification solution instead of forcing the classifier to make hard binary decisions.*

V. Graph Construction

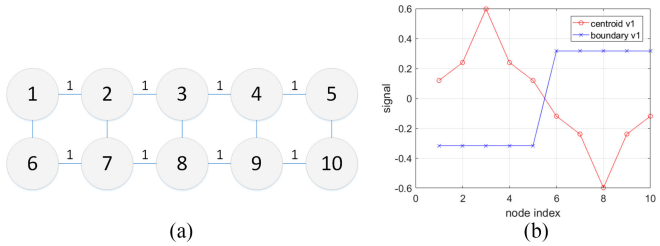
A. Construct Graph with Negative Edges

In a semi-supervised learning problem, often we are given only a *feature vector* for each sample in a high-dimensional feature space, with (possibly noisy) labels assigned to a small sample subset. To compute a graph-based binary classifier, we must first construct an appropriate graph where edges reflect inter-node (dis)similarity relationships based on features. We discuss our graph construction strategy here. *To the best of our knowledge, the construction of similarity graphs with both*



(a) (b)

Fig. 3. Two constructions of similarity graphs with negative edges: (a) connect cluster centroids (in red), and (b) connect boundary nodes of two clusters. A blue line denotes the border between two clusters in each graph.



(a) (b)

Fig. 4. Example of a 10-node graph. Nodes 1 to 5 (6 to 10) belong to one class and are connected by edges of weight 1. For centroid-based graph, nodes between the two classes are connected by edges of weight 0.1, except $(3, 8)$ which is connected by an edge of weight -1. For boundary-based graph, nodes between the two classes are connected by edges of weight -1. (a) 10-node graph. (b) first eigenvectors.

positive and negative edges from feature vectors for classification has not been studied in the graph-based classifier literature.

We first construct a graph $\mathcal{G}$ with nodes $\mathcal{V}$ representing N samples. For each sample i we assume that there exists a corresponding feature vector $\mathbf{h}_i$ of dimension Q. Then we can assign *positive* edge weight $w_{i,j}$ using a Gaussian kernel:

$$w_{i,j} = \exp\left(-\frac{(\mathbf{h}_i - \mathbf{h}_j)^T \boldsymbol{\Xi} (\mathbf{h}_i - \mathbf{h}_j)}{\sigma_h^2}\right) \tag{17}$$

where σ_h is a parameter. $\boldsymbol{\Xi}$ is a $Q \times Q$ diagonal matrix, where $\Xi_{i,i}$ is a feature weight for the i-th feature. We assign positive edges with weights $w_{i,j}$ to connect node i to its ω's nearest neighbors[4] j.

This positive weight assignment is similar to those in previous works on spectral clustering [48] and graph-based classifier learning [10], [14], where a closer distance in the Q-dimensional feature space leads to a larger edge weight. To improve clustering/classification performance, feature parameter $\Xi_{i,i}$ is set larger if the i-th feature is more discriminate. For optimization of feature parameters $\Xi_{i,i}$—which is not the focus of this paper—see [49].

We propose two methods to insert negative edges into an initial graph with only positive edges. The first results in a graph $\mathcal{G}^1$ that is robust to label noise but not precise in designating inter-node dissimilarity relationships, and the second results in a graph $\mathcal{G}^2$ that is precise in designating dissimilarity relationships but not robust to noisy labels.

In the first *centroid-based* method, we divide the samples into two similar clusters based on observed labels (or estimated labels from previous iteration), then connect the two respective *centroids* with negative edges, as illustrated in Fig. 3(a). The idea is that even if some sample labels are corrupted by noise, given that the two clusters are sufficiently different, then at least the *centers* (centroids) of the two clusters are expected to have opposing labels. In one sense, this is analogous to the k-neighborhood graph with positive edges, where here we connect nodes that we deem are most dissimilar with negative edges. However, ideally the *boundaries* of the clusters should define the label crossover points. Thus $\mathcal{G}^1$ is robust but not precise.

[4]If this relationship is not symmetric, i.e., if i is one of ω closest neighbors to j but j is not one of ω closest neighbors to i, then we keep edge (i, j) of weight $w_{i,j}$ anyway. Thus each node has $\geq \omega$ neighbors.

In the second *boundary-based* method, we connect the boundary samples of the two clusters with negative edges, as illustrated in Fig. 3(b). This construction leads to enhancement of the cluster boundaries during filtering and thus improves classification performance. However, given that the labels are noise-corrupted, the exact locations of the boundaries are initially uncertain, and hence $\mathcal{G}^2$ is not precise.

We thus propose to combine the two graphs as follows and iterate. For each graph, we construct a graph Laplacian $\mathbf{L}_i$ and compute a suitable perturbation matrix $\boldsymbol{\Delta}_i$ (to be discussed in Sections VI and VII). We then combine them as a convex combination:

$$\mathbf{L}^* = \beta\,(\mathbf{L}_1 + \boldsymbol{\Delta}_1) + (1 - \beta)\,(\mathbf{L}_2 + \boldsymbol{\Delta}_2) \tag{18}$$

where $0 \leq \beta \leq 1$ is a parameter that changes from 1 to 0 as we iterate. Thus $\mathbf{L}^*$ will be robust early in the iterations, and precise late in the iterations.

B. Example of Graph with Negative Edges

As illustration, we consider a simple example in Fig. 4(a): a 10-node graph where nodes 1 through 5 are similar and are connected by edges of weight 1, and nodes 6 to 10 are similar. For the centroid-based graph, nodes between the two classes are connected by edges of weight 0.1, except the two respective centroids $(3, 8)$ that are connected by an edge of weight -1. Graph Laplacian $\mathbf{L}$ for this graph has smallest eigenvalue -0.8, and the corresponding eigenvectors $\mathbf{v}_1$ is shown in Fig. 4(b). We see that the maximum and minimum of $\mathbf{v}_1$ are located at the endpoints $(3, 8)$ of the lone negative edge, and thus during signal restoration, the prior will promote opposite label assignments for samples 3 and 8, which agrees with the dissimilarity notion of negative edges. *More generally, low graph frequency components $\mathbf{v}_i$ of an indefinite graph Laplacian $\mathbf{L}$ are useful in restoring signal $\mathbf{x}$, leveraging inter-node dissimilarity information embedded in negative edges.* This point is also argued in [13] for spectral clustering.

For the boundary-based graph, boundary nodes between the two classes are all connected by edges of weight -1. Graph Laplacian $\mathbf{L}$ for this graph has smallest eigenvalue -2, and the corresponding eigenvectors $\mathbf{v}_1$ is also shown in Fig. 4(b). We see that each pair of boundary nodes across two clusters have the same opposite values, and thus during restoration, the prior

will promote opposite label assignments for all negative-edge-connected pairs.

VI. Finding a Perturbation Matrix

We now address the problem of identifying a perturbation matrix $\mathbf{\Delta}$ such that $\mathbf{L}+\mathbf{\Delta}$ is PSD. To impart intuition on the effects of $\mathbf{\Delta}$ on the eigenvalues of $\mathbf{L}+\mathbf{\Delta}$, consider first the *Weyl's inequality* [50]. Let a Hermitian matrix $\mathbf{L}\in\mathbb{R}^{N\times N}$ have spectral decomposition $\mathbf{L}=\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ as described in (1), with eigenvalues λ_k along the diagonal of diagonal matrix $\mathbf{\Lambda}$. Let $\mathbf{\Delta}$ be a Hermitian matrix with eigenvalues $\gamma_1\leq\ldots\leq\gamma_N$. Weyl's inequality states that $\mathbf{L}+\mathbf{\Delta}$ has eigenvalues $\nu_1\leq\ldots\leq\nu_N$, such that:

$$\lambda_i+\gamma_1\leq\nu_i\leq\lambda_i+\gamma_N \tag{19}$$

In words, (19) states that the i-th eigenvalue ν_i of $\mathbf{L}+\mathbf{\Delta}$ is the i-th eigenvalue λ_i of $\mathbf{L}$ shifted by an amount in the range $[\gamma_1,\gamma_N]$. Assume that $\lambda_1<0$, hence $\mathbf{L}$ is indefinite. The Weyl's inequality then implies that for $\mathbf{L}+\mathbf{\Delta}$ to be PSD: i) a necessary (but not sufficient) condition is $\gamma_N\geq-\lambda_1$; ii) a sufficient (but not necessary) condition is $\gamma_1\geq-\lambda_1$.

Obviously, there exists an infinite number of feasible solutions $\mathbf{\Delta}$. Thus a well chosen criterion must be used to differentiate them.

A. Matrix Perturbation: Minimum-Norm Criterion

One reasonable choice is the *minimum-norm criterion*, i.e., find $\mathbf{\Delta}$ with the smallest norm such that $\mathbf{L}+\mathbf{\Delta}$ is PSD:

$$\min_{\mathbf{\Delta}}\|\mathbf{\Delta}\|\ \text{ s.t. }\mathbf{x}^T(\mathbf{L}+\mathbf{\Delta})\mathbf{x}\geq0,\ \forall\mathbf{x} \tag{20}$$

where $\|.\|$ is a *unitarily invariant norm* on $\mathbb{R}^{N\times N}$; i.e., $\|\mathbf{U}\mathbf{\Delta}\mathbf{V}\|=\|\mathbf{\Delta}\|$ for all orthogonal $\mathbf{U}$ and $\mathbf{V}$.

It turns out that the solution to (20) is a special case of Theorem 5.1 in [51], which we rephrase as follows. Assume that $\mathbf{L}$ has exactly p negative eigenvalues. Theorem 5.1 in [51] states that the optimal perturbation matrix $\mathbf{\Delta}$ with minimum norm $\|\mathbf{\Delta}\|$, such that $\mathbf{L}+\mathbf{\Delta}$ is PSD, is:

$$\mathbf{\Delta}=\mathbf{V}\,\mathrm{diag}\,(\boldsymbol{\tau})\,\mathbf{V}^T \tag{21}$$

where $\boldsymbol{\tau}=[\tau_1,\ldots,\tau_n]$:

$$\tau_i=\begin{cases}-\lambda_i & \text{if } 1\leq i\leq p\\ 0 & \text{o.w.}\end{cases} \tag{22}$$

See [51] for a complete proof. We only make a few important observations. First, it is clear that $\mathbf{L}+\mathbf{\Delta}$ is PSD:

$$\begin{aligned}\mathbf{L}+\mathbf{\Delta}&=\mathbf{V}\mathrm{diag}\,(\lambda_1-\lambda_1,\ldots,\lambda_p-\lambda_p,\lambda_{p+1},\ldots,\lambda_n)\mathbf{V}^T\\&=\mathbf{V}\mathrm{diag}\,(0,\ldots,0,\lambda_{p+1},\ldots,\lambda_n)\mathbf{V}^T\end{aligned}$$

Since all the negative eigenvalues of $\mathbf{L}$ have been eliminated, $\mathbf{L}+\mathbf{\Delta}$ is PSD.

Second, due to the definition (21) of $\mathbf{\Delta}$, $\mathbf{L}+\mathbf{\Delta}$ can be spectrally decomposed using the *same* eigenvectors $\mathbf{V}$ as original $\mathbf{L}$. As discussed previously and also argued in [13] for spectral clustering, maintaining the same eigen-space in the perturbed matrix $\mathbf{L}+\mathbf{\Delta}$—especially the low graph frequency components—is desirable.

Third, by eliminating all negative eigenvalues of $\mathbf{L}$ to 0, the first $p+1$ eigenvectors $\mathbf{v}_1,\ldots,\mathbf{v}_p$ will all evaluate to 0 in the quadratic regularizer:

$$\mathbf{v}_i^T(\mathbf{L}+\mathbf{\Delta})\mathbf{v}_i=0,\ \ 1\leq i\leq p+1 \tag{23}$$

Eigenvalue 0 has multiplicity $p+1$ because $\mathbf{L}$ contains the DC component $\mathbf{1}$ that the regularizer also evaluates to 0. Hence the regularizer expresses no preference among the first $p+1$ eigenvectors. This is problematic during graph signal restoration. This means that though the graph structure $\mathcal{G}$ has a notion of frequencies and the original (numerically unstable) smoothness prior prefers low frequencies, the augmented regularizer does not differentiate and maps the lowest $p+1$ frequencies all to zero. This is not desirable.

B. Matrix Perturbation: Eigen-structure Preservation

The main problem with the minimum-norm criterion is that the differentiation among different frequency components (eigenvectors) is removed by setting all negative eigenvalues of $\mathbf{L}$ to 0. Thus, it is desirable to perturb $\mathbf{L}$ minimally, subject to the constraint that the frequency components *and* their frequency preferences (i.e., low frequencies are still preferred over high frequencies) are preserved.

Suppose first that we want to preserve the entire eigen-structure of $\mathbf{L}$: eigenvectors of $\mathbf{L}$ *and* spacings between neighboring eigenvalues of $\mathbf{L}$ during perturbation.[5] In other words, for perturbed $\mathbf{L}+\mathbf{\Delta}$, we require

$$\lambda_{i+1}-\lambda_i=\nu_{i+1}-\nu_i. \tag{24}$$

One method of achieving this, leveraging on the Weyl's inequality [50], is to select perturbation matrix $\mathbf{\Delta}=\eta\,\mathbf{I}$, for some $\eta>0$, so that $\mathbf{\Delta}$ has eigenvalue η with multiplicity N. Clearly eigenvectors of $\mathbf{L}$ are preserved in $\mathbf{L}+\mathbf{\Delta}=\mathbf{V}(\mathbf{L}+\eta\mathbf{I})\mathbf{V}^T$, and eigenvalue spacings are also preserved: $\nu_i=\lambda_i+\eta$ and $\nu_{i+1}-\nu_i=\lambda_{i+1}-\lambda_i$.

Given $\mathbf{\Delta}=\eta\,\mathbf{I}$, one interpretation of the smoothness prior $\mathbf{x}^T(\mathbf{L}+\mathbf{\Delta})\mathbf{x}$ is that it is a weighted sum of signal variations and signal energies:

$$\begin{aligned}\mathbf{x}^T(\mathbf{L}+\mathbf{\Delta})\mathbf{x}&=\mathbf{x}^T\mathbf{L}\mathbf{x}+\eta\,\mathbf{x}^T\mathbf{I}\mathbf{x}\\&=\sum_{i,j}w_{i,j}(x_i-x_j)^2+\eta\sum_i x_i^2\end{aligned} \tag{25}$$

In other words, to make smoothness prior $\mathbf{x}^T\mathbf{L}\mathbf{x}$ numerically stable, we consider in addition a weighted signal energy term to avoid pathological solutions like $-\infty\mathbf{v}_1$ for $\lambda_1<0$.

To find the perturbation matrix $\mathbf{\Delta}=\eta\,\mathbf{I}$ with the minimum norm $\|\mathbf{\Delta}\|$ such that $\mathbf{L}+\mathbf{\Delta}$ is PSD, we only need to identify the smallest eigenvalue $\lambda_{\min}=\lambda_1<0$ of $\mathbf{L}$ and set $\eta=-\lambda_{\min}$. Computing $\lambda_{\min}$ directly can be computationally expensive, however; we discuss faster methods to compute lower bounds for $\lambda_{\min}$ next.

[5]Though one can presumably preserve order of eigenvectors without preserving the same spacing in eigenvalues, by using $\eta\mathbf{I}$ as perturbation matrix $\mathbf{\Delta}$ we can achieve PSD without performing expensive eigen-decomposition.

C. A Simple Lower Bound for $\lambda_{\min}$

We can compute a lower bound for $\lambda_{\min}$ simply as follows. Denote by $\mathbf{L}^+$ and $\mathbf{L}^-$ the graph Laplacian matrices corresponding to edges with positive and negative weights in graph $\mathcal{G}$ respectively; clearly $\mathbf{L} = \mathbf{L}^+ + \mathbf{L}^-$. The Rayleigh quotient for $\mathbf{L}$ can be expanded as:

$$\frac{\mathbf{x}^T \mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\mathbf{x}^T\left(\mathbf{L}^+ + \mathbf{L}^-\right)\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{26}$$

Because $\mathbf{L}^+$ containing only positive edges is PSD, the first term in the numerator $\mathbf{x}^T\mathbf{L}^+\mathbf{x}$ is lower-bounded by 0. For the second term, we can first define $\mathcal{L}^- = -\mathbf{L}^-$, which is PSD, and write:

$$\frac{\mathbf{x}^T\mathbf{L}^-\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = -\left(\frac{\mathbf{x}^T\mathcal{L}^-\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right) \geq -\lambda^-_{\max} \tag{27}$$

where $\lambda^-_{\max}$ is the largest eigenvalue of $\mathcal{L}^-$. Since $-\lambda^-_{\max}$ is also the lower bound of the Rayleigh quotient for $\mathbf{L}$, it is also the lower bound for the smallest eigenvalue $\lambda_{\min}$ of $\mathbf{L}$:

$$-\lambda^-_{\max} \leq \lambda_{\min} \tag{28}$$

Thus a perturbation matrix $\mathbf{\Delta} = \lambda^-_{\max}\mathbf{I}$ would result in $\mathbf{L} + \mathbf{\Delta}$ that is PSD. $\lambda^-_{\max}$ for matrix $\mathcal{L}^-$ can be computed using the *power iteration method*, which has complexity $O(N)$ for a sparse graph per iteration [52]. However, convergence speed depends on the distance between $\lambda^-_{\max}$ and the next largest eigenvalue λ^-_{N-1} in $\mathcal{L}^-$; i.e., smaller $|\lambda^-_{\max} - \lambda^-_{N-1}|$ means a slower convergence rate. Further, this lower bound (28) is often loose in practice. We next discuss a faster and more robust computation of a lower bound for $\lambda_{\min}$.

VII. Fast Computation

Our goal is to obtain a lower bound $\lambda^{\#}_{\min}$ for $\lambda_{\min}$ robustly and efficiently. Having obtain $\lambda^{\#}_{\min}$, we can add perturbation matrix $\mathbf{\Delta}^{\#} = -\lambda^{\#}_{\min}\mathbf{I}$ to $\mathbf{L}$, so that the resulting $\mathbf{L} + \mathbf{\Delta}^{\#}$ is PSD. State-of-the-art eigenvalue methods include Lanczos method and its variants [52], Jacobi-Davidson [53] and Chebyshev-Davidson [54]; these methods find the extremal eigenvalues (eigenvalues with the largest or smallest magnitude) or eigenvalues in the vicinity of a pre-determined shift, which requires prior knowledge about the range of the target eigenvalue one is seeking. In our proposal, no such prior knowledge is required.

A. Matrix Inertia

We first define matrix inertia. The *inertia* $\mathrm{In}(\mathbf{A})$ of a matrix $\mathbf{A}$ is a set of three numbers counting the positive, negative, and zero eigenvalues in $\mathbf{A}$:

$$\mathrm{In}(\mathbf{A}) = \left(i^+(\mathbf{A}),\, i^-(\mathbf{A}),\, i^0(\mathbf{A})\right) \tag{29}$$

where $i^+(\mathbf{A})$, $i^-(\mathbf{A})$ and $i^0(\mathbf{A})$ denote respectively the number of positive, negative and zero eigenvalues in matrix $\mathbf{A}$. Inertia is an intrinsic property of the matrix; according to *Sylvester's Law of Inertia*, the inertia of a matrix is invariant to any congruent transform, i.e.,

$$\mathrm{In}(\mathbf{A}) = \mathrm{In}(\mathbf{P}^T\mathbf{A}\mathbf{P}) \tag{30}$$

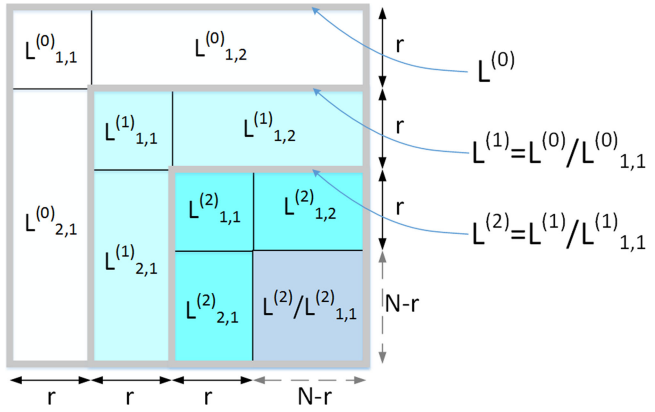


Fig. 5. Example of recursively partitioning Schur complement $\mathbf{L}^t/\mathbf{L}^t_{1,1}$ into two set of nodes: $\mathbf{L}^1 = \mathbf{L}^0/\mathbf{L}^0_{1,1}$ into $\mathbf{L}^1_{1,1}$ and $\mathbf{L}^1/\mathbf{L}^1_{1,1}$, then $\mathbf{L}^2 = \mathbf{L}^1/\mathbf{L}^1_{1,1}$ into $\mathbf{L}^2_{1,1}$ and $\mathbf{L}^2/\mathbf{L}^2_{1,1}$.

where $\mathbf{P}$ is an invertible matrix.

B. Graph Partition

To reduce complexity, we can divide the node set $\mathcal{N}$ into two subsets $\mathcal{N}_1$ and $\mathcal{N}_2$, so that intensive computation is performed in the node subsets separately. Note that partitioning a graph into two node sets to reduce complexity is also done in *Kron reduction* [55]. However, [55] considers only PSD $\mathbf{L}$ (possibly with self-loops), while we consider indefinite $\mathbf{L}$ that requires perturbation $\mathbf{\Delta}$ to make $\mathbf{L} + \mathbf{\Delta}$ PSD.

Given the two sets $\mathcal{N}_1$ and $\mathcal{N}_2$, we can write the graph Laplacian $\mathbf{L}$ in blocks:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{1,1} & \mathbf{L}_{1,2} \\ \mathbf{L}^T_{1,2} & \mathbf{L}_{2,2} \end{bmatrix} \tag{31}$$

where $\mathbf{L}_{1,1}$ and $\mathbf{L}_{2,2}$ are sub-matrices of respective dimension $|\mathcal{N}_1| \times |\mathcal{N}_1|$ and $|\mathcal{N}_2| \times |\mathcal{N}_2|$ corresponding to node sets $\mathcal{N}_1$ and $\mathcal{N}_2$, and $\mathbf{L}_{1,2}$ is a $|\mathcal{N}_1| \times |\mathcal{N}_2|$ sub-matrix corresponding to cross-connections between $\mathcal{N}_1$ and $\mathcal{N}_2$.

We can now relate the inertia of $\mathbf{L}$ with its sub-matrices using the *Haynsworth Inertia Additivity* formula [23]:

$$\mathrm{In}(\mathbf{L}) = \mathrm{In}(\mathbf{L}_{1,1}) + \mathrm{In}(\mathbf{L}/\mathbf{L}_{1,1}) \tag{32}$$

where $\mathbf{L}/\mathbf{L}_{1,1}$ is the *Schur Complement* (SC) of block $\mathbf{L}_{1,1}$ of matrix $\mathbf{L}$ in (31), which is defined as

$$\mathbf{L}/\mathbf{L}_{1,1} = \mathbf{L}_{2,2} - \mathbf{L}^T_{1,2}\mathbf{L}^{-1}_{1,1}\mathbf{L}_{1,2} \tag{33}$$

Thus, if we can ensure that $\mathbf{L}_{1,1}$ *and* its SC do not contain negative eigenvalues, then $\mathbf{L}$ will also have no negative eigenvalues and is PSD. We develop an efficient algorithm based on this idea next.

C. Eigenvalue Lower Bound Algorithm

We propose the following recursive algorithm to find a lower bound $\lambda^{\#}_{\min}$ for $\mathbf{L}$. See Fig. 5 for an illustration. We initialize $t := 0$ and $\mathbf{L}^0 := \mathbf{L}$. We define a recursive algorithm $EvalBound\left(\mathbf{L}^t, t\right)$ that returns a lower bound $\lambda^t_{\min}$ for eigenvalues in $\mathbf{L}^t$. It has two steps as described below.

Step 1: We first partition node set $\mathcal{N}^t$ in $\mathbf{L}^t$ into two subsets $\mathcal{N}_1^t$ and $\mathcal{N}_2^t$, where $|\mathcal{N}_1^t| = r$. r is a pre-defined parameter to control computation complexity. $\mathcal{N}_1^t$ can be chosen by first randomly selecting a node in $\mathcal{N}^t$, then perform *breadth-first search* (BFS) [56] until r nodes are discovered. We eigen-decompose $\mathbf{L}_{1,1}^t$ to find its smallest eigenvalue λ_1^t. We define the *augmented eigenvalue* $\kappa_{\min}^t$ as:

$$\kappa_{\min}^t = \begin{cases} \lambda_1^t - \epsilon & \text{if } \lambda_1^t \leq 0 \\ 0 & \text{o.w.} \end{cases} \tag{34}$$

where $\epsilon > 0$ is a small parameter. We perturb matrix $\mathbf{L}^t$ using computed $\kappa_{\min}^t$, i.e., $\mathcal{L}^t = \mathbf{L}^t - \kappa_{\min}^t \mathbf{I}$. It is clear that $\mathcal{L}_{1,1}^t$ is positive definite (PD) and thus invertible.

Step 2: We ensure SC of $\mathcal{L}_{1,1}^t$ of $\mathcal{L}^t$ is PSD. By definition, the SC is:

$$\mathcal{L}^t / \mathcal{L}_{1,1}^t = \mathcal{L}_{2,2}^t - \left(\mathbf{L}_{1,2}^t\right)^T \left(\mathcal{L}_{1,1}^t\right)^{-1} \mathbf{L}_{1,2}^t \tag{35}$$

$\mathcal{L}^t / \mathcal{L}_{1,1}^t$ can be interpreted as a $|\mathcal{N}_2^t| \times |\mathcal{N}_2^t|$ graph Laplacian matrix for nodes $\mathcal{N}_2^t$. If $|\mathcal{N}_2^t| \leq r$, then we eigen-decompose $\mathcal{L}^t / \mathcal{L}_{1,1}^t$ and find its smallest eigenvalue λ_2^t. We compute $\lambda_{\min}^t := \kappa_{\min}^t + \min(\lambda_2^t, 0)$. We exit the algorithm with $\lambda_{\min}^t$ as solution.

If $|\mathcal{N}_2^t| > r$, we set $\mathbf{L}^{t+1} := \mathcal{L}^t / \mathcal{L}_{1,1}^t$ and recursively call $\eta_{\min}^t := EvalBound(\mathbf{L}^{t+1}, t+1)$. Upon return, we compute $\lambda_{\min}^t := \kappa_{\min}^t + \eta_{\min}^t$ and exit the algorithm with $\lambda_{\min}^t$ as solution.

D. Proof of Algorithm Correctness

We now prove that $EvalBound(\mathbf{L}, 0)$ returns a lower bound for the true minimum eigenvalue $\lambda_{\min}$ of $\mathbf{L}$. Specifically, we prove by induction the following recursion invariant: *At each recursive call t, given $\mathbf{L}^t$, the computed $\lambda_{\min}^t$ is a lower bound for eigenvalues of $\mathbf{L}^t$.*

We first examine the base case. At a leaf recursive call τ, in Step 1, $\mathbf{L}^\tau$ is perturbed using computed $\kappa_{\min}^\tau$ such that $\mathcal{L}_{1,1}^\tau = \mathbf{L}_{1,1}^\tau - \kappa_{\min}^\tau \mathbf{I}$ is PD. In Step 2, if computed $\lambda_2^\tau \geq 0$, then SC $\mathcal{L}^\tau / \mathcal{L}_{1,1}^\tau$ is PSD. Since $\mathcal{L}_{1,1}^\tau$ and its SC $\mathcal{L}^\tau / \mathcal{L}_{1,1}^\tau$ are both PSD, by (32) $\mathcal{L}^\tau$ is also PSD. Hence $\lambda_{\min}^\tau = \kappa_{\min}^\tau$ is a lower-bound for matrix $\mathbf{L}^\tau$.

If $\lambda_2^\tau < 0$, then perturbed SC, $\mathcal{L}^\tau / \mathcal{L}_{1,1}^\tau - \lambda_2^\tau \mathbf{I}$, is PSD. It turns out that if we perturb $\mathcal{L}^\tau$ again using λ_2^τ, i.e., $\mathcal{L}'^\tau = \mathcal{L}^\tau - \lambda_2^\tau \mathbf{I}$, then $\mathcal{L}'^\tau$ is PSD. This is because $\mathcal{L}'^\tau_{1,1} = \mathcal{L}_{1,1}^\tau - \lambda_2^\tau \mathbf{I}$ is PD, and its SC $\mathcal{L}'^\tau / \mathcal{L}'^\tau_{1,1}$ is PSD by the following lemma:

Lemma 1: If $\mathbf{L}_{1,1}$ is PD and $\mathbf{L}/\mathbf{L}_{1,1} + \delta \mathbf{I}$ is PSD for $\delta > 0$, then SC $\mathbf{L}'/\mathbf{L}'_{1,1}$, where $\mathbf{L}' = \mathbf{L} + \delta \mathbf{I}$, is also PSD.

See Appendix for a full proof. In this case $\lambda_{\min}^\tau = \kappa_{\min}^\tau + \lambda_2^\tau$, hence $\lambda_{\min}^\tau$ is a lower bound for $\mathbf{L}^\tau$.

Consider now the inductive case, where at iteration t we assume that, $\eta_{\min}^t := EvalBound(\mathbf{L}^{t+1}, t+1)$ is a lower bound for $\mathbf{L}^{t+1} = \mathcal{L}^t / \mathcal{L}_{1,1}^t$. From Step 1, we know that $\mathbf{L}^t$ is perturbed using $\kappa_{\min}^t$ so that $\mathcal{L}_{1,1}^t = \mathbf{L}_{1,1}^t - \kappa_{\min}^t \mathbf{I}$ is PD. By assumption, we know that $\mathcal{L}^t / \mathcal{L}_{1,1}^t$ can be perturbed using $\eta_{\min}^t$ such that $\mathcal{L}^t / \mathcal{L}_{1,1}^t - \eta_{\min}^t \mathbf{I}$ is PSD. By Lemma 1, we know that $\mathcal{L}^t - \eta_{\min}^t \mathbf{I}$ is also PSD. Thus $\lambda_{\min}^t := \kappa_{\min}^t + \eta_{\min}^t$ is a lower bound for $\mathbf{L}^t$.

Since both the base case and the inductive case are proven, the recursion invariant is also proven, and $EvalBound(\mathbf{L}, 0)$ returns a lower bound for $\lambda_{\min}$ of $\mathbf{L}$. □

E. Computation Complexity

We can estimate the computation cost of our algorithm as follows. For each recursive call, the cost of eigen-decomposing a $r \times r$ matrix is $O(r^3)$ operations. The number of recursive calls is $O(N/r)$. Thus the complexity of step 1 of our algorithm is $O((N/r)r^3) = O(Nr^2)$.

The cost of computing SC in (35) can be bounded as follows. In the extreme case when $r = 1$, the off-diagonal blocks $\mathbf{L}_{1,2}^t$ are vectors, and thus computing (35) throughout the algorithm means that each off-diagonal entry in $\mathbf{L}$ is accessed exactly once, resulting in $O(N^2)$. However, if we assume the original Laplacian $\mathbf{L}$ is sparse, then only $O(N)$ entries in $\mathbf{L}$ are non-zero, reducing the complexity to $O(N)$. When $r > 1$, each entry in $r \times r$ matrix $(\mathcal{L}_{1,1}^t)^{-1}$ will access an entry in block $\mathbf{L}_{1,2}^t$, resulting in complexity $O(Nr^2)$. Thus the overall complexity is $O(Nr^2)$. Compared to the complexity $O(N^3)$ of eigen-decomposition of the larger matrix $\mathbf{L}$, this represents a non-trivial computation saving.

VIII. Algorithm Development

Having discussed a fast method to compute $\mathbf{\Delta}$ such that $\mathbf{L} + \mathbf{\Delta}$ is PSD, we now discuss how to optimize (16). The same algorithm can be used to solve (9) with one fewer prior.

A. Iterative Reweighted Least Squares Algorithm

To solve (16), we employ the following optimization strategy. We first replace the l_0-norm in (16) with a weighted l_2-norm:

$$\min_{\mathbf{x}} (\mathbf{y} - \mathbf{H}\mathbf{x})^T \mathbf{B} (\mathbf{y} - \mathbf{H}\mathbf{x}) + \mu_1 \mathbf{x}^T \mathbf{L}_g \mathbf{x} + \mu_2 \mathbf{x}^T (\mathbf{L}^+)^2 \mathbf{x} \tag{36}$$

where $\mathbf{B}$ is a $K \times K$ diagonal matrix with weights $b_1, \ldots, b_K$ on its diagonal. In other words, the fidelity term is now a weighted sum of label differences: $(\mathbf{y} - \mathbf{H}\mathbf{x})^T \mathbf{B} (\mathbf{y} - \mathbf{H}\mathbf{x}) = \sum_{i=1}^K b_i (y_i - \mathbf{H}_{i,:}\mathbf{x})^2$.

The weights b_i should be set so that the weighted l_2-norm mimics the l_0-norm. To accomplish this, we employ the *iterative reweighted least squares* (IRLS) strategy [25], which has been proven to have superlinear local convergence, and solve (36) iteratively, where the weights $b_i^{(t+1)}$ of iteration $t+1$ is computed using solution $x_i^{(t)}$ of the previous iteration t, i.e.,

$$b_i^{(t+1)} = \frac{1}{(y_i - \mathbf{H}_{i,:}\mathbf{x}^{(t)})^2 + \epsilon} \tag{37}$$

for a small $\epsilon > 0$ to maintain numerical stability. Using this weight update, we see that the weighted quadratic term $(\mathbf{y} - \mathbf{H}\mathbf{x})^T \mathbf{B} (\mathbf{y} - \mathbf{H}\mathbf{x})$ mimics the original l_0-norm $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_0$ in the original objective (16) when the solution $\mathbf{x}$ converges.

1) Linear System per Iteration: For a given weight matrix $\mathbf{B}$, it is clear that the objective (36) is an unconstrained quadratic programming problem with three quadratic terms. One can thus

take the derivative with respect to $\mathbf{x}$ and equate it to zero, resulting in:

$$\left(\mathbf{H}^T\mathbf{B}\mathbf{H} + \mu_1\mathbf{L}_g + \mu_2(\mathbf{L}^+)^2\right)\mathbf{x}^* = \mathbf{H}^T\mathbf{B}^T\mathbf{y} \tag{38}$$

(38) is a linear system of equations, where the matrix on the left is sparse, symmetric and positive definite. Thus it can be solved by fast methods like conjugate gradient instead of matrix inversion.

B. Interpreting Computed Solution $\mathbf{x}^*$

After the IRLS algorithm converges to a solution $\mathbf{x}^*$, we interpret the classification results as follows. If (9) with one graph signal smoothness prior is used as objective, then a simple thresholding at 0 is sufficed to estimate binary label x_i:

$$x_i = \text{sign}\,(x_i^*) \tag{39}$$

On the other hand, if (16) with the generalized smoothness prior is used in addition, then we perform thresholding by a pre-defined value τ on $\mathbf{x}^*$ to divide it into three parts, including the rejection option for ambiguous labels:

$$x_i = \begin{cases} 1, & x_i^* > \tau \\ \text{Rejection}, & -\tau < x_i^* < \tau \\ -1, & x_i^* < -\tau. \end{cases} \tag{40}$$

Typically, the fraction of tolerable rejection labels is set per application requirement. Clearly, eliminating more ambiguous labels leads to a larger tolerable rejection rate, resulting in a smaller classification error rate. Note that the generalized smoothness prior induces a slope that reflects the confidence level of the classified signal samples. If we perform simple hard thresholding as done in (39), then the benefit of differentiating between more and less confidently predicted samples will simply disappear.

IX. Experimentation

A. Experiment Setup

1) Datasets for Training and Testing: To evaluate the performances of different classification methods, we selected four two-class datasets from the KEEL (Knowledge Extraction based on Evolutionary Learning) database [57], which contains a rich collection of labeled and unlabeled datasets for data mining and analysis and face gender dataset provided in [58].

The first dataset is the Phoneme dataset that provides values of five categorical attributes to distinguish nasal sounds (class 0) from oral sounds (class 1). The second is the Banana dataset, an artificial dataset where 5300 instances belong to several clusters with a banana shape. In the dataset, two attributes were extracted to classify two kinds of banana shapes. The third is the Face Gender dataset that consists of 7900 face images (395 individuals, 20 images per individual). We extract the Local Binary Pattern (LBP) features to represent the faces for classifying the genders of faces. The fourth dataset called "Sonar, Mines vs. Rocks" contains various patterns obtained by bouncing sonar signals off metal cylinders and rocks at various angles and under various conditions. Each pattern is a set of 60 numbers in the range $[0.0, 1.0]$, where each number represents the energy within a particular frequency band, integrated over a certain period of time. These patterns are used to classify an object to a metal cylinder or a rock.

For our experiments, we randomly sampled 300 instances from the first and second dataset, 400 instances from the third and 210 instances from the fourth, and used 70% of the samples as training data and 30% as testing data. We repeated the process 100 times for each dataset and then calculated the average performance of the 100 trials in terms of classification error rate.

2) Graph Construction: To construct a graph for our proposed methods, we first constructed an initial graph with positive edge weights. For each sample (node), we found its three nearest neighbors according to the Euclidean distances between the node and its neighbors, and connected these nodes using edges with positive weights that are normalized to [0, 1] using the Gaussian kernel in (17). We performed clustering using only the labeled nodes in the graph (because they are more reliable) and found the centroids and boundaries of the two clusters. The combined graph Laplacian matrix $\mathbf{L}_g$ is computed using parameter β in (18), which decreases with iteration. Specifically, β decreases for each solved solution in (38). We then assigned a negative edge weight between each pair with a value normalized to $[-10, 0]$, $[-20, 0]$, $[-1, 0]$ and $[-1, 0]$ for the four datasets respectively, where the magnitude is proportional to the Euclidean distance between the pair. For boundaries, we paired the cluster boundaries based on feature distances to find the boundary samples of the two clusters and assigned a negative edge weight between each pair with a value normalized to $[-1, 0]$, $[-0.1, 0]$, $[-1, 0]$ and $[-0.1, 0]$ for the four datasets respectively, where the magnitude is proportional to the Euclidean distance between the pair.

3) Comparison Schemes: We tested our proposed algorithm against eight schemes: i) linear SVM, ii) SVM with a RBF kernel (named SVM-RBF), iii) a more robust version of the famous AdaBoost called RobustBoost [26] that claims robustness against label noise, iv) a graph classifier with the graph signal smoothness prior (2) where the edge weights of the graph are all positive (named Graph-Pos), v) a graph classifier with a graph containing negative edge weights where the graph Laplacian $\mathbf{L}$ is perturbed by the minimum-norm perturbation criteria in (20) to eliminate negative eigenvalues for numerical stability (named Graph-MinNorm), vi) a bandlimited graph method proposed in [17] (named Graph-Bandlimited), vii) a graph classifier using a smoothness prior based on the adjacency matrix proposed in [9] (named Graph-AdjSmooth), and viii) a semi-supervised learning algorithm based on graph wavelet [6] (named Graph-Wavelet).

We implemented four variants of our proposed minimum-variance perturbation graph classifier. The first three utilize the generalized graph Laplacian $\mathbf{L}_g$ without the generalized smoothness term (i.e., $\mu_2 = 0$ in (16) and $\tau = 0$ in (40)) based on three different negative edge weights assignment schemes as described in Section V-A: i) assigning negative edge weights between the centroid sample pairs (named Proposed-Centroid); ii) assigning negative edge weights between boundary sample pairs (named Proposed-Boundary); and iii) assigning negative

TABLE I
CLASSIFICATION ERROR RATES IN THE PHONEME DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	21.83%	23.35%	24.55%	25.05%	25.64%
SVM-RBF	16.63%	16.84%	17.48%	17.72%	19.34%
RobustBoost [26]	12.81%	14.91%	17.94%	19.33%	21,50%
Graph-Pos	13.22%	14.91%	16.79%	18.17%	20.70%
Graph-MinNorm	11.97%	13.69%	15.94%	17.38%	19.82%
Graph-Bandlimited [17]	11.70%	14.06%	17.05%	18.70%	21.29%
Graph-AdjSmooth [9]	11.31%	13.69%	16.79%	18.65%	20.67%
Graph-Wavelet [6]	27.25%	28.84%	30.48%	31.95%	33.51%
Proposed-Centroid	10.81%	**13.09%**	16.18%	17.87%	20.47%
Proposed-Boundary	12.14%	14.44%	17.18%	19.02%	21.51%
Proposed-Hybrid	**10.62%**	13.31%	**16.01%**	**17.86%**	**19.77%**
Proposed-Rej	9.95% (9.85%)	12.04% (9.92%)	14.44% (9.65%)	15.37% (10.00%)	17.52% (9.23%)

TABLE II
CLASSIFICATION ERROR RATES IN THE BANANA DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	54.71%	54.97%	54.70%	53.95%	53.42%
SVM-RBF	12.49%	13.27%	13.72%	16.23%	18.63%
RobustBoost [26]	20.42%	22.73%	24.53%	25.12%	27.52%
Graph-Pos	14.05%	15.89%	18.02%	20.76%	21.93%
Graph-MinNorm	10.23%	12.37%	14.44%	17.41%	18.69%
Graph-Bandlimited [17]	7.53%	11.77%	15.80%	19.14%	21.07%
Graph-AdjSmooth [9]	8.85%	12.08%	15.28%	18.26%	20.67%
Graph-Wavelet [6]	23.18%	24.25%	25.70%	27.15%	30.13%
Proposed-Centroid	**5.02%**	10.40%	13.99%	16.83%	19.56%
Proposed-Boundary	13.73%	16.92%	20.00%	21.82%	23.90%
Proposed-Hybrid	5.36%	**9.43%**	**12.83%**	**16.11%**	**18.46%**
Proposed-Rej	3.83% (9.40%)	6.58% (9.87%)	9.28% (9.05%)	12.21% (9.94%)	14.09% (9.93%)

edge weights between the centroid sample pairs and between the boundary sample pairs (named Proposed-Hybrid). The fourth variant is the proposed method in (16) with rejection (named Proposed-Rej) where the rejection rate is controlled to be within 9–10% by tuning parameters in (40).

B. Performance Evaluation

To evaluate the robustness of different classification schemes against label noise, we randomly selected a portion of samples from the training set and reversed their labels. All the classifiers were then trained using the same set of features and labels. Each test set was classified by the classifiers and the results are compared with the ground-truth labels.

1) Numerical Comparisons for Different Label Noise: The resulting classification error rates for the first three datasets using different classifiers are presented in Tables I–III, where the percentage of randomly erred training labels ranges from 0% to 20%. The comparisons show that our proposed scheme achieves the lowest classification error rates when compared to the competing schemes under almost all training label error rates. The parameters (μ_1, μ_2, τ) used for the three datasets respectively are: $(1, 2, [0.012, 0.029])$, $(0.1, 1, [0.000055, 0.00035])$, and $(0.1, 1, [0.012, 0.0305])$. Parameter ϵ is fixed to 0.0001 in our experiments, which is a small value to maintain numerical stability when $\left(y_i - \mathbf{H}_{i,:}\mathbf{x}^{(t)}\right)^2$ is close to 0 in (37). Parameter τ is adjusted per dataset, so that the resulting rejection rate is close to 10%. Compared to the graph classifiers Graph-Pos and Graph-AdjSmooth, our results show that adding negative edge weights can effectively improve the classification accuracy by 1.03–8.5% and 0.54–7.68%, respectively. We can also observe that our proposed matrix perturbation scheme significantly outperforms the minimum-norm based perturbation (Graph-MinNorm) in classification accuracy. Compared to Graph-Bandlimited and Graph-Wavelet, the proposed hybrid method improves the classification accuracy by 1.06–15.73% and 11.11–19.21%, respectively.

TABLE III
CLASSIFICATION ERROR RATES IN THE FACE GENDER DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
SVM-Linear	17.65%	18.22%	18.77%	19.59%	21.6%
SVM-RBF	12.14%	12.16%	12.83%	16.30%	24.01%
RobustBoost [26]	9.15%	11.09%	14.36%	17.36%	20.68%
Graph-Pos	13.15%	13.62%	14.38%	15.39%	16.54%
Graph-MinNorm	7.15%	8.26%	9.48%	10.37%	12.01%
Graph-Bandlimited [17]	5.78%	11.83%	15.30%	19.74%	23.44%
Graph-AdjSmooth [9]	1.25%	5.01%	7.94%	11.45%	15.39%
Graph-Wavelet [6]	20.02%	19.95%	20.12%	20.7%	21.43%
Proposed-Centroid	**0.96%**	3.11%	5.65%	7.98%	11.11%
Proposed-Boundary	10.81%	12.09%	13.17%	14.33%	15.96%
Proposed-Hybrid	1.41%	**3.02%**	**4.57%**	**7.37%**	**8.94%**
Proposed-Rej	0.61% (9.29%)	1.56% (9.26%)	2.73% (9.79%)	4.54% (9.53%)	6.47% (9.61%)

TABLE IV
AVERAGE CLASSIFICATION ERROR RATES IN THE SONAR, MINES VS. ROCKS DATASET FOR COMPETING SCHEMES UNDER DIFFERENT TRAINING LABEL ERROR RATES AND THREE DIFFERENT μ_1 WEIGHTS

% label noise	0%	5%	10%	15%	20%
Graph-Pos	22.62%	24.28%	26.07%	28.63%	30.27%
Graph-MinNorm	21.89%	23.53%	25.61%	28.14%	30.04%
Graph-AdjSmooth [9]	24.20%	25.56%	26.76%	27.91%	30.82%
Proposed-Hybrid	**19.42%**	**20.99%**	**22.36%**	**24.30%**	**25.93%**

Further, as shown in Table IV, we evaluate the performances of those four graph classifiers that employ a smoothness prior (i.e., Graph-Pos, Graph-AdjSmooth, Graph-MinNorm, and Proposed-Hybrid) using a range of weight parameter values μ_1. For the first three methods, we set μ_1 to be 10, 1, and 0.1, respectively, and then calculate the average classification error rate accordingly, whereas for Proposed-Hybrid, we set the value of μ_1 to be 0.1, 0.01, and 0.001, respectively. Table IV shows that the proposed method is not sensitive to the change of μ_1 and, compared to the other three graph classifiers, improves the classification accuracy by 2.47–4.89% for the "Sonar, Mines vs. Rocks" dataset.

2) Graph Classifier with Non-Zero Rejection Rate: By allowing a certain amount of ambiguous samples to remain unlabeled (less than 10% rejection rate in our experiments), our proposed generalized graph signal smoothness prior can further improve the classification accuracy. We note that a user may define the desired classifier performance as a weighted sum of classification error and rejection rate for different applications, as done in [59]. Table V shows the classification error and rejection rates in the "Banana" dataset for our proposed method with rejection under different training label error rates and μ_2, where the values of τ are set the same as that used in the first row. It

TABLE V
CLASSIFICATION ERROR AND REJECTION RATES IN THE BANANA DATASET FOR THE PROPOSED METHOD (WITH REJECTION) UNDER DIFFERENT TRAINING LABEL ERROR RATES AND μ_2 (THE NUMBERS IN THE PARENTHESES OF THE LAST ROW INDICATE THE REJECTION RATES)

% label noise	0%	5%	10%	15%	20%
Proposed-Rej ($\mu_2 = 1$)	3.83% (9.40%)	6.58% (9.87%)	9.28% (9.05%)	12.21% (9.94%)	14.09% (9.93%)
Proposed-Rej ($\mu_2 = 0.8$)	3.75% (9.56%)	6.58% (9.90%)	9.17% (9.20%)	12.07% (10.04%)	14.10% (9.85%)
Proposed-Rej ($\mu_2 = 0.6$)	3.72% (9.56%)	6.54% (9.74%)	9.09% (9.79%)	12.18% (9.82%)	14.18% (9.86%)

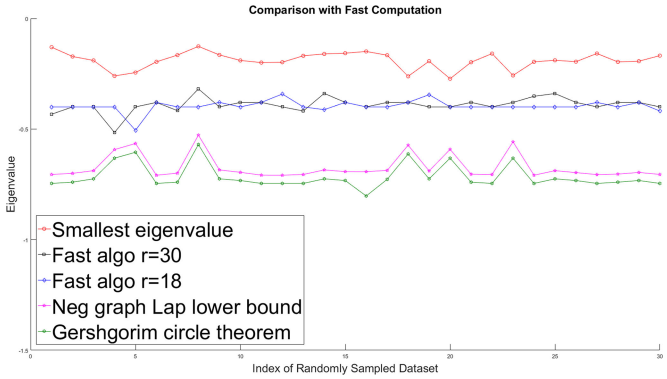


Fig. 6. Comparison of the actual smallest eigenvalues and their lower-bounds using $r = 30$ and 18 for the Phoneme dataset ($N = 300$), corresponding to 99% and 99.64% computation reduction.

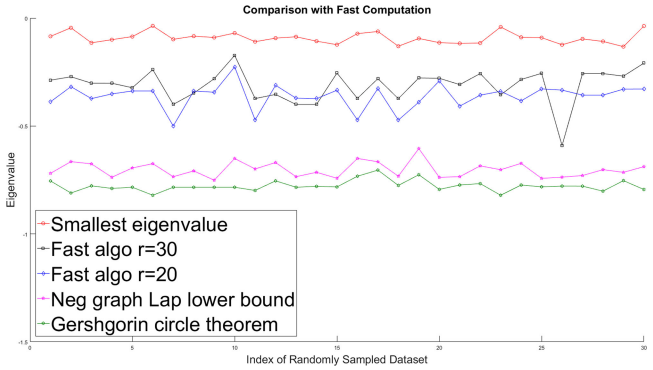


Fig. 7. Comparison of the actual smallest eigenvalues and their lower-bounds using $r = 30$ and 20 for the gender dataset ($N = 400$), corresponding to 99% and 99.75%, computation reduction, respectively.

shows that as μ_2 for the generalized smoothness term increases, the rejection rate also increases, which is consistent with our explanation in Section IV that the second smoothness term promotes ambiguity in the solution instead of forcing the solution to be strictly binary. As a result, using our algorithm, one can thus tune μ_2 and τ to adjust the preference of classification error versus rejection rate.

3) Fast Computation: In Section VII-C, we proposed a fast eigen-decomposition scheme to lower-bound the smallest eigenvalue $\lambda_{\min}$ of the graph Laplacian $\mathbf{L}$ for a graph with negative edge weights. Figs. 6 and 7 show the actual minimum eigenvalues (denoted Smallest eigenvalue), their computed lower-bounds (denoted Fast), negative graph Laplacian $\mathbf{L}^-$ lower bound $-\lambda^-_{\max}$ (28) and lower-bounds computed using the Gershgorin circle theorem [60] in the first 30 out of 100

TABLE VI
CLASSIFICATION ERROR RATES IN THE THREE DATASETS FOR THE PROPOSED MINIMUM-VARIANCE MATRIX PERTURBATION METHOD WITH ACTUAL MINIMUM AND LOWER-BOUND EIGENVALUES UNDER DIFFERENT TRAINING LABEL ERROR RATES

% label noise	0%	5%	10%	15%	20%
Phoneme dataset					
Proposed	10.62%	13.31%	16.01%	17.86%	19.77%
Fast ($r = 150$)	10.99%	13.27%	16.43%	17.73%	20.29%
Fast ($r = 100$)	10.74%	13.20%	16.15%	17.94%	20.83%
Banana dataset					
Proposed	5.36%	9.43%	12.83%	16.11%	18.46%
Fast ($r = 150$)	5.44%	9.40%	12.61%	15.97%	18.18%
Fast ($r = 100$)	5.31%	9.46%	12.80%	15.91%	18.45%

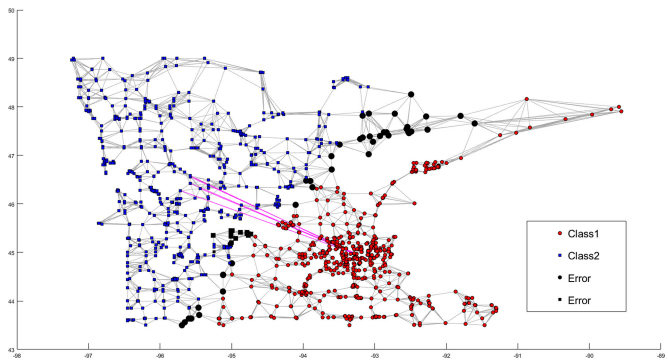


Fig. 8. Visualization classification result in graph based on first eigenvector result for Minnesota road network dataset, the purple line is negative edges.

sampled data subsets for two datasets, respectively. In the experiment, we set parameter r to be about $\sqrt{N}$ and 30 of total number of samples N, which correspond to about 99% and 99.64–99.75% computation reduction, respectively, since the computation complexity is reduced from $O(N^3)$ to $O(Nr^2)$ as explained in Section VII-E.

The results show that $\lambda^{\#}_{\min}$ computed by our proposed fast algorithm is an actual lower-bound for the true minimum eigenvalue $\lambda_{\min}$, i.e., $\lambda^{\#}_{\min} \leq \lambda_{\min}$. $\lambda^{\#}_{\min}$ is also a tighter lower bound than the two alternatives computed using negative graph Laplacian and the Gershgorin circle theorem. The proposed fast algorithm then obtains $\mathbf{L}_g$ using matrix perturbation $-\lambda^{\#}_{\min}\mathbf{I}$. Table VI compares the classification error rates of the proposed perturbation method (without rejection) using the actual minimum eigenvalues with the approximation computing the lower-bound eigenvalues by our proposed fast algorithm. Results show that the fast algorithm leads to slight performance differences compared to the full computation method.

4) Visualization Result: In Section V-B, we use a simple example to illustrate why by using negative edge weights, the resulting low graph frequency components of an indefinite graph Laplacian $\mathbf{L}$ can be useful in restoring signal $\mathbf{x}$. In this subsection, we use also the Minnesota road network dataset that provides 2642 x- and y- coordinates with road network data for illustration. To properly visualize the graph on a 2D plot, we randomly select 1400 nodes to construct a graph. We cluster the selected nodes into two groups via k-means, and assign a negative edge connecting the two cluster centers with a weight normalized to $[-1, 0]$ based on the Euclidean distance between the nodes. Fig. 8 shows the first eigenvector of the graph Laplacian with negative edges, which reflect labels of the two clusters.

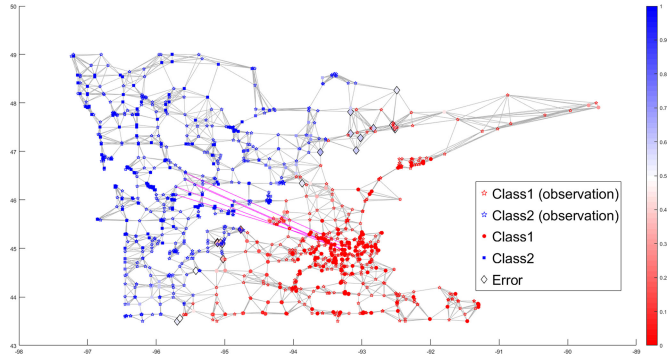


Fig. 9. Visualization reconstruction of x in graph for Minnesota road network dataset (the deeper the color is, the reconstruction of x is closer to 1 or -1), the purple line is negative edges.

Further, we show the restored signal $\mathbf{x}$ in Fig. 9—a reconstruction of the target cluster index signal—before thresholding to -1 and 1, where the deeper the color, the closer the reconstructed sample is to 1 or -1. This shows that the restored signal matches well with the first eigenvector and the true cluster index signal.

X. Conclusion

To address the semi-supervised learning problem, in this paper we view a classifier as a graph signal in a high-dimensional feature space, and pose a maximum a posteriori (MAP) problem to restore the classifier signal given partial and noisy labels. Unlike previous graph-based classifier works, we consider in addition edges with negative weights that signify dissimilarity between sample pairs. To achieve a stable signal smoothness prior, we derive a minimum-norm perturbation matrix $\mathbf{\Delta}$ that preserves the original eigen-structure, so that when added to the graph Laplacian $\mathbf{L}$, the matrix sum is positive semi-definite (PSD). We can compute a fast approximation to $\mathbf{\Delta}$ using a recursive algorithm based on the Haynsworth inertia additivity formula. Finally, we show that a generalized smoothness prior can promote ambiguity in the classifier signal, so that estimated labels with low confidence can be rejected. Experimental results show that our proposal outperforms SVM variants and previous graph-based classifiers using positive-edge graphs noticeably.

Appendix

We prove that if $\mathbf{L}_{1,1}$ is PD and $\mathbf{L}/\mathbf{L}_{1,1} + \delta\mathbf{L}$ is PSD for $\delta > 0$, then given perturbed matrix $\mathbf{L}' = \mathbf{L} + \delta\mathbf{I}$, $\mathbf{L}'/\mathbf{L}'_{1,1}$ is also PSD. By definition, $\mathbf{L}/\mathbf{L}_{1,1} + \delta\mathbf{I}$ is PSD means:

$$\mathbf{x}^T \left(\mathbf{L}_{2,2} - \mathbf{L}_{1,2}^T \mathbf{L}_{1,1}^{-1} \mathbf{L}_{1,2} + \delta\mathbf{I}\right) \mathbf{x} \geq 0$$

Let $\mathbf{L}_{1,1}$ be spectrally decomposed to $\mathbf{L}_{1,1} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix containing all positive eigenvalues, since $\mathbf{L}_{1,1}$ is PD. We can thus rewrite above:

$$\begin{aligned} &\mathbf{x}^T \mathbf{L}_{2,2} \mathbf{x} - \underbrace{\mathbf{x}^T \mathbf{L}_{1,2}^T \mathbf{V}}_{\mathbf{y}^T} \operatorname{diag}\left(\lambda_1^{-1}, \ldots, \lambda_n^{-1}\right) \underbrace{\mathbf{V}^T \mathbf{L}_{1,2} \mathbf{x}}_{\mathbf{y}} + \delta \mathbf{x}^T \mathbf{x} \\ &= \mathbf{x}^T \mathbf{L}_{2,2} \mathbf{x} + \delta \mathbf{x}^T \mathbf{x} - \sum_i \lambda_i^{-1} y_i^2 \end{aligned} \tag{41}$$

If $\mathbf{L}$ is now perturbed by $\delta\mathbf{I}$, we can similarly write the resulting SC $\mathbf{L}'/\mathbf{L}'_{1,1}$ in quadratic form:

$$\begin{aligned} &\mathbf{x}^T \left(\mathbf{L}_{2,2} + \delta\mathbf{I} - \mathbf{L}_{1,2}^T (\mathbf{L}_{1,1} + \delta\mathbf{I})^{-1} \mathbf{L}_{1,2}\right) \mathbf{x} \\ &= \mathbf{x}^T \mathbf{L}_{2,2} \mathbf{x} + \delta \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{L}_{1,2}^T (\mathbf{L}_{1,1} + \delta\mathbf{I})^{-1} \mathbf{L}_{1,2} \mathbf{x} \end{aligned} \tag{42}$$

The first two terms are the same as ones in (41). The third term can be rewritten as:

$$\begin{aligned} &\underbrace{\mathbf{x}^T \mathbf{L}_{1,2}^T \mathbf{V}}_{\mathbf{y}^T} \operatorname{diag}\left((\lambda_1 + \delta)^{-1}, \ldots, (\lambda_n + \delta)^{-1}\right) \underbrace{\mathbf{V}^T \mathbf{L}_{1,2} \mathbf{x}}_{\mathbf{y}} \\ &= \sum_i (\lambda_i + \delta)^{-1} y_i^2 \end{aligned} \tag{43}$$

Since $\lambda_i > 0$, we see that $\lambda_i^{-1} > (\lambda_i + \delta)^{-1}$. Hence this third term has magnitude strictly smaller than one in (41). Thus, non-negativity in (41) implies non-negativity in (42). Thus $\mathbf{L}'/\mathbf{L}'_{1,1}$ is PSD. □

Acknowledgment

The authors would like to thank V. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran for providing source code of the graph-wavelet-based semi-supervised learning algorithm. The authors would also like to thank Prof. M. Ng of Hong Kong Baptist University for the valuable discussion on an early draft of this paper.

References

[1] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[2] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.*, Whistler, BC, Canada, Dec. 2003, pp. 321–328.

[3] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Learning Theory*, vol. 3120, J. Shawe-Taylor and Y. Singer, Eds. Berlin, Germany: Springer, 2004, pp. 624–638.

[4] M. Gavish, B. Nadler, and R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 367–374.

[5] D. Shuman, M. Faraji, and P. Vandergheynst, "Semi-supervised learning with spectral graph wavelets," in *Proc. Int. Conf. Sampling Theory Appl.*, Singapore, May 2011.

[6] V. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran, "Wavelet-regularized graph semi-supervised learning," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 2013, pp. 423–426.

[7] A. Guillory and J. Bilmes, "Label selection on graphs," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 691–699.

[8] L. Zhang, C. Cheng, J. Bu, D. Cai, X. He, and T. Huang, "Active learning based on locally linear reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2026–2038, Oct. 2014.

[9] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.

[10] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2014, 492–501.

[11] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1996.

[12] 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.

[13] A. Knyazev, "Signed Laplacian for spectral clustering revisited," Jan. 2017. [Online]. Available: https://arxiv.org/abs/1701.01394

[14] Y. Mao, G. Cheung, C.-W. Lin, and Y. Ji, "Image classifier learning from noisy labels via generalized graph smoothness priors," in *Proc. IEEE 12th Image, Video, Multimedia Signal Process. Workshop*, Bordeaux, France, Jul. 2016, pp. 1–5.

[15] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process.*, Pula, Italy, Oct. 2013, pp. 1–6.

[16] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 2013, pp. 491–494.

[17] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation of graph structured data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 5445–5449.

[18] Y. Mao, G. Cheung, and Y. Ji, "On constructing z-dimensional DIBR-synthesized images," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1453–1468, Aug. 2016.

[19] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of AC signal," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2896–2909, Jun. 2016.

[20] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Inter-block soft decoding of JPEG images with sparsity and graph-signal smoothness priors," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, Sep. 2015, pp. 1628–1632.

[21] W. Hu, G. Cheung, and M. Kazui, "Graph-based dequantization of block-compressed piecewise smooth images," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 242–246, Feb. 2016.

[22] J. Pang and G. Cheung, "Graph Laplacian regularization for image denoising: Analysis inn the continuous domain," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1770–1785, Apr. 2017.

[23] E. V. Haynsworth and A. M. Ostrowski, "On the inertia of some classes of partitioned matrices," *Linear Algebra Appl.*, vol. 1, no. 2, pp. 299–316, 1968.

[24] K. Bredies and M. Holler, "A TGV-based framework for variational image decompression, zooming and reconstruction. Part I: Analytics," *SIAM J.*, vol. 8, no. 4, pp. 2814–2850, 2015.

[25] I. Daubechies, R. Devore, M. Fornasier, and S. Gunturk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.

[26] Y. Freund, "A more robust boosting algorithm," May 2009. [Online]. Available: https://arxiv.org/abs/0905.2138

[27] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph," in *Proc. 1st Workshop Unsupervised Learn. Natural Lang. Process.*, Edinburgh, U.K., Jul. 2011, pp. 53–63.

[28] Y. Wang and A. Pal, "Detecting emotions in social media: A constrained optimization approach," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, Jul. 2015, pp. 996–1002.

[29] W. Hu, G. Cheung, A. Ortega, and O. Au, "Multi-resolution graph Fourier transform for compression of piecewise smooth images," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 419–433, Jan. 2015.

[30] Y. Jin and D. Shuman, "An m-channel critically sampled filter bank for graph signal," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LA, USA, Mar. 2017, pp. 3909–3913.

[31] J. Zeng, G. Cheung, and A. Ortega, "Bipartite subgraph decomposition for critically sampled wavelet filterbanks on arbitrary graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 6210–6214.

[32] J. Zeng, G. Cheung, and A. Ortega, "Bipartite subgraph decomposition for graph wavelet signal decomposition," *IEEE Trans. Signal Process.*, vol. 26, no. 4, pp. 1770–1785, Jul. 2017.

[33] H. Shomorony and A. S. Avestimehr, "Sampling large data on graphs," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Atlanta, GA, USA, Dec. 2014, pp. 933–936.

[34] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[35] A. Knyazev and A. Malyshev, "Accelerated graph-based spectral polynomial filters," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Boston, MA, USA, Sep. 2015, pp. 1–6.

[36] A. Knyazev, "Edge-enhancing filters with negative weights," in *Proc. IEEE Int. Conf. Signal Inf. Process.*, Orlando, FL, USA, Dec. 2015, pp. 260–264.

[37] A. Gadde, A. Knyazev, D. Tian, and H. Mansour, "Guided signal reconstruction with application to image magnification," in *Proc. IEEE Int. Conf. Signal Inf. Process.*, Orlando, FL, USA, Dec. 2015, pp. 938–942.

[38] D. Zelazo and M. Burger, "On the definiteness of the weighted Laplacian and its connection to effective resistance," in *Proc. 53rd IEEE Conf. Decis. Control*, Los Angeles, CA, USA, Dec. 2014, pp. 2895–2900.

[39] Y. Cheng, S. Z. Khong, and T. T. Georgiou, "On the definiteness of graph Laplacians with negative weights: Geometrical and passivity-based approaches," in *Proc. 2016 Amer. Control Conf.*, Boston, MA, USA, Jul. 2016, pp. 2488–2493.

[40] L. Chu *et al.*, "Finding gangs in war from signed networks," in *Proc. 22nd ACM SIGKDD Int.Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1505–1514.

[41] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. D. Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proc. SIAM Int. Conf. Data Mining*, Columbus, OH, USA, May 2010, pp. 559–570.

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

[43] A. Sandryhaila and J. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Aug. 2014.

[44] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 2014, pp. 872–876.

[45] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.

[46] A. Brew, D. Greene, and P. Cunningham, "The interaction between supervised learning and crowdsourcing," in *Proc. Comput. Social Sci. Wisdom Crowds Workshop Neural Inf. Process. Syst.*, Whistler, BC, Canada, Dec. 2010.

[47] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[48] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[49] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.

[50] H. Weyl, "Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen," *Math. Ann.*, vol. 71, pp. 441–479, 1912.

[51] N. Higham and S. H. Cheng, "Modifying the inertia of matrices arising in optimization," *ELSEVIER Linear Algebra Appl.*, vols. 275–279, pp. 261–279, May 1998.

[52] G. Golub and C. F. V. Loan, *Matrix Computations*, (Johns Hopkins Studies in the Mathematical Sciences). Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012.

[53] G. Sleijpen and H. V. D. Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 2, pp. 401–425, 1996.

[54] Y. Zhou and Y. Saad, "A Chebyshev-Davidson algorithm for large symmetric problems," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 3, pp. 954–971, 2007.

[55] F. Dörfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 150–163, Jan. 2013.

[56] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. New York, NY, USA: McGraw Hill, 1986.

[57] J. A.-F. *et al.*, "Keel: A software tool to assess evolutionary algorithms to data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Feb. 2009.

[58] L. Spacek, "Face recognition data," University Essex, Colchester, U.K., Feb. 2007. [Online]. Available: http://cswww.essex.ac.uk/mv/allfaces/faces94.html

[59] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 41–46, Sep. 1970.

[60] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

Gene Cheung (M'00–SM'07) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 1998 and 2000, respectively.

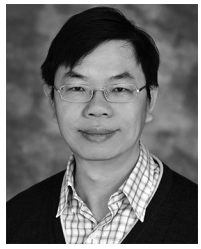
He was a Senior Researcher with Hewlett-Packard Laboratories, Tokyo, Japan, from 2000 to 2009. He is currently an Associate Professor with the National Institute of Informatics, Tokyo, Japan. Since 2015, he has also been an Adjunct Associate Professor with the Hong Kong University of Science and Technology, Hong Kong. His research interests include 3-D imaging and graph signal processing. .

Prof. Cheung was as Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA (2007–2011), the DSP Applications Column in *IEEE Signal Processing Magazine* (2010–2014), the *SPIE Journal of Electronic Imaging* (2014–2016), and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2016–2017). Since 2015, he has been an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and since 2011, for the *APSIPA Journal on Signal and Information Processing*, and since 2011, he has been an Area Editor for the *EURASIP Signal Processing: Image Communication*. He was a member of the Multimedia Signal Processing Technical Committee in the IEEE Signal Processing Society (2012–2014), and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee (2015–2017, 2018–2020). He is a coauthor of the best student paper award in the IEEE Workshop on Streaming and Media Communications 2011 (in conjunction with ICME 2011), ICIP 2013, ICIP 2017, and IVMSP 2016, the best paper runner-up award in ICME 2012, and the IEEE Signal Processing Society Japan best paper award 2016.

Weng-Tai Su received the B.S. degree in electrical engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2012, and the M.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2014. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering, National Tsing Hua University.

His research interests mainly include deep learning, computer vision, image/video processing, and image classification/clustering.

Yu Mao received the B.E. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree in informatics from The Graduate University for Advanced Studies, Tokyo, Japan, in 2016. His research interests include graph signal processing, data mining and recommendation systems. He was a recipient of the Best Student Paper Award in the IEEE Image, Video, and Multidimensional Signal Processing Workshop 2016.



Chia-Wen Lin (S'94–M'00–SM'04–F'18) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000.

He is currently a Professor with the Department of Electrical Engineering, Institute of Communications Engineering, NTHU. He is also the Deputy Director of the AI Research Center, NTHU. From 2000 to 2007, he was with the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan. Prior to joining Academia, he was with the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan, during 1992–2000. His research interests include image and video processing, computer vision, and video networking.

Dr. Lin was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE MULTIMEDIA, and the *Journal of Visual Communication and Image Representation*. He was a Steering Committee member of the IEEE TRANSACTIONS ON MULTIMEDIA from 2014 to 2015. He is a Distinguished Lecturer of IEEE Circuits and Systems Society from 2018 to 2019. He was a Chair of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society from 2013 to 2015. He was a Technical Program Cochair of the IEEE ICME 2010, and will be the General Co-Chair of IEEE VCIP 2018 and a Technical Program Co-Chair of IEEE ICIP 2019. His papers won the Best Paper Award of IEEE VCIP 2015, the Top 10% Paper Awards of IEEE MMSP 2013, and the Young Investigator Award of VCIP 2005. He was the recipient of the Young Investigator Award presented by Ministry of Science and Technology, Taiwan, in 2006.