Semi-Supervised Learning with Adaptive Spectral Transform

Hanxiao Liu
School of Computer Science
Carnegie Mellon University

Yiming Yang
School of Computer Science
Carnegie Mellon University

Abstract

This paper proposes a novel nonparametric framework for semi-supervised learning and for optimizing the Laplacian spectrum of the data manifold simultaneously. Our formulation leads to a convex optimization problem that can be efficiently solved via the bundle method, and can be interpreted as to asymptotically minimize the generalization error bound of semi-supervised learning with respect to the graph spectrum. Experiments over benchmark datasets in various domains show advantageous performance of the proposed method over strong baselines.

1 INTRODUCTION

Graph representation of data is ubiquitous in machine learning. In many scenarios, we are given a partially labeled graph with only a small number of labeled vertices, and the task is to predict the missing labels of the large number of unlabeled vertices.

With limited supervision available, it is often crucial to leverage the intrinsic manifold structure of both the labeled and unlabeled vertices during the training phase. Various graph-based semi-supervised learning (SSL) algorithms have been proposed for this purpose, including label propagation [1], Gaussian fields [2] and Laplacian Support Vector Machines [3]. Many of those approaches rely on the assumption that strongly connected vertices are likely to share the same labels, and fall under the manifold regularization framework [4] where the graph Laplacian [5] plays a key role.

Given a graph, the graph Laplacian characterizes how the label of each vertex diffuses (propagates) from itself to its *direct* neighbors. While the graph Laplacian in its original form may not be sufficiently expressive for modeling complex graph transduction patterns, it has been shown that a rich family of important graph transduction patterns under various assumptions, such as multi-step random walk, heat diffusion [6] and von-Neumann diffusion [7], can be incorporated into SSL by transforming the spectrum[1] of the graph Laplacian with nonnegative nondecreasing functions [8, 9, 10]. The collection of those functions are referred to as the *Spectral Transformation* (ST) family.

Despite of the expressiveness of the ST family, how to find the optimal ST for any problem in hand is an open challenge. While manual specification [8, 10] is clearly suboptimal, various approaches have been proposed to automatically find the optimal ST. Among the existing works, parametric approaches assume the optimal ST belongs some pre-specified function family (e.g. the polynomial or exponential), and then find the function hyperparameter via grid search or curve-fitting [11]. However, the fundamental question about how to choose the function family is left unanswered, and it is not clear whether commonly used parametric function families are rich enough to subsume the true optimal ST. On the other hand, a more flexible nonparametric framework based on kernel-target alignment has been studied in [9], where the optimization of ST can be efficiently solved via quadratically constrained quadratic programming (QCQP). However, the target matrix itself may be unreliable as it is constructed based a very small number of observed labels, and it is not conclusive whether a better alignment score always leads to a better prediction performance.

Note all the above approaches are *two-step* procedures, where the optimal ST is empirically estimated in some preprocessing step before SSL is carried out (with the ST obtained in the previous step). We argue that the separation of ST-finding from SSL may result in suboptimal performance, as combining the two steps together will allow the learned ST to better adapt to the problem structure.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

[1] In this paper, we refer to the spectrum of a matrix as the multiset of its eigenvalues.

This paper addresses the above challenge by proposing a principled optimization framework which *simultaneously* conducts SSL and finds the optimal ST for the graph Laplacian used in SSL. Starting with the natural formulation of the joint optimization, we show how to reformulate it as an equivalent *convex* optimization problem via Lagrangian duality, and then derive an efficient algorithm using the bundle method. We refer to our new approach as *Adaptive Spectral Transform* (AST), meaning that the ST is automatically adapted to the problem in hand and its target domain.

Besides strong empirical performance over benchmark datasets across various domains, insights are provided regarding the advantageous performance of AST by revisiting an existing theorem on SSL from a new angle. Specifically, we show that AST aims to asymptotically minimize the generalization error bound of SSL.

2 THE PROPOSED FRAMEWORK

Let us start with the formal definition of SSL (2.1), and then move on to the spectral transformation (ST) of the graph Laplacian (2.2), and finally the adaptation of ST (AST) for specific problems (2.3).

2.1 SSL with the Graph Laplacian

Given a graph G of m vertices, where each vertex denotes an instance and each edge encodes the affinity between a pair of instances. Suppose only a very small set $\mathcal{T}$ of l vertices has been labeled where $l \ll m$, our task is to predict the missing labels of the remaining $m-l$ vertices based on both the l labeled vertices and the intrinsic manifold structure of G.

Denote by y_i the true label and by $\mathrm{f}_i \in \mathbb{R}$ the system-estimated score for vertex i, resp. In order to leverage the labels, we hope f_i and y_i to be as close as possible for all $i \in \mathcal{T}$. Meanwhile, to leverage the large amount of unlabeled vertices, we want the scores for all (both labeled and unlabeled) vertices to be smooth w.r.t. the graph structure of G. The two desired properties entail the following optimization problem:

$$\min_{\mathrm{f}\in\mathbb{R}^m} \frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \mathrm{f}^\top \mathcal{L}\mathrm{f} \tag{1}$$

where the first term is the empirical loss of the system-predicted scores $\mathrm{f} \in \mathbb{R}^m$, $\mathcal{L}$ in the second term is the normalized graph Laplacian matrix associated with G characterizing G's manifold structure. Specifically, denote by A the adjacency matrix of G, by D a diagonal matrix of degrees with $d_{ii} = \sum_j a_{ij}$ and by $L = D - A$ the graph Laplacian. The normalized graph Laplacian is defined as $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, with its eigensystem denoted by $\{(\lambda_i, \phi_i)\}_{i=1}^m$. For convenience, we assume the eigenvalues of $\mathcal{L}$ are in the increasing order: $\lambda_1 \leq \lambda_2 \ldots \leq \lambda_m$. It is well known that the smallest eigenvalue λ_1 is always zero, and that ϕ_i's with small indices tend to be "smoother" over the data manifold than those with large indices [5].

In (1), the label information is encoded in the empirical loss $\ell(\mathrm{f}_i, y_i)$. E.g., one could specify $\ell(\mathrm{f}_i, y_i)$ to be $(\mathrm{f}_i - y_i)^2$. The manifold assumption is encoded in the second term (a.k.a. the manifold regularizer) involving the graph Laplacian, satisfying

$$\mathrm{f}^\top \mathcal{L}\mathrm{f} \equiv \frac{1}{2}\sum_{i\sim j} a_{ij}\left(\frac{\mathrm{f}_i}{\sqrt{d_{ii}}} - \frac{\mathrm{f}_j}{\sqrt{d_{jj}}}\right)^2 \tag{2}$$

$$\equiv \sum_{i=1}^m \lambda_i \langle \phi_i, \mathrm{f}\rangle^2 \tag{3}$$

Eq. (2) suggests that the regularizer essentially encourages scores f_i, f_j (normalized by the squared root of degrees) to be close when vertices i, j are strongly connected in G, namely when a_{ij} is large. An alternative perspective, as implied by (3), is to think of the regularizer as penalizing the projection of f onto different bases (the $\phi_i' s$) with different weights (the $\lambda_i' s$), where the smooth components in f are going to receive lighter penalty than the nonsmooth ones.

2.2 Transforming the Laplacian Spectrum

Although the graph Laplacian gives a nice characterization about how vertices in G influence their direct neighbors, it is not sufficiently expressive for modeling complex label propagation patterns, such as multi-step influence from a given vertex to its indirect neighbors and the decay of such influence. As a simple remedy to incorporate a richer family of label propagation patterns over the manifold, various methods have been proposed based on transforming the spectrum of $\mathcal{L}$ using some nonnegative nondecreasing function, known as the spectral transformation [8, 9, 10].

As an example, by taking the exponential of the Laplacian spectrum, one gets $\sum_{i=1}^m e^{\beta\lambda_i}\phi_i\phi_i^\top = e^{\beta\mathcal{L}}$ where β is a nonnegative scalar. The transformed Laplacian has a neat physical interpretation in terms of heat diffusion process, and is closely related to infinite random walk with decay over the manifold [6]. From (3)'s perspective, the replacement of λ_i with e^{λ_i} can be viewed as a way to exaggerate the difference in weighing the bases. That is, the nonsmooth components in f are going to receive a larger relative penalty during the optimization after the exponential transformation.

Formally, we define the *Spectral Transformation* (ST) over $\mathcal{L}$ as $\sigma(\mathcal{L}) := \sum_{i=1}^m \sigma(\lambda_i)\phi_i\phi_i^\top$, where $\sigma : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is a nondecreasing function which transforms each Laplacian eigenvalue to a nonnegative scalar. Besides the aforementioned diffusion kernel where $\sigma(x) = e^{\beta x}$, other commonly used STs include $\sigma(x) = x+\beta$ (Gaussian field), $\sigma(x) = \frac{1}{(\alpha-x)^\beta}$ (multi-step random walk), $\sigma(x) = \left[\cos\left(\frac{\pi}{4}x\right)\right]^{-1}$ (inverse cosine) [8, 11], etc.

The ST-enhanced SSL is formulated as

$$\min_{\mathrm{f}\in\mathbb{R}^m} \; \frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \mathrm{f}^\top \sigma(\mathcal{L})\mathrm{f} \tag{4}$$

2.3 Adapting the Spectral Transform

The nature of SSL described in (4) crucially depends our choice of ST. It is a common practice to manually specify σ [8, 10] or to learn the hyperparameter of σ within a pre-specified function family [12, 11]. Both methods are suboptimal when the true optimal σ^* lies in a broader function space.

In this paper, we focus on automatically learning σ^* from data with no assumption on its function form. In terms of SSL, we argue it suffices to learn $\{\sigma^*(\lambda_i)\}_{i=1}^m$ instead of the analytical expression of σ^*, as the objective in (4) is uniquely determined by these m transformed eigenvalues. Therefore, in the following we switch from the task of making σ adaptive to the equivalent task of making each $\sigma(\lambda_i)$ adaptive.

Define $\theta \in \mathbb{R}^m$ where $\theta_i := \sigma(\lambda_i)^{-1}$. We are going to focus on learning θ as notation-wise it is more convenient to work with the reciprocals. After substituting the ST σ with θ in (4), the optimization becomes

$$\min_{\mathrm{f}\in\mathbb{R}^m} \; \underbrace{\frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \sum_{i=1}^m \theta_i^{-1} \langle \phi_i, \mathrm{f}\rangle^2}_{\underset{\sim}{C}(\mathrm{f};\theta)} \tag{5}$$

When $\theta_i = 0$, we define $\theta_i^{-1} := 0$ as its pseudo-inverse. For brevity, in the following we assume all the θ_i's are strictly positive. The singular case where some θ_i's are exactly zero will be studied specifically in Section 3.4.

To determine θ for (5), Zhu et al. [9] proposed a two-step procedure based on empirical kernel-target alignment. In the first step, an empirical estimation about θ is obtained by maximizing the alignment score between the kernel matrix implied by θ, i.e. $\sum_i \theta_i \phi_i \phi_i^\top$, and a target kernel matrix induced from a small amount of observed labels. In the second step, the estimated $\hat{\theta}$ is plugged-into the SSL objective (5) for learning f.

Different from existing (manual/parametric/two-step) approaches, we argue that it is beneficial to put the task of finding the optimal θ^* and the task of SSL into a unified optimization framework, as the two procedures can mutually reinforce each other, thus making θ^* more adapted to the problem structure.

It may appear straightforward to approach the aforementioned goal by minimizing (5) w.r.t. f and w.r.t. θ in an alternating manner. Unfortunately, the resulting optimization is non-convex, and a meaningless solution can be obtained by simply setting all the θ_i^{-1}'s to zero.

Instead, we propose to achieve this goal by solving the following optimization problem (AST)

$$\min_{\theta\in\Theta} \; \left(\min_{\mathrm{f}\in\mathbb{R}^m} \; \underset{\sim}{C}(\mathrm{f},\theta)\right) + \tau\|\theta\|_1 \tag{6}$$

where $\underset{\sim}{C}(\mathrm{f};\theta)$ is the SSL objective defined in (5), τ is a positive scalar-valued tuning parameter, and Θ is the space of θ, i.e. the set of all possible reciprocals of the transformed Laplacian spectrum

$$\begin{aligned} \Theta &= \left\{\theta : \theta_i = \sigma(\lambda_i)^{-1}, \forall i = 1, 2 \dots m,\ \sigma \text{ is a valid ST}\right\} \\ &\equiv \{\theta : \theta_1 \geq \theta_2, \dots \geq \theta_m \geq 0\} \end{aligned} \tag{7}$$

One may check the second equality above by recalling that (i) the λ_i's are in the increasing order (ii) σ can be *any* nonnegative nondecreasing function.

The intuition behind optimization (6) is that we want the optimal θ^* (and the associated optimal ST) to simultaneously satisfy the following criteria:

(a) It should tend to minimize the SSL objective (5). As in multiple kernel learning [13, 14], this is arguably the most natural and effective way to make θ^* adaptive to the problem structure.

(b) It should have a moderate ℓ_1-norm. Namely the "transformed" data manifold should have a moderate total effective resistance [15]. This additional requirement is crucial as it precludes degenerate solutions. It also adds to the stability of our bundle method for optimization (Section 3).

3 OPTIMIZATION STRATEGIES

Let us present our optimization strategies for solving (6), starting with the following theorem

Theorem 1 (Convexity of AST)**.**
(6) is a convex optimization problem over θ.

After presenting the proof for Theorem 1 (Section 3.1), we propose our method to compute the gradient for (6)'s structured objective function in Section 3.2, and offer a bundle method for efficient optimization in Section 3.3. We will study the singular case where some θ_i's are allowed to be exactly zero in Section 3.4, which can be particularly useful in large-scale scenarios. The SSL subroutine for AST is discussed in Section 3.5.

3.1 Proof of Convexity

We proof Theorem 1 by first reformulating (6)'s objective function into an equivalent minimax-type function via Lagrangian duality, and then showing the convexity of the equivalent optimization problem.

The Lagrangian dual for $\underline{C}(\mathrm{f};\theta)$ is

$$\underbrace{-\omega(-u) - \frac{1}{4\gamma}\sum_{i=1}^{m}\theta_i \langle \phi_i, u\rangle^2}_{\bar{C}(u;\theta)} \tag{8}$$

where $\omega(\cdot)$ is the conjugate function for $\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i)$. It is not hard to verify that the Slater's condition holds for optimization (5), i.e. $\min_{\mathrm{f}\in\mathbb{R}^m} \ \underline{C}(\mathrm{f};\theta)$, thus strong duality ensures that

$$\min_{\mathrm{f}\in\mathbb{R}^m} \underline{C}(\mathrm{f};\theta) = \max_{u\in\mathbb{R}^m} \ \bar{C}(u;\theta) \tag{9}$$

and optimization (6) for AST can be recast as

$$\min_{\theta\in\Theta} \ \underbrace{\left(\max_{u\in\mathbb{R}^m} \ \bar{C}(u;\theta)\right)}_{g(\theta)} + \tau\|\theta\|_1 \tag{10}$$

We claim the resulting equivalent problem (10) is convex over θ. To see this, notice that $\bar{C}(u;\theta)$ defined in (8) is an affine over θ for each given u, and recall that the pointwise maximum of any set of convex functions (affines) is still convex, the first structured term $g(\theta)$ in optimization (10), i.e. $\max_{u\in\mathbb{R}^m} \ \bar{C}(u;\theta)$, is hence convex over θ. The conclusion follows by further noticing the second term $\|\theta\|_1$ in (10) is also a convex function, and that Θ defined in (7) is a convex domain.

3.2 Computing the Structured Gradient

In this section, we discuss our method to compute the gradient of $g(\theta) := \max_u \ \bar{C}(u;\theta)$ in (10), denoted by $\nabla_\theta g(\theta)$, as a prerequisite for subsequent optimization algorithms. We rely on Danskin's Theorem [16] as $g(\theta)$ is the maximum of infinite number of functions:

Theorem 2 (Danskin's Theorem)**.** *If function $g(\theta)$ is in the form of $g(\theta) := \max_{u\in\mathcal{U}} \bar{C}(u;\theta)$ where $\mathcal{U}$ is a compact space and $\bar{C}(\cdot;\theta)$ is a differentiable function with $\bar{C}(u;\theta)$ and $\nabla\bar{C}(u;\theta)$ depending continuously on u and θ, then the subgradient of $g(\theta)$, i.e. $\partial_\theta g(\theta)$, is given by $\partial_\theta \bar{C}(\hat{u};\theta)$ where $\hat{u} \in \mathrm{argmax}_{u\in\mathcal{U}} \ \bar{C}(u;\theta)$.*

For our case $\mathcal{U} := \mathbb{R}^m$ and the subgradient $\partial_\theta g(\theta)$ can be substituted with gradient $\nabla_\theta g(\theta)$ as the function of interest is differentiable. Recall that we have assumed all θ_i's to be positive, $\bar{C}(u;\theta)$ is strictly convex over u and therefore $\hat{u} := \mathrm{argmax}_u \ \bar{C}(u;\theta)$ is always unique. Suppose $\hat{u}$ is given, following Theorem 2 we have

$$\nabla_\theta g(\theta) = \nabla_\theta \bar{C}(\hat{u};\theta) = -\frac{\left(\langle\phi_1,\hat{u}\rangle^2, \ldots, \langle\phi_m,\hat{u}\rangle^2\right)^\top}{4\gamma} \tag{11}$$

To compute the R.H.S. of (11), we have to get $\hat{u}$ in advance via solving $\max_u \bar{C}(u;\theta)$. In case the conjugate function involved in $\bar{C}(u;\theta)$ is hard to work with, it is more convenient to first obtain the primal solution $\hat{\mathrm{f}}$ by solving the corresponding primal problem $\min_{\mathrm{f}} \ \underline{C}(\mathrm{f};\theta)$ described in (5), and then recover the dual solution $\hat{u}$ from $\hat{\mathrm{f}}$ via the K.K.T. condition.

According to the stationarity condition, $\hat{u}$ and $\hat{\mathrm{f}}$ must satisfy $\hat{u} = 2\gamma\left(\sum_{i=1}^m \theta_i^{-1}\phi_i\phi_i^\top\right)\hat{\mathrm{f}}$. This suggests an alternative to (11), i.e. to compute the gradient of $g(\theta)$ directly based on the primal variable via

$$\nabla_\theta g(\theta) = -\gamma\left(\frac{\langle\phi_1,\hat{\mathrm{f}}\rangle^2}{\theta_1^2}, \ldots, \frac{\langle\phi_m,\hat{\mathrm{f}}\rangle^2}{\theta_m^2}\right)^\top \tag{12}$$

where $\hat{\mathrm{f}} := \mathrm{argmin}_{\mathrm{f}} \ \underline{C}(\mathrm{f},\theta)$ is obtained by applying any SSL algorithm[2] to (5).

3.3 Bundle Method for AST

After obtaining $\nabla_\theta g(\theta)$ according to section 3.2, it is straightforward to minimize the AST objective in (10): $g(\theta) + \gamma\|\theta\|_1$ via the subgradient method or proximal gradient method. However, both algorithms have slow convergence rate, and it can be tricky to choose a suitable step size to ensure efficient convergence.

We propose to use the bundle method for (10) (equivalently, (6)), which has been found particularly efficient in solving problems involving structured loss functions [17, 18]. Our method is a variant of bundle method for regularized risk minimization (BMRM) [19], and subsumes the semi-infinite linear programming (SILP) for large-scale multiple kernel learning [20].

The key idea is to replace the "tough" part in (10), i.e. $g(\theta)$, with an "easy" piecewise linear function $\tilde{g}(\theta)$ that lowerbounds the original $g(\theta)$. After the replacement, optimization (10) becomes

$$\min_{\theta\in\Theta} \ \tilde{g}(\theta) + \tau\|\theta\|_1 \tag{13}$$

We then alternate between solving the surrogate problem (13) and refining the lowerbound $\tilde{g}(\theta)$ until convergence. Note (13) is a Linear Programming (LP), as its objective function is piecewise linear and its feasible set Θ defined in (7) is a polyhedron.

[2] Many off-the-shelf SSL solvers can be easily modified for solving the primal problem (5).

```
Algorithm 1: Bundle Method for AST
Input
  $\epsilon$  desired convergence accuracy
  $\mathcal{L}$  normalized graph Laplacian of $G$
  $\ell$  loss function based on available labels
  $\gamma$  tuning parameter for manifold regularization in (5)
  $\tau$  tuning parameter for the $\ell_1$-norm in (6)
Output
  f  system-inferred vertex labels
  $\theta$  system-inferred reciprocals of
       the transformed Laplacian eigenvalues
Initialization
$t \leftarrow 0$;
/*take pseudo-inverse when necessary*/;
$\{\lambda_i, \phi_i\}_{i=1}^m \leftarrow \text{eig}(\mathcal{L})$, $\{\theta_i^{(0)} \leftarrow \lambda_i^{-1}\}_{i=1}^m$;
do
    /*solve (5) via standard SSL*/;
    $\mathrm{f}^{(t)} \leftarrow \text{argmin}_{\mathrm{f}\in\mathbb{R}^m}\ \underline{C}(\mathrm{f};\theta^{(t)})$;
    $g(\theta^{(t)}) \leftarrow \underline{C}(\mathrm{f}^{(t)};\theta^{(t)})$;
    /*according to (12)*/;
    $\nabla g(\theta^{(t)}) \leftarrow -\gamma\left(\frac{\langle\phi_1,\mathrm{f}^{(t)}\rangle^2}{\theta_1^{(t)2}}, \ldots, \frac{\langle\phi_m,\mathrm{f}^{(t)}\rangle^2}{\theta_m^{(t)2}}\right)^\top$;
    $t \leftarrow t + 1$;
    /*update the piecewise-linear lowerbound*/;
    $\tilde{g}^{(t)}(\theta) \leftarrow \max_{0\le i\le t-1}\ g(\theta^{(i)}) + \langle\nabla g(\theta^{(i)}), \theta - \theta^{(i)}\rangle$;
    /*solve the linear programing*/;
    $\theta^{(t)} \leftarrow \text{argmin}_{\theta\in\{\theta|\theta_1\ge\theta_2\ge\ldots\ge\theta_m\ge0\}}\ \tilde{g}^{(t)}(\theta) + \tau\|\theta\|_1$;
while $g(\theta^{(t-1)}) + \|\theta^{(t-1)}\|_1 - \tilde{g}^{(t)}(\theta^{(t)}) - \|\theta^{(t)}\|_1 > \epsilon$;
/*terminate when the piecewise-linear lowerbound is sufficiently close to the original function*/;
```

To obtain a piecewise lowerbound $\tilde{g}(\theta)$ for $g(\theta)$, recall any convex function can be lowerbounded by its tangents. Hence it suffices to let $\tilde{g}(\theta)$ be the supremum of a set of tangents associated with historical iterations. Specifically, we define $\tilde{g}(\theta)$ at the t-th iteration as

$$\tilde{g}^{(t)}(\theta) := \max_{0\le i\le t-1}\ g(\theta^{(i)}) + \left\langle \nabla g(\theta^{(i)}), \theta - \theta^{(i)} \right\rangle \tag{14}$$

where superscript "$^{(i)}$" indexes the quantity associated with the i-th iteration. It is not hard to verify that $g^{(t)}(\theta) \le g(\theta)$ always holds, and that $g^{(t)}(\theta)$ tends to better approximate $g(\theta)$ as t increases. Details of the bundle method for AST is presented in Algorithm 1.

3.4 Singular Cases: Towards More Scalability

Now let us focus on the singular cases where some θ_i's (and their pseudo-inverse θ_i^{-1}'s) are exactly zero. This may happen in two scenarios:

(a) During the bundle method, some θ_i's are shrunk to zero after solving the LP (13) due to the presence of the ℓ_1-regularization over θ.

(b) Small-valued θ_i's associated with those nonsmooth ϕ_i's are truncated to be zero for the sake of scalability. This strategy will substantially reduce the parameter size of SSL, and has been successfully applied to large-scale problems [21].

In the following, we will assume $\theta_i > 0$ for $1 \le i \le k$ and $\theta_i = 0$ for $k < i \le m$, where $k \ll m$. To handle the singular case, we modify $\underline{C}(\mathrm{f};\theta)$ in (5) as

$$\frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \sum_{1\le i\le k} \theta_i^{-1} \left\langle \phi_i, \mathrm{f} \right\rangle^2 + \sum_{k<i\le m} \mathbf{1}_{\{\langle\phi_i,\mathrm{f}\rangle=0\}} \tag{15}$$

where $\mathbf{1}_{\{\cdot\}}$ equals zero if the inside-bracket condition is satisfied and equals $+\infty$ otherwise. The third term in (15) is crucial in that otherwise the projection of f onto ϕ_i for any $k < i \le m$ will be left unregularized and the resulting model can easily over-fit.

The solution f^* for minimizing (15) must lie in the span of $\{\phi_i\}_{i=1}^k$ as otherwise the indicator function will go to infinity. Let $\mathrm{f} := \sum_{1\le j\le k} \alpha_j \phi_j$. (15) can be reduced to consist of only k ($k \ll m$) parameters

$$\frac{1}{l}\sum_{i\in\mathcal{T}} \ell\left(e_i^\top \sum_{1\le j\le k} \alpha_j \phi_j, y_i\right) + \gamma \sum_{1\le i\le k} \theta_i^{-1} \alpha_i^2 \tag{16}$$

where e_i stands for the i-th unit vector in $\mathbb{R}^m$.

Applying similar analysis[3] in the previous subsections to the modified $\underline{C}(\mathrm{f};\theta)$ in (15), for singular cases the gradient of $g(\theta)$ during bundle method is given by

$$\nabla_\theta g(\theta) = -\gamma\left(\frac{\langle\phi_1,\hat{\mathrm{f}}\rangle^2}{\theta_1^2}, \ldots \frac{\langle\phi_k,\hat{\mathrm{f}}\rangle^2}{\theta_k^2}, 0, \ldots 0\right)^\top \tag{17}$$

$$\equiv -\gamma\left(\frac{\hat{\alpha_1}^2}{\theta_1^2}, \ldots \frac{\hat{\alpha_k}^2}{\theta_k^2}, 0, \ldots 0\right)^\top \tag{18}$$

where $\hat{\mathrm{f}}$ and $\hat{\alpha}$ are solutions for minimizing (15) and minimizing (16), respectively. Eq. (18) holds because $\langle\phi_i, \hat{\mathrm{f}}\rangle = \sum_{1\le j\le k} \hat{\alpha}_j \langle\phi_i, \phi_j\rangle = \hat{\alpha}_i$.

To carry out bundle method for the singular case, we need to compute $\nabla_\theta g(\theta)$ via (18), which requires $\hat{\alpha}$ as the solution of minimizing (16). Compared to solving optimization (5) w.r.t. $\mathrm{f} \in \mathbb{R}^m$ for the non-singular case, minimizing (16) w.r.t. $\alpha \in \mathbb{R}^k$ can be performed much more efficiently due to the substantially reduced parameter size (recall $k \ll m$). In fact, once the top-k eigenvalues/eigenvectors $\{\lambda_j, \phi_j\}_{j=1}^k$ of $\mathcal{L}$ is obtained, the time/space complexity for both the LP subroutine and the SSL subroutine (16) in AST will become independent from m, which is desirable for large problems.

[3] The analysis follows Sections 3.1, 3.3 and 3.2. We omit the details due to the space limit.

3.5 Solving the SSL Subroutine

Both the original and the singular AST involve solving a standard SSL problem as their intermediate subroutines, i.e. minimizing (5) w.r.t. f or minimizing (16) w.r.t. α. Here we use the later to demonstrate how existing off-the-self machine learning toolkits can be conveniently leveraged for this purpose.

We choose the squared hinge loss as our loss function $\ell(\cdot,\cdot)$. Besides large-margin property, its smoothness often leads to efficient optimization [22]. In this case, minimizing (16) can be formulated as

$$\min_{\alpha\in\mathbb{R}^k} \frac{1}{l}\sum_{i\in\mathcal{T}} \max\left(1-y_i e_i^\top \Phi\alpha, 0\right)^2 + \gamma\alpha^\top \mathrm{diag}(\theta_1^{-1}, \theta_2^{-1}, \dots \theta_k^{-1})\alpha \tag{19}$$

where $\Phi = [\phi_1, \phi_2, \dots \phi_k] \in \mathbb{R}^{m\times k}$.

By defining $C := (\gamma l)^{-1}$, $w_j := \alpha_j\sqrt{\frac{2}{\theta_j}}$ for $1 \le j \le k$ and $x_i := \mathrm{diag}\left(\sqrt{\frac{\theta_1}{2}}, \sqrt{\frac{\theta_2}{2}} \dots \sqrt{\frac{\theta_k}{2}}\right)\Phi^\top e_i$ for $\forall i \in \mathcal{T}$, (19) can be recast as

$$\min_{w\in\mathbb{R}^k} C\sum_{i\in\mathcal{T}} \max\left(1-y_i\langle x_i, w\rangle, 0\right)^2 + \frac{1}{2}\|w\|_2^2 \tag{20}$$

Note that (20) is the standard formulation of L2-SVM and can be efficiently solved via existing solvers such as LIBLINEAR [23]. After obtaining the solution $\hat{w}$ for (20), the solution $\hat{\alpha}$ for (19) can be easily recovered by rescaling $\hat{w}$, and then be plugged-into (17) to compute $\nabla_\theta g(\theta)$ required by the bundle method.

4 THEORETICAL INSIGHTS

In this section we provide theoretical intuitions to justify the proposed method. We are going to show that AST can be interpreted as an automatic procedure to asymptotically minimize the SSL generalization error bound w.r.t. different STs.

Our analysis is based an existing theorem on the relationship between the generalization performance of SSL and any given (fixed) graph-Laplacian spectrum [10]. While proving the theorem is not the contribution of this paper, our method provides a new angle to utilize the theorem. To the best of our knowledge, none of the previous work, including [10], have formulated or provided any algorithmic solution to *automatically* determine the optimal spectrum among all candidate spectrums in this manner (i.e. formulating and solving optimization (6)).

Theorem 3 (Adapted from [10]). *Suppose indices of the labeled vertices in $\mathcal{T}$ are sampled from* $\{1, 2, \dots, m\}$ *uniformly at random. Let $\hat{\mathrm{f}}(\mathcal{T})$ be the system-predicted scores in $\mathbb{R}^m$ obtained via solving optimization* (4) *for any given $\mathcal{T}$, and let ℓ be a convex loss function such that $|\nabla\ell| \le b$. We have*

$$\frac{1}{m-l}\mathbb{E}_{\mathcal{T}}\sum_{i\notin\mathcal{T}} \ell\left(\hat{\mathrm{f}}_i(\mathcal{T}), y_i\right) \le \left(\min_{\mathrm{f}\in\mathbb{R}^m} \frac{1}{m}\sum_{i=1}^{m} \ell\left(\mathrm{f}_i, y_i\right) + \gamma\mathrm{f}^\top\sigma(\mathcal{L})\mathrm{f}\right) + \frac{b^2\mathrm{tr}\left(\sigma(\mathcal{L})^{-1}\right)}{2\gamma lm} \tag{21}$$

The L.H.S. of (21) stands for the empirical risk of SSL for any given ST σ.

To see the connections between AST and Theorem 3, let $\tau = \frac{b^2}{2\gamma lm}$ and recall that $\sigma(\mathcal{L}) = \sum_{i=1}^m \sigma(\lambda_i)\phi_i\phi_i^\top = \sum_{i=1}^m \theta_i^{-1}\phi_i\phi_i^\top$, we rewrite the R.H.S. of (21) as

$$\left(\min_{\mathrm{f}\in\mathbb{R}^m} \frac{1}{m}\sum_{i=1}^{m} \ell\left(\mathrm{f}_i, y_i\right) + \gamma\sum_{i=1}^{m}\theta_i^{-1}\left\langle\phi_i, \mathrm{f}\right\rangle^2\right) + \tau\|\theta\|_1 \tag{22}$$

By comparing the AST objective function in (6) with (22), we see that AST is essentially trying to minimize a surrogate of (22) where the true loss $\frac{1}{m}\sum_{i=1}^m \ell\left(\mathrm{f}_i, y_i\right)$ based on all the m vertex labels is substituted by the empirical loss $\frac{1}{l}\sum_{i\in\mathcal{T}} \ell\left(\mathrm{f}_i, y_i\right)$ based on l partially observed vertex labels. The two loss functions are asymptotically equivalent as $l \to m$. This substitution is necessary since in practice it is impossible for us to access all of the m vertex labels during the training phase.

Notice there is an additional isotonic constraint $\theta_1 \ge \theta_2 \dots \theta_m \ge 0$ for AST when minimizing the generalization error bound (22) w.r.t. θ, indicating AST always favours the smooth components over the non-smooth ones in the final prediction $\hat{\mathrm{f}}$.

5 EXPERIMENTS

5.1 Methods for Comparison

We compare the performance of the following methods in our experiments:

(a) **SSL** is the standard SSL in (1) with squared hinge loss. This amounts to taking the ST in (4) to be the identity function $\sigma(x) = x$.

(b) **Diffusion** is the ST-enhanced SSL described in (4), where σ is parametrized as $\sigma(x) = e^{\beta x}$ a.k.a. the heat diffusion kernel. Prior to SSL, β is empirically estimated by maximizing the kernel alignment score [12] via grid search over $[10^{-4}, 10^4]$.

(c) **GRF** is another ST-enhanced SSL algorithm with $\sigma(x) = x+\beta$, a.k.a. the kernel of Gaussian random field. As in Diffusion, β is empirically estimated before SSL via kernel alignment over $[10^{-5}, 10^3]$.

(d) **NKTA** is nonparametric kernel-target alignment [9], a two-step procedure for ST-enhanced SSL. Prior to SSL, we find σ that maximizes the kernel alignment score without assuming its parametric form. Then, we solve (4) with the empirically estimated ST. We follow the formulation of [9] and solve the QCQP subroutine using SeDuMi[4].

(e) **AST** is our proposed method of Adaptive Spectral Transform. Different from the aforementioned two-step kernel alignment approaches, the optimal ST is obtained *along with* SSL by solving the convex optimization problem (6) via bundle method.

5.2 Experimental Settings

We compare AST against the baselines over the benchmark datasets from three different domains:

1. **20NewsGroup** for document classification. We use the PC-vs-Mac subset consisting of 1,993 documents with binary labels. Following [9], a symmetrized unweighted 10-nearest neighbor (10NN) graph is constructed based on the cosine similarity between documents.

2. **Isolet** for spoken letter recognition consisting of 7,797 instances from 26 classes [5]. We construct a 10NN graph using the Euclidean distance between the audio features.

3. **MNIST** for pattern recognition of the handwritten digits. We use the full training set consisting of 60,000 images from 10 classes (digits 0-9). A 10NN graph is constructed based on the Euclidean distance among the images.

For all datasets, parameter γ for manifold regularization is fixed to be 10^{-3} for all methods as we find the results are not sensitive to the choice of γ. Instead of tuning the hyperparameter τ for our method AST, we simply fix it to be 10^{-2} across all experiments. For all datasets, only the top-50 Laplacian eigenvectors are used for SSL. For AST we use the singular version as described in Section 3.4 with $k = 50$.

Given a dataset of m data points, we randomly sample l labeled vertices and predict the remaining unlabeled $m - l$ vertices with methods described in subsection 5.1. The training size l gradually increases from 2^4 to 2^7, and the experiment is repeated for 30 times for each given training size. The mean and standard variance of the prediction accuracy are reported.

5.3 Results

The results are presented in Table 1, 2 and 3. For all aforementioned baselines, the prediction accuracy improves and the variance tends to decrease as we gradually enlarge the training size.

Table 1: Results on 20NewsGroup (PC-vs-Mac)

Training Size	16	32	64	128
SSL	70.3 ± 15.0	78.8 ± 9.6	80.1 ± 6.3	81.2 ± 0.3
Diffusion	69.4 ± 10.7	75.0 ± 7.3	82.5 ± 3.6	85.6 ± 2.2
GRF	70.3 ± 13.6	74.1 ± 8.9	77.6 ± 6.3	80.8 ± 5.1
NKTA	72.0 ± 17.5	75.0 ± 15.1	82.0 ± 10.8	87.6 ± 4.6
AST	**72.5** **± 13.9**	**79.7** **± 8.9**	**86.9** **± 3.0**	**88.5** **± 2.1**

Table 2: Results on Isolet

Training Size	16	32	64	128
SSL	33.3 ± 6.2	40.0 ± 7.6	44.6 ± 7.8	60.3 ± 6.7
Diffusion	32.5 ± 5.9	47.3 ± 4.5	59.8 ± 4.3	66.8 ± 2.8
GRF	32.7 ± 6.5	40.9 ± 8.6	45.8 ± 7.6	61.3 ± 6.6
NKTA	32.4 ± 6.7	40.7 ± 6.7	48.9 ± 9.5	64.7 ± 5.2
AST	**34.0** **± 4.3**	**48.7** **± 4.4**	**60.1** **± 4.4**	**67.4** **± 2.6**

Table 3: Results on MNIST

Training Size	16	32	64	128
SSL	70.1 ± 7.2	81.5 ± 6.5	89.5 ± 2.2	92.8 ± 1.5
Diffusion	**71.7** **± 6.5**	84.0 ± 5.0	91.0 ± 1.6	93.1 ± 1.5
GRF	68.8 ± 6.8	80.9 ± 6.2	89.4 ± 2.2	92.8 ± 1.4
NKTA	61.3 ± 16.9	77.0 ± 10.0	91.0 ± 2.5	94.3 ± 0.9
AST	68.5 ± 6.9	**84.4** **± 5.3**	**92.8** **± 1.4**	**94.5** **± 0.9**

First, it is evident that all ST-enhanced methods outperform the traditional SSL on average, which justifies the effectiveness of allowing richer graph transduction

[4] http://sedumi.ie.lehigh.edu/downloads

[5] Algorithms in subsection 5.1 can be trivially extended to the multi-class case by decomposing the original problem into multiple binary SSL tasks.

patterns over the data manifold.

Secondly, among two-step methods based on empirical kernel-target alignment, it is evident that the nonparametric method NKTA outperforms the two parametric methods Diffusion and GRF, which justifies our previous argument that pre-specifying ST to be within some common function family is too restrictive to accurately capture the “true” graph transduction pattern.

Finally, between nonparametric methods, we observe that the performance of AST dominates NKTA over all datasets. This confirms our intuition that ST-finding and SSL are able to mutually reinforce each other during the joint optimization. The advantageous empirical performance of AST also justifies our previous theoretical analysis in Section 4.

We also notice AST yields much more stable performance than NKTA. We conjecture that NKTA might be subject to noise as it is trying to fit the target kernel matrix—a quantity induced from only a very limited amount of labels. On the other hand, AST is designed to be adaptive to the problem structure—an arguably more robust reference.

We plotted out the STs produced by different baseline methods over MNIST when $l = 128$ in Figure 1. Each sub-figure contains 30 curves in total corresponding to the 30 different runs. From the figure we see that while the STs produced by Diffusion and GRF are restricted to specific parametric forms, STs produced by NKTA and AST are more flexible. Figure 1 also shows that STs produced by AST tend to be have lower variance than those produced by NKTA, which justifies our previous stability claim about AST.

An empirical comparison of the speed of all the baseline methods is presented in Table 4.

Table 4: Total CPU time taken by different methods over the MNIST dataset when $l = 128$ given the top-50 eigenvalues/eigenvectors. The convergence tolerance ϵ for AST is set to be 10^{-3}.

Method	SSL	Diffusion	GRF	NKTA	AST
Time (secs)	0.148	0.564	0.738	24.152	2.556

6 CONCLUSION

We proposed a new nonparametric framework for carrying out SSL and finding the Laplacian spectrum of the data manifold simultaneously. Different from existing two-step approaches based on manual specification or kernel-target alignment, our approach unifies both tasks into a joint optimization problem and is naturally adaptive to the problem structure. Our formulation enjoys convexity and can be efficiently solved using the bundle method. Theoretical insights are provided to show that the proposed algorithm attempts to asymptotically minimize the SSL generalization error bound w.r.t. the Laplacian spectrum. The merits of our framework are verified by its advantageous empirical performance over strong baselines.

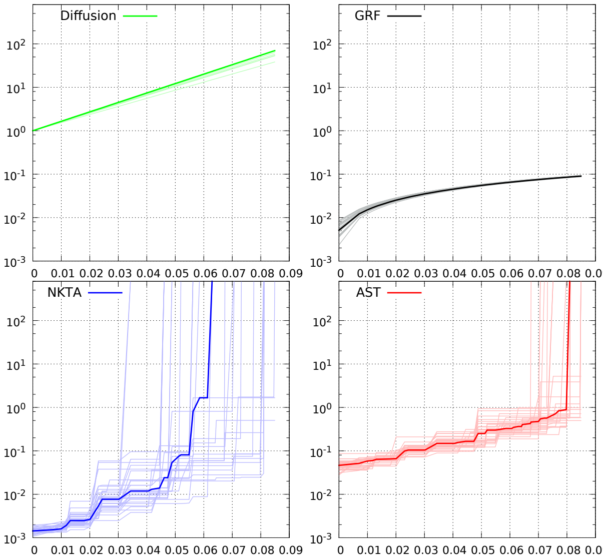

Figure 1: STs produced by methods described in subsection 5.1 on the MNIST dataset (each sub-figure contains the results of 30 different runs), where the x-axis and y-axis (log-scale) correspond to the original spectrum λ_i's and the transformed spectrum $\sigma(\lambda_i)$'s, resp.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work is supported in part by the National Science Foundation (NSF) under grants IIS-1546329, IIS-1216282 and IIS-1350364.

References

[1] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.

[2] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

[3] Stefano Melacci and Mikhail Belkin. Laplacian support vector machines trained in the primal. *The Journal of Machine Learning Research*, 12:1149–1184, 2011.

[4] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric

framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[5] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[6] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.

[7] Takahiko Ito, Masashi Shimbo, Taku Kudo, and Yuji Matsumoto. Application of kernels to link analysis. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 586–592. ACM, 2005.

[8] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[9] Xiaojin Zhu, Jaz Kandola, Zoubin Ghahramani, and John D Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 1641–1648, 2004.

[10] Rie Johnson and Tong Zhang. Graph-based semi-supervised learning and spectral kernel design. *Information Theory, IEEE Transactions on*, 54(1):275–288, 2008.

[11] Jérôme Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009.

[12] N Shawe-Taylor and A Kandola. On kernel target alignment. *Advances in neural information processing systems*, 14:367, 2002.

[13] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[14] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

[15] Stephen Boyd. Convex optimization of graph laplacian eigenvalues. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1311–1319, 2006.

[16] John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

[17] Marius Kloft, Ulf Brefeld, Pavel Laskov, Klaus-Robert Müller, Alexander Zien, and Sören Sonnenburg. Efficient and accurate lp-norm multiple kernel learning. In *Advances in neural information processing systems*, pages 997–1005, 2009.

[18] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012.

[19] Choon Hui Teo, SVN Vishwanthan, Alex J Smola, and Quoc V Le. Bundle methods for regularized risk minimization. *The Journal of Machine Learning Research*, 11:311–365, 2010.

[20] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[21] Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *Advances in neural information processing systems*, pages 522–530, 2009.

[22] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *The Journal of Machine Learning Research*, 9:1369–1398, 2008.

[23] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.