Projection-free Graph-based Binary Classification using Gershgorin Disc Perfect Alignment

Anonymous Author(s)
Affiliation
Address
email

Abstract

In semi-supervised graph-based binary classifier learning, a subset of known labels $\hat{x}_i$ are used to deduce unknown labels, assuming that the label signal $\mathbf{x}$ is smooth with respect to a similarity graph $\mathcal{G}$ described by a Laplacian matrix $\mathbf{L}$. When restricting labels x_i to binary values, the problem is NP-hard. While a conventional semi-definite programming (SDP) relaxation can be solved in polynomial time using, for example, the alternating direction method of multipliers (ADMM), the complexity of iteratively projecting a candidate matrix $\mathbf{M}$ onto the positive semi-definite (PSD) cone ($\mathbf{M} \succeq 0$) remains high. In this paper, leveraging on a recent theory called Gershgorin disc perfect alignment (GDPA), we propose a fast projection-free method by solving a sequence of linear programs (LP) until convergence. Specifically, we first recast the SDP relaxation to its SDP dual. To solve the dual efficiently, we replace the PSD cone constraint with a set of linear constraints—sufficient conditions based on Gershgorin circle theorem (GCT) to ensure candidate $\mathbf{M} \succeq 0$—so that the optimization becomes a LP. By GDPA, these GCT lower bounds are the tightest possible if the candidate solution $\mathbf{M}$ is a graph Laplacian matrix corresponding to a balanced signed graph. We thus restrict our solution search to this set of Laplacian matrices for efficiency. Finally, we convert our converged LP solution to the SDP primal variables via complementary slackness condition. Experimental results show that ...

1 Introduction

Binary classification—assignment of labels to a N-sample set $\mathbf{x} \in \{-1, 1\}^N$ to differentiate between two distinct classes—is a basic machine learning problem [1]. One classical setting is semi-supervised graph classifier learning, where a subset of M known labels, $\hat{x}_i, 1 \leq i \leq M$, are used to deduce labels of unknown samples x_i, $M+1 \leq i \leq N$, in signal $\mathbf{x}$, assuming that $\mathbf{x}$ is smooth with respect to a similarity graph $\mathcal{G}$, described by a graph Laplacian matrix $\mathbf{L}$. This graph-based binary classification problem is NP-hard in general [2]. A conventional *semi-definite programming* (SDP) relaxation [3] converts the binary label constraint to a more relaxed *positive semi-definite* (PSD) cone constraint (*i.e.*, matrix variable satisfying $\mathbf{M} \succeq 0$), and the converted problem can be solved in polynomial time using, for example, the *alternating direction method of multipliers* (ADMM) [4]. However, ADMM still requires projection to the PSD cone $\mathcal{H} = \{\mathbf{M} \,|\, \mathbf{M} \succeq 0\}$ per iteration, which is expensive ($\mathcal{O}(N^3)$) due to full eigen-decomposition. An alternative approach first eliminates the binary constraint, minimizes directly a quadratic graph smoothness criterion called *graph Laplacian regularization* (GLR) $\mathbf{x}^\top \mathbf{L} \mathbf{x}$ [5], and then performs subsequent rounding of x_i to $\{-1, 1\}$. However, this results in sub-par performance compared to SDP relaxation [6].

To ensure matrix variable $\mathbf{M}$ is PSD without eigen-decomposition, one naïve approach is to enforce linear constraints derived directly from the *Gershgorin circle theorem* (GCT) [7]. By GCT, any real

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

eigenvalue λ of a real symmetric matrix $\mathbf{M}$ resides inside at least one *Gershgorin disc* i with center $c_i(\mathbf{M}) = M_{ii}$ and radius $r_i(\mathbf{M}) = \sum_{j \neq i} |M_{ij}|$, *i.e.*, $\exists i$ such that

$$c_i(\mathbf{M}) - r_i(\mathbf{M}) \leq \lambda \leq c_i(\mathbf{M}) + r_i(\mathbf{M}). \tag{1}$$

This implies that the smallest eigenvalue, $\lambda_{\min}(\mathbf{M})$, of $\mathbf{M}$ is lower-bounded by the smallest disc left-end, denoted by $\lambda^-_{\min}(\mathbf{M})$, *i.e.*,

$$\lambda^-_{\min}(\mathbf{M}) = \min_i c_i(\mathbf{M}) - r_i(\mathbf{M}) \leq \lambda_{\min}(\mathbf{M}). \tag{2}$$

Thus, to guarantee $\mathbf{M} \succeq 0$, one can impose the constraint $\lambda^-_{\min}(\mathbf{M}) \geq 0$. However, GCT lower bound $\lambda^-_{\min}(\mathbf{M})$ tends to be loose, and imposing this constraint naïvely would result in a sub-optimal solution to the posed SDP problem.

Orthogonally, recent advances in *graph signal processing* (GSP) [8] have led to the development of numerous graph spectral analysis and processing tools [9]. In particular, in one recent metric learning work called *Gershgorin disc perfect alignment* (GDPA) [10], it was proven that given a graph Laplacian matrix $\mathbf{L}_B$ corresponding to a balanced signed graph $\mathcal{G}_B$ [11], one can perform a similarity transform[1], $\mathbf{C} = \mathbf{S}\mathbf{L}_B\mathbf{S}^{-1}$, where $\mathbf{S} = \text{diag}(v_1^{-1}, \ldots, v_N^{-1})$ and $\mathbf{v}$ is the first eigenvector of $\mathbf{L}$, such that the Gershgorin disc left-ends of transformed matrix $\mathbf{C}$ are all perfectly aligned at $\mathbf{C}$'s smallest eigenvalue $\lambda_{\min}(\mathbf{C}) = \lambda_{\min}(\mathbf{L}_B)$. This means that transformed $\mathbf{C}$ has $\lambda^-_{\min}(\mathbf{C}) = \lambda_{\min}(\mathbf{C})$, *i.e.*, *the GCT lower bound is as tight as possible*. [10] exploits this theorem to design a fast projection-free algorithm to optimize a PSD metric matrix $\mathbf{M}$ by solving a sequence of linear programs (LP) with tight GCT linear constraints $\lambda^-_{\min}(\mathbf{S}\mathbf{M}\mathbf{S}^{-1}) \geq 0$ that replace the PSD cone constraint $\mathbf{M} \succeq 0$.

Leveraging on [10], in this paper we develop a fast projection-free algorithm to solve the SDP relaxation problem for semi-supervised graph classifier learning. Note that GDPA is applicable only for matrices that can be interpreted as graph Laplacians of balanced graphs. The crux of our proposal is observing that, while the matrix variable required to be PSD in the original SDP relaxation has no such interpretation, the variable in the corresponding *SDP dual* has this balanced graph Laplacian interpretation. Our algorithm thus performs fast GDPA-style optimization on the SDP dual, then coverts the obtained solution back to SDP primal variables via complementary slackness condition in LP. still in progress... Experimental results show that ...

2 Related Work

[12]

[13]

3 Preliminaries

3.1 Graph Definitions

Suppose we are given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$ nodes and edges $(i, j) \in \mathcal{E}$ connecting nodes i and j with weight $w_{ij} \in \mathbb{R}^+$. Denote by $\mathbf{W}$ the *adjacency matrix*, where $W_{ij} = w_{ij}$. Assuming that the edges are undirected, $\mathbf{W}$ is symmetric. Define next the diagonal *degree matrix* $\mathbf{D}$ where $D_{ii} = \sum_j W_{ij}$. The *combinatorial graph Laplacian matrix* [14] is then defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. To properly account for self-loops, the *generalized graph Laplacian matrix* [15] is defined as $\mathcal{L} = \mathbf{D} - \mathbf{W} + \text{diag}(\mathbf{W})$.

3.2 GDPA-based Optimization

A previous work on metric learning [10] has shown that given a generalized graph Laplacian matrix $\mathbf{L}$ corresponding to a balanced and connected signed graph $\mathcal{G}$ (with or without self-loops), a similarity transformation[2] $\mathbf{B} = \mathbf{S}\mathbf{L}\mathbf{S}^{-1}$ called *Gershgorin Disc Perfect Alignment* (GDPA) can be performed,

[1] Similar transform $\mathbf{B} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}$ and the original matrix $\mathbf{A}$ share the same set of eigenvealues. `https://en.wikipedia.org/wiki/Matrix_similarity`

[2] A similarity transform $\mathbf{C} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}$ of a square matrix $\mathbf{A}$, where $\mathbf{S}$ is an invertible matrix, means that $\mathbf{C}$ and $\mathbf{A}$ share the same eigenvalues.

where $\mathbf{S} = \text{diag}(1/v_1, \dots, 1/v_N)$ and $\mathbf{v}$ is the first eigenvector of $\mathbf{L}$, so that the Gershgorin disc left-ends of $\mathbf{B}$ are all aligned at $\lambda_{\min}(\mathbf{B}) = \lambda_{\min}(\mathbf{L})$. In other words, there is no gap in the GCT lower bound, *i.e.*, $\min_i c_i(\mathbf{B}) - r_i(\mathbf{B}) = \lambda_{\min}(\mathbf{B})$. Instead of (3), one can now write *signal-adaptive* linear constraints to optimize variable $\mathbf{L}^t$ at iteration t to replace the PSD cone constraint:

$$L_{ii} - \sum_{j \neq i} |s_i L_{ij} / s_j| \geq 0, \quad \forall i \in \{1, \dots, N\} \tag{3}$$

where scalars $s_i = 1/v_i^{t-1}$ and $\mathbf{v}^{t-1}$ is the first eigenvector of the previous solution $\mathbf{L}^{t-1}$ at iteration $t-1$.

4 Formulation of Graph-based Binary Classification

We first formulate the graph-based binary classification problem and relax it to an SDP problem in Section 4.1. We then convert it to its SDP dual in standard form in Section 4.2.

4.1 SDP Primal

Given a PSD combinatorial graph Laplacian matrix $\mathcal{L}$ of a similarity graph $\mathcal{G}$, one can formulate a graph-based binary classification problem as follows:

$$\min_{\mathbf{x}} \mathbf{x}^\top \mathcal{L} \mathbf{x}, \quad \text{s.t.} \left\{ \begin{array}{l} x_i^2 = 1, \; \forall i \in \{1, \dots, N\} \\ x_i = \hat{x}_i, \; \forall i \in \{1, \dots, M\} \end{array} \right. . \tag{4}$$

The objective (4) states that the reconstructed signal $\mathbf{x}$ should be smooth w.r.t. graph $\mathcal{G}$ specified by $\mathcal{L}$. Because $\mathcal{L}$ is PSD, the objective is lower-bounded by 0. The first constraint is a binary constraint that ensures $x_i \in \{-1, 1\}$. The second constraint ensures that entries x_i in reconstructed signal $\mathbf{x}$ agrees with known labels $\hat{x}_i$ in $\{1, \dots, M\}$.

Optimization (4) is NP-hard because of the binary constraint on x_i's [2]. One can define an SDP relaxation [2] as follows. Define first $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ and $\mathbf{M} = [\mathbf{X} \;\; \mathbf{x}; \; \mathbf{x}^\top \;\; 1]$. $\mathbf{M}$ is PSD because: i) scalar 1 is PSD, and ii) the Schur complement 1 of $\mathbf{M}$ is $\mathbf{X} - \mathbf{x}\mathbf{x}^\top = \mathbf{0}$ is also PSD. Thus, constraints $\mathbf{M} \succeq 0$ and rank$(\mathbf{X}) = 1$ is equivalent to $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, which together with $X_{ii} = 1, \forall i$ would imply $x_i^2 = 1, \forall i$. Instead, we drop the non-convex rank constraint and write the SDP relaxation for optimization variable $\mathbf{M}$ as

$$\min_{\mathbf{x}, \mathbf{X}} \text{Tr}(\mathcal{L}\mathbf{X}) \;\; \text{s.t.} \left\{ \begin{array}{l} X_{ii} = 1, i \in \{1, \dots, N\} \\ \mathbf{M} = \left[\begin{array}{cc} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{array} \right] \succeq 0 \\ x_i = \hat{x}_i, \;\; i \in \{1, \dots, M\} \end{array} \right. \tag{5}$$

where $\text{Tr}(\mathbf{x}^\top \mathcal{L} \mathbf{x}) = \text{Tr}(\mathcal{L}\mathbf{x}\mathbf{x}^\top) = \text{Tr}(\mathcal{L}\mathbf{X})$. Because (5) has linear objective and constraints with an additional PSD cone constraint, $\mathbf{M} \succeq 0$, it is an SDP problem, solvable in polynomial time $\mathcal{O}(N^3)$ using algorithms such as ADMM [4]. However, $\mathcal{O}(N^3)$ is still expensive for large graphs.

4.2 SDP Dual

Instead of formulation (5), we derive the dual problem based on SDP duality theory [3]. To rewrite (5) in standard form, we first define

$$\mathbf{L} = \left[\begin{array}{cc} -\mathcal{L} & \mathbf{0}_N \\ \mathbf{0}_N^\top & 0 \end{array} \right], \quad \mathbf{A}_i = \text{diag}(\mathbf{e}_{N+1}(i)), \quad \mathbf{B}_i = \left[\begin{array}{cc} \mathbf{0}_{N \times N} & \mathbf{e}_N(i) \\ \mathbf{e}_N(i)^\top & 0 \end{array} \right]. \tag{6}$$

where $\mathbf{e}_N(i) \in \{0, 1\}^N$ is a length-N binary *canonical vector* with a single non-zero entry at the i-th entry, and diag$(\mathbf{v})$ is a diagonal matrix with diagonal entries equal to $\mathbf{v}$. Note that $\mathbf{A}_i$ and $\mathbf{B}_i$ are symmetric. We can now rewrite optimization (5) as

$$\max_{\mathbf{M}} \; \mathbf{L} \cdot \mathbf{M}, \quad \text{s.t.} \left\{ \begin{array}{l} \mathbf{A}_i \cdot \mathbf{M} = 1, \quad \forall i \in \{1, \dots, N+1\} \\ \mathbf{B}_i \cdot \mathbf{M} = 2\hat{x}_i, \;\; \forall i \in \{1, \dots, M\} \\ \mathbf{M} \succeq 0 \end{array} \right. \tag{7}$$

where $\mathbf{L} \cdot \mathbf{M}$ is the inner product and equals to $\langle \mathbf{L}, \mathbf{M} \rangle = \sum_{i,j} L_{ij} M_{ij}$. The first and second constraints in (7) correspond to the first and last constraints in (5), respectively.

Given SDP in standard form (7), we now write the corresponding SDP dual formulation as follows. First, we collect M known labels $\hat{x}_i$, $i \in \{1, \ldots, M\}$, into a vector $\mathbf{b} \in \mathbb{R}^M$ of length M, *i.e.*,

$$b_i = 2\hat{x}_i, \quad \forall i \in \{1, \ldots, M\}. \tag{8}$$

We now define the SDP dual of (7) as

$$\min_{\mathbf{y},\mathbf{z}} \ \mathbf{1}_{N+1}^\top \mathbf{y} + \mathbf{b}^\top \mathbf{z}, \qquad \text{s.t. } \mathbf{H} \triangleq \sum_{i=1}^{N+1} y_i \mathbf{A}_i + \sum_{i=1}^{M} z_i \mathbf{B}_i - \mathbf{L} \succeq 0 \tag{9}$$

where dual optimization variables are $\mathbf{y} \in \mathbb{R}^{N+1}$ and $\mathbf{z} \in \mathbb{R}^M$.

4.3 Reformulating the SDP Dual

Matrix $\mathbf{H} \in \mathbb{R}^{(N+1)\times(N+1)}$ in (9) is not a graph Laplacian corresponding to a balanced signed graph. Writing $\mathbf{H}$ in sub-matrix form, we get

$$\mathbf{H} = \begin{bmatrix} \mathcal{L}_y & \mathbf{g} \\ \mathbf{g}^\top & y_{N+1} \end{bmatrix}$$

where $\mathbf{g} = [z_1, \ldots, z_M, \mathbf{0}_{N-M}]^\top$. Matrix $\mathcal{L}_y \in \mathbb{R}^{N\times N}$, equals to $\mathcal{L}_y = \text{diag}(y_1, \ldots, y_N) + \mathcal{L}$, is a Laplacian corresponding to a N-node positive graph $\mathcal{G}^+$, but $(N+1)$-th node has both positive and negative edges to $\mathcal{G}^+$ stemming from negative z_i's and positive z_i's, respectively.

Denote by $\mathcal{G}$ an $N+1$-node graph corresponding to graph Laplacian $\mathbf{H}$, where the first N nodes form a positive graph $\mathcal{G}^+$, and the $(N+1)$-th node has both positive and negative edges, with respective weights $\{w^+_{N+1,i}\}$ and $\{w^-_{N+1,i}\}$, to $\mathcal{G}^+$, and a self-loop with weight u_{N+1}. We construct an augmented graph $\bar{\mathcal{G}}$ from $\mathcal{G}$ with $N+2$ nodes as follows:

1. the first N nodes have the same inter-connections as $\mathcal{G}^+$,
2. the $(N+1)$-th node has positive edges $\{w^+_{N+1,i}\}$ and the $(N+2)$-th node has negative edges $\{w^-_{N+1,i}\}$ to the first N nodes, and
3. the $(N+1)$-th and $(N+2)$-th nodes have self-loops each with weight $u_{N+1}/2$.

Denote by $\bar{\mathbf{H}} \in \mathbb{R}^{(N+2)\times(N+2)}$ the graph Laplacian matrix corresponding to $\bar{\mathcal{G}}$. We prove that the smallest eigenvalue of $\bar{\mathbf{H}}$ is a lower bound of the smallest eigenvalue of $\mathbf{H}$.

Lemma 1 *The smallest eigenvalue $\lambda_{\min}(\bar{\mathbf{H}})$ of graph Laplacian $\bar{\mathbf{H}}$ corresponding to augmented graph $\bar{\mathcal{G}}$ is a lower bound for $\lambda_{\min}(\mathbf{H})$ of Laplacian $\mathbf{H}$ corresponding to $\mathcal{G}$, i.e.,*

$$\lambda_{\min}(\bar{\mathbf{H}}) \leq \lambda_{\min}(\mathbf{H}).$$

Proof 1 *Denote by $\mathcal{G}$ the graph represented by generalized graph Laplacian $\mathbf{H}$, with inter-node edge weights $\{w_{ij}\}$ and self-loop weights $\{u_i\}$. Denote by $\mathbf{v} \in \mathbb{R}^{N+1}$ the first eigenvector of $\mathbf{H}$ corresponding to the smallest eigenvalue $\lambda_{\min}(\mathbf{H})$. GLR of $\mathbf{H}$ computed using $\mathbf{v}$ is*

$$\begin{aligned}
\mathbf{v}^\top \mathbf{H} \mathbf{v} &= \sum_{(i,j)\in\mathcal{E}} w_{ij}(v_i - v_j)^2 + \sum_{i\in\mathcal{V}} u_i v_i^2 \\
&= \sum_{(i,j)\in\mathcal{E}\,|\,1\leq i,j\leq N} w_{ij}(v_i - v_j)^2 + \sum_{(N+1,j)\in\mathcal{E}} w_{N+1,j}(v_{N+1} - v_j)^2 + \sum_{i=1}^{N+1} u_i v_i^2
\end{aligned}$$

Now construct $\boldsymbol{\alpha} \in \mathbb{R}^{N+2}$, where $\boldsymbol{\alpha} = [v_1, \ldots, v_N, v_{N+1}, v_{N+1}]$. GLR of $\bar{\mathbf{H}}$ computed using $\boldsymbol{\alpha}$ is

$$\begin{aligned}
\boldsymbol{\alpha}^\top \bar{\mathbf{H}} \boldsymbol{\alpha} = &\sum_{(i,j)\in\mathcal{E}\,|\,1\leq i,j\leq N} w_{ij}(\alpha_i - \alpha_j)^2 + \sum_{(N+1,j)\in\mathcal{E}} w^+_{N+1,j}(\alpha_{N+1} - \alpha_j)^2 + \\
&+ \sum_{(N+2,j)\in\mathcal{E}} w^-_{N+1,j}(\alpha_{N+2} - \alpha_j)^2 + \sum_{i=1}^{N} u_i \alpha_i^2 + \frac{1}{2} u_{N+1} \alpha_{N+1}^2 + \frac{1}{2} u_{N+1} \alpha_{N+2}^2
\end{aligned}$$

Given the definition of $\boldsymbol{\alpha}$, one can see that $\mathbf{v}^\top \mathbf{H} \mathbf{v} = \boldsymbol{\alpha}^\top \bar{\mathbf{H}} \boldsymbol{\alpha}$. Since by definition the first eigenvector $\mathbf{v}$ is the vector that minimizes the Rayleigh quotient of $\mathbf{H}$, we can write

$$\lambda_{\min}(\mathbf{H}) = \frac{\mathbf{v}^\top \mathbf{H} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \overset{(a)}{\geq} \frac{\boldsymbol{\alpha}^\top \bar{\mathbf{H}} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \boldsymbol{\alpha}} \overset{(b)}{\geq} \lambda_{\min}(\bar{\mathbf{H}})$$

where (a) is true since $\mathbf{v}^\top \mathbf{v} \leq \boldsymbol{\alpha}^\top \boldsymbol{\alpha}$ by construction, and (b) is true since $\lambda_{\min}(\bar{\mathbf{H}}) = \min_{\mathbf{x}} \frac{\mathbf{x}^\top \bar{\mathbf{H}} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$.

5 Algorithm Development

5.1 GDPA-based Optimization

Instead of directly solving the SDP dual (9), we solve a sequence of linear programs (LP) via GDPA as follows. Replacing the PSD cone constraint $\mathbf{H} \succeq 0$ in (9), we write instead GCT constraints (disc center minus radius ≥ 0) for similar transform $\mathbf{S}\mathbf{H}\mathbf{S}^{-1}$, where $\mathbf{S} = \text{diag}(s_1, \ldots, s_{N+1})$, *i.e.*,

$$\begin{aligned} &\min_{\mathbf{y},\mathbf{z}} \;\; \mathbf{1}_{N+1}^\top \mathbf{y} + \mathbf{b}^\top \mathbf{z} \\ &\text{s.t.} \begin{cases} y_i + \mathcal{L}_{ii} - \sum_{j \neq i} \left| \frac{s_i}{s_j} \mathcal{L}_{ij} \right| \geq 0, & \forall i \in \{M+1, \ldots, N+1\} \\ y_i + \mathcal{L}_{ii} - \sum_{j \neq i} \left| \frac{s_i}{s_j} \mathcal{L}_{ij} \right| - \left| \frac{s_i}{s_{N+1}} z_i \right| \geq 0, & \forall i \in \{1, \ldots, M\} \\ y_{N+1} - \sum_{j=1}^{M} \left| \frac{s_{N+1}}{s_j} z_j \right| \geq 0 \end{cases} \end{aligned}$$

where the indices for summation $\sum_{j \neq i}$ are $\{1, \ldots, N\} \setminus j$. Note the absolute value operation can be appropriately remove for each term $\frac{s_i}{s_j} \mathcal{L}_{ij}$ and $\frac{s_i}{s_j} z_i$, since the sign for scalars s_i as well as $\mathcal{L}_{ij}$ and z_i are known *a priori*. We next discuss computation of suitable scalars s_i.

5.2 Scalars Computation

We split z_i's into two groups, $\mathcal{Z}^+ = \{i \,|\, z_i > 0\}$ and $\mathcal{Z}^- = \{i \,|\, z_i < 0\}$. divide $\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2$, where

$$\begin{aligned} \mathbf{H}_1 &= \alpha \left(\sum_i y_i \mathbf{A}_i - \mathbf{L} \right) + \sum_{i \in \mathcal{Z}^+} z_i \mathbf{B}_i \\ \mathbf{H}_2 &= (1 - \alpha) \left(\sum_i y_i \mathbf{A}_i - \mathbf{L} \right) + \sum_{i \in \mathcal{Z}^-} z_i \mathbf{B}_i \end{aligned}$$

Since both $\mathbf{H}_1$ and $\mathbf{H}_2$ are Laplacian corres. balanced graph, we can enforce PSD of $\mathbf{H}_1$ and $\mathbf{H}_2$ via GDPA linear constraints. Since both $\mathbf{H}_1$ and $\mathbf{H}_2$ are PSD, $\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2$ is also PSD. But how to find α? Are there better ways to split up $\sum_i y_i \mathbf{A}_i - \mathbf{L}$ into two parts?

We compute appropriate scalars s_i^{t+1} for the next iteration $t+1$ as follows. Using previous solution $(\mathbf{y}^t, \mathbf{z}^t)$ at iteration t, we first compute corresponding matrix $\tilde{\mathbf{H}}_1^t$ in (9) using $(\mathbf{y}^t, \tilde{\mathbf{z}}^t)$, where $\tilde{z}_i^t = \max(0, z_i^t), \forall i$. $\tilde{\mathbf{H}}_1^t$ is now a generalized graph Laplacian matrix corresponding to an irreducible balanced signed graph $\mathcal{G}$. We compute $\tilde{\mathbf{H}}_1^t$'s first eigenvector $\mathbf{v}^t$ via LOBPCG, and then scalars $s_i^t = 1/v_i^t, \forall i$. These computed scalars s_i^t are then used in (5.1) to compute a new solution $(\mathbf{y}_1^{t+1}, \mathbf{z}_1^{t+1})$.

Next, again using solution $(\mathbf{y}^t, \mathbf{z}^t)$ at iteration t, we first compute corresponding matrix $\tilde{\mathbf{H}}_2^t$ in (9) using $(\mathbf{y}^t, \tilde{\mathbf{z}}^t)$, where $\tilde{z}_i^t = \min(0, z_i^t), \forall i$. $\tilde{\mathbf{H}}_2^t$ is also a Laplacian corresponding to an irreducible balanced signed graph $\mathcal{G}$. We compute $\tilde{\mathbf{H}}_2^t$'s first eigenvector $\mathbf{v}^t$ and then scalars $s_i^t = 1/v_i^t, \forall i$. These computed scalars s_i^t are then used to compute solution $(\mathbf{y}_2^{t+1}, \mathbf{z}_2^{t+1})$. Finally, we keep the better of two solutions $(\mathbf{y}_1^{t+1}, \mathbf{z}_1^{t+1})$ and $(\mathbf{y}_2^{t+1}, \mathbf{z}_2^{t+1})$—one with the smaller objective value—as the solution for iteration $t+1$.

The following is not needed anymore. Suggest to remove.

We next write (**??**) in standard form. We first define variable $\mathbf{w} = [\mathbf{y} \;\; \mathbf{z} \;\; \tilde{\mathbf{z}}]$. Then we can write the following LP standard form:

$$\max_{\mathbf{w}} \; \mathbf{c}^\top \mathbf{w} \quad \text{s.t.} \;\; \mathbf{P}\mathbf{w} \leq \mathbf{d}$$

where constants $\mathbf{c} \in \mathbb{R}^K$, $\mathbf{P} \in \mathbb{R}^{K \times K}$ and $\mathbf{d} \in \mathbb{R}^K$, for $K = N + 1 + 2M$ are defined as follows:

$$\mathbf{c} = [-\mathbf{1}_{N+1}^\top \;\; -\mathbf{b}^\top \;\; -\mathbf{0}_M^\top]^\top$$

$$\mathbf{P} = \begin{bmatrix} -\mathbf{I}_{N \times N+1} & & \mathbf{0}_{N \times M} & \mathbf{E}_{N \times M} \\ \mathbf{0}_{1 \times N} & -1 & \mathbf{0}_{1 \times M} & \left|\frac{s_{N+1}}{s_{o^{-1}(1)}}\right| \dots \left|\frac{s_{N+1}}{s_{o^{-1}(M)}}\right| \\ \mathbf{0}_{M \times N+1} & & \mathbf{I}_{M \times M} & -\mathbf{I}_{M \times M} \\ \mathbf{0}_{M \times N+1} & & -\mathbf{I}_{M \times M} & -\mathbf{I}_{M \times M} \end{bmatrix}$$

$$E_{ij} = \begin{cases} \left|\frac{s_i}{s_{N+1}}\right| & \text{if } \; i \in \mathcal{F} \text{ and } j = o(i) \\ 0 & \text{o.w.} \end{cases}$$

$$d_i = \begin{cases} \mathcal{L}_{ii} - \sum_{j \neq i} \left|\frac{s_i}{s_j} \mathcal{L}_{ij}\right| & \text{if } \; i < N+1 \\ 0 & \text{if } \; i \geq N+1 \end{cases}$$

Apr-14-2021 by Cheng: fixed bugs in my implementation on Eq. (5.2). any update on this?The implementation of Eq. (5.2) works properly now. I need to look into the solutions $\mathbf{y}$ and $\mathbf{z}$ for their meanings. The dual of the LP (5.2) is

$$\min_{\mathbf{v}} \; \mathbf{d}^\top \mathbf{v} \quad \text{s.t.} \begin{cases} \mathbf{P}^\top \mathbf{v} = \mathbf{c} \\ \mathbf{v} \geq \mathbf{0} \end{cases}$$

where $\mathbf{v} \in \mathbb{R}^K$. Apr-20-2021 by Cheng: Specifically,

$$v_i = \begin{cases} 1, \;\; i \leq N+1 \\ \frac{1}{2}\left(\left|\frac{s_i}{s_{N+1}}\right| + \left|\frac{s_{N+1}}{s_{o^{-1}(j)}}\right| - b_{o(j)}\right), \;\; N+1 < i \leq N+1+M, \; j = i - N - 1 \\ \frac{1}{2}\left(\left|\frac{s_i}{s_{N+1}}\right| + \left|\frac{s_{N+1}}{s_{o^{-1}(j)}}\right| + b_{o(j)}\right), \;\; N+1+M < i \leq N+1+2M, \; j = i - N - 1 - M \end{cases}$$

Apr-14-2021 by Cheng: the implementation of Eq. (5.2) works properly. I also need to look into the solutions $\mathbf{v}$ for its meaning. In terms of objective values, as an example, I ran Eq.(5), Eq.(7), Eq.(9), Eq.(5.2), and Eq.(5.2) using the same data samples, with the following objective values: Eq.(5): 6.5372, Eq.(7): -6.5372, Eq.(9): -6.5372, Eq.(5.2): 0.0054359, and Eq.(5.2): -0.0056196. For Eq.(5.2) and Eq.(5.2), we need to first initialize $\mathbf{y}$ and $\mathbf{z}$ to compute a set of initial scalars. The initialization is important since it significantly affect the converged objective value. I need to look into this as well. Note that d_i, $i < N+1$, is the Gershgorin disc i's left-end. The objective is thus to minimize a weighted sum of disc left-ends using non-negative variable $\mathbf{v}$.

We can first solve Eq. (**??**) via interior-point. Then, we can solve Eq. (5.2) to get $\mathbf{v}$ via the following complementary slackness condition add ref:

$$\begin{cases} (\mathbf{d} - \mathbf{P}\mathbf{w})^\top \mathbf{v} = 0 \\ (\mathbf{P}^\top \mathbf{v} - \mathbf{c})^\top \mathbf{w} = 0 \end{cases}$$

6 Experimental Results

competing schemes: 1. SDP interior-point/simplex/ADMM from [4]. 2. SDCut fast SDP from [13]. 3. GLR quadratic closed-form. 4. LP approaches to SDP [16].

7 Conclusion

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.

[2] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[3] Lieven Vandenberghe and Stephen P. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[4] N. Rontsis, P. J. Goulart, and Y. Nakatsukasa, "Efficient semidefinite programming with approximate ADMM," 2019.

[5] 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," in *IEEE Signal Processing Magazine*, May 2013, vol. 30, no.3, pp. 83–98.

[6] Y. Bengio, Olivier Delalleau, and Nicolas Roux, "Label propagation and quadratic criterion," *Semi-Supervised Learning*, 09 2006.

[7] R. S. Varga, *Gershgorin and his circles*, Springer, 2004.

[8] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," in *Proceedings of the IEEE*, May 2018, vol. 106, no.5, pp. 808–828.

[9] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, "Graph spectral image processing," in *Proceedings of the IEEE*, May 2018, vol. 106, no.5, pp. 907–930.

[10] C. Yang, G. Cheung, and W. Hu, "Signed graph metric learning via gershgorin disc perfect alignment," 2020.

[11] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory.," *Psychological review*, vol. 63 5, pp. 277–93, 1956.

[12] Z. Li, J. Liu, and X. Tang, "Pairwise constraint propagation by semidefinite programming for semi-supervised classification," in *ACM International Conferene on Machine Learning*, Helsinki, Finland, July 2008.

[13] P. Wang, C. Shen, and A. van den Hengel, "A fast semidefinite approach to solving binary quadratic problems," in *IEEE International Conference on Computer Vision and Pattern Recognition*, Portland, OR, June 2013.

[14] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," in *IEEE Transactions on Signal Processing*, July 2016, vol. 64, no.14, pp. 3775–3789.

[15] H. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," in *IEEE Journal of Selected Topics in Signal Processing*, July 2017, vol. 11, no.6, pp. 825–841.

[16] K. K. Sivaramakrishnan, *Linear Programming Approaches to Semidefinite Programming Problems*, Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, New York, 2002.