Laplacian Spectrum Learning

Pannagadatta K. Shivaswamy and Tony Jebara

Department of Computer Science
Columbia University
New York, NY 10027
{pks2103,jebara}@cs.columbia.edu

Abstract. The eigenspectrum of a graph Laplacian encodes smoothness information over the graph. A natural approach to learning involves transforming the spectrum of a graph Laplacian to obtain a kernel. While manual exploration of the spectrum is conceivable, non-parametric learning methods that adjust the Laplacian's spectrum promise better performance. For instance, adjusting the graph Laplacian using kernel target alignment (KTA) yields better performance when an SVM is trained on the resulting kernel. KTA relies on a simple surrogate criterion to choose the kernel; the obtained kernel is then fed to a large margin classification algorithm. In this paper, we propose novel formulations that jointly optimize relative margin and the spectrum of a kernel defined via Laplacian eigenmaps. The large relative margin case is in fact a strict generalization of the large margin case. The proposed methods show significant empirical advantage over numerous other competing methods.

Keywords: relative margin machine, graph Laplacian, kernel learning, transduction

1 Introduction

This paper considers the transductive learning problem where a set of labeled examples is accompanied with unlabeled examples whose labels are to be predicted by an algorithm. Due to the availability of additional information in the unlabeled data, both the labeled and unlabeled examples will be utilized to estimate a kernel matrix which can then be fed into a learning algorithm such as the support vector machine (SVM). One particularly successful approach for estimating such a kernel matrix is by transforming the spectrum of the graph Laplacian [8]. A kernel can be constructed from the eigenvectors corresponding to the smallest eigenvalues of a Laplacian to maintain smoothness on the graph. In fact, the diffusion kernel [5] and the Gaussian field kernel [12] are based on such an approach and explore smooth variations of the Laplacian via specific parametric forms. In addition, a number of other transformations are described in [8] for exploring smooth functions on the graph. Through the controlled variation of the spectrum of the Laplacian, a family of allowable kernels can be explored in an attempt to improve classification accuracy. Further, Zhang & Ando [10] provide generalization analysis for spectral kernel design.

Kernel target alignment (KTA for short) [3] is a criterion for evaluating a kernel based on the labels. It was initially proposed as a method to choose a kernel from a family of candidates such that the Frobenius norm of the difference between a label matrix and the kernel matrix is minimized. The technique estimates a kernel independently of the final learning algorithm that will be utilized for classification. Recently, such a method was proposed to transform the spectrum of a graph Laplacian [11] to select from a general family of candidate kernels. Instead of relying on parametric methods for exploring a family of kernels (such as the scalar parameter in a diffusion or Gaussian field kernel), Zhu *et al.* [11] suggest a more general approach which yields a kernel matrix nonparametrically that aligns well with an ideal kernel (obtained from the labeled examples).

In this paper, we propose novel quadratically constrained quadratic programs to jointly learn the spectrum of a Laplacian with a large margin classifier. The motivation for large margin spectrum transformation is straightforward. In kernel target alignment, a simpler surrogate criterion is first optimized to obtain a kernel by transforming the graph Laplacian. Then, the kernel obtained is fed to a classifier such as an SVM. This is a two-step process with a different objective function in each step. It is more natural to transform the Laplacian spectrum jointly with the classification criterion in the first place rather than using a surrogate criterion to learn the kernel.

Recently, another discriminative criterion that generalizes large absolute margin has been proposed. The large *relative* margin [7] criterion measures the margin relative to the spread of the data rather than treating it as an absolute quantity. The key distinction is that large relative margin jointly maximizes the margin while controlling or minimizing the spread of the data. Relative margin machines (RMM) implement such a discriminative criterion through additional linear constraints that control the spread of the projections. In this paper, we consider this aggressive classification criterion which can potentially improve over the KTA approach. Since large absolute margin and large relative margin criteria are more directly tied to classification accuracy and have generalization guarantees, they potentially could identify better choices of kernels from the family of admissible kernels. In particular, the family of kernels spanned by spectral manipulations of the Laplacian will be considered. Since the RMM is more general compared to SVM, by proposing a large relative margin spectrum learning, we encompass large margin spectrum learning as a special case.

1.1 Setup and notation

In this paper we assume that a set of labeled examples $(\mathbf{x}_i, y_i)_{i=1}^{l}$ and an unlabeled set $(\mathbf{x}_i)_{i=l+1}^{n}$ are given such that $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{\pm 1\}$. We denote by $\mathbf{y} \in \mathbb{R}^l$ the vector whose i^{th} entry is y_i and by $\mathbf{Y} \in \mathbb{R}^{l \times l}$ a diagonal matrix such that $\mathbf{Y}_{ii} = y_i$. The primary aim is to obtain predictions on the unlabeled examples; we are thus in a so-called transductive setup. However, the unlabeled examples can be also be utilized in the learning process.

Assume we are given a graph with adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ where the weight $\mathbf{W}_{ij}$ denotes the edge weight between nodes i and j (corresponding to the examples $\mathbf{x}_i$ and $\mathbf{x}_j$). Define the graph Laplacian as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D}$ denotes a diagonal matrix whose i^{th} entry is given by the sum of the i^{th} row of $\mathbf{W}$. We assume that $\mathbf{L} = \sum_{i=1}^{n} \theta_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top$ is the eigendecomposition of $\mathbf{L}$. It is assumed that the eigenvalues are already arranged such that $\theta_i \leq \theta_{i+1}$ for all i. Further, we let $\mathbf{V} \in \mathbb{R}^{n \times q}$ be the matrix whose i^{th} column is the $(i+1)^{\text{th}}$ eigenvector (corresponding to the $(i+1)^{\text{th}}$ smallest eigenvalue) of $\mathbf{L}$. Note that the first eigenvector (corresponding to the smallest eigenvalue) has been deliberately left out from this definition. Further, $\mathbf{U} \in \mathbb{R}^{n \times q}$ is defined to be the matrix whose i^{th} column is the i^{th} eigenvector. $\mathbf{v}_i$ ($\mathbf{u}_i$) denotes the i^{th} column of $\mathbf{V}^\top$ ($\mathbf{U}^\top$). For any eigenvector (such as $\boldsymbol{\phi}, \mathbf{u}$ or $\mathbf{v}$), we use the horizontal overbar (such as $\bar{\phi}, \bar{\mathbf{u}}$ or $\bar{\mathbf{v}}$) to denote the subvector containing only the first l elements of the eigenvector, in other words, only the entries that correspond to the labeled examples. We overload this notation for matrices as well; thus $\bar{\mathbf{V}} \in \mathbb{R}^{l \times q}$ ($\bar{\mathbf{U}} \in \mathbb{R}^{l \times q}$) denotes [1] the first l rows of $\mathbf{V}$ ($\mathbf{U}$). $\boldsymbol{\Delta}$ is assumed to be a $q \times q$ diagonal matrix whose diagonal elements denote scalar values δ_i (*i.e.,* $\boldsymbol{\Delta}_{ii} = \delta_i$). Finally $\mathbf{0}$ and $\mathbf{1}$ denote vectors of all zeros and all ones; their dimensionality can be inferred from the context.

2 Learning from the graph Laplacian

The graph Laplacian has been particularly popular in transductive learning. While we can hardly do justice to all the literature, this section summarizes some of the most relevant previous approaches.

Spectral Graph Transducer The spectral graph transducer [4] is a transductive learning method based on a relaxation of the combinatorial graph-cut problem. It obtains predictions on labeled and unlabeled examples by solving for $\mathbf{h} \in \mathbb{R}^n$ via the following problem:

$$\min_{\mathbf{h} \in \mathbb{R}^n} \frac{1}{2} \mathbf{h}^\top \mathbf{V} \mathbf{Q} \mathbf{V}^\top \mathbf{h} + C(\mathbf{h} - \boldsymbol{\tau})^\top \mathbf{P} (\mathbf{h} - \boldsymbol{\tau}) \;\; \text{s.t. } \mathbf{h}^\top \mathbf{1} = 0, \;\; \mathbf{h}^\top \mathbf{h} = n \qquad (1)$$

where $\mathbf{P}$ is a diagonal matrix[2] with $\mathbf{P}_{ii} = \frac{1}{l_+}$ ($\frac{1}{l_-}$) if the i^{th} example is positive (negative); $\mathbf{P}_{ii} = 0$ for unlabeled examples (i.e., for $l+1 \leq i \leq n$). Further, $\mathbf{Q}$ is also a diagonal matrix. Typically, the diagonal element $\mathbf{Q}_{ii}$ is set to i^2 [4]. $\boldsymbol{\tau}$ is a vector in which the values corresponding to the positive (negative) examples are set to $\sqrt{\frac{l_-}{l_+}}$ ($\sqrt{\frac{l_+}{l_-}}$).

Non-parametric transformations via kernel target alignment (KTA) In [11], a successful approach to learning a kernel was proposed which involved transforming the spectrum of a Laplacian in a non-parametric way. The empirical

[1] We clarify that $\bar{\mathbf{V}}^\top$ ($\bar{\mathbf{U}}^\top$) denotes the transpose of $\bar{\mathbf{V}}$ ($\bar{\mathbf{U}}$).

[2] $l_+(l_-)$ is the number of positive (negative) labeled examples.

alignment between two kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ is defined as [3]:

$$\hat{A}(\mathbf{K}_1, \mathbf{K}_2) := \frac{\langle \mathbf{K}_1, \mathbf{K}_2 \rangle_F}{\sqrt{\langle \mathbf{K}_1, \mathbf{K}_1 \rangle_F \langle \mathbf{K}_2, \mathbf{K}_2 \rangle_F}}.$$

When the target $\mathbf{y}$ (the vector formed by concatenating y_i's) is known, the ideal kernel matrix is $\mathbf{y}\mathbf{y}^\top$ and a kernel matrix $\mathbf{K}$ can be learned by maximizing the alignment $\hat{A}(\bar{\mathbf{K}}, \mathbf{y}\mathbf{y}^\top)$. The kernel target alignment approach [11] learns a kernel via the following formulation:[3]

$$\max_{\boldsymbol{\Delta}} \ \hat{A}(\bar{\mathbf{U}}\boldsymbol{\Delta}\bar{\mathbf{U}}^\top, \mathbf{y}\mathbf{y}^\top) \tag{2}$$
$$\text{s.t. } \operatorname{trace}(\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top) = 1 \qquad \delta_i \geq \delta_{i+1} \ \forall 2 \leq i \leq q-1, \quad \delta_q \geq 0, \quad \delta_1 \geq 0.$$

The above optimization problem transforms the spectrum of the given graph Laplacian $\mathbf{L}$ while maximizing the alignment score of the labeled part of the kernel matrix ($\bar{\mathbf{U}}\boldsymbol{\Delta}\bar{\mathbf{U}}^\top$) with the observed labels. The trace constraint on the overall kernel matrix ($\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$) is used merely to control the arbitrary scaling. The above formulation can be posed as a quadratically constrained quadratic program (QCQP) that can be solved efficiently [11]. The ordering on the δ's is in reverse order as that of the eigenvalues of $\mathbf{L}$ which amounts to monotonically inverting the spectrum of the graph Laplacian $\mathbf{L}$. Only the first q eigenvectors are considered in the formulation above due to computational considerations.

The eigenvector $\boldsymbol{\phi}_1$ is made up of a constant element. Thus, it merely amounts to adding a constant to all the elements of the kernel matrix. Therefore, the weight on this vector (i.e. δ_1) is allowed to vary freely. Finally, note that the $\boldsymbol{\phi}$'s are the eigenvectors of $\mathbf{L}$ so the trace constraint on $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$ merely corresponds to the constraint $\sum_{i=1}^{q} \delta_i = 1$ since $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$.

Parametric transformations A number of methods have been proposed to obtain a kernel from the graph Laplacian. These methods essentially compute the Laplacian over labeled and unlabeled data and transform its spectrum with a particular mapping. More precisely, a kernel is built as $\mathbf{K} = \sum_{i=1}^{n} r(\theta_i)\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\top$ where $r(\cdot)$ is a monotonically decreasing function. Thus, an eigenvector with a small eigenvalue will have a large weight in the kernel matrix. Several methods fall into this category. For example, the diffusion kernel [5] is obtained by the transformation $r(\theta) = \exp(-\theta/\sigma^2)$ and the Gaussian field kernel [12] uses the transformation $r(\theta) = \frac{1}{\sigma^2+\theta}$. In fact, kernel PCA [6] also performs a similar operation. In kPCA, we retain the top k eigenvectors of a kernel matrix. From an equivalence that exists between the kernel matrix and the graph Laplacian (shown in the next section), we can in fact conclude that kernel PCA features also fall under the same family of monotonic transformations. While these are very interesting transformations, [11] showed that KTA and learning based approaches are empirically superior to parametric transformations so we will not elaborate further on these approaches but rather focus on learning the spectrum of a graph Laplacian.

[3] In fact, [11] proposes two formulations, we are considering the one that was shown to have superior performance (the so-called improved order method).

3 Why learn the Laplacian spectrum?

We start with an optimization problem which is closely related to the spectral graph transducer (1). The main difference is in the choice of the loss function. Consider the following optimization problem:

$$\min_{\mathbf{h}\in\mathbb{R}^n} \frac{1}{2}\mathbf{h}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{h} + C\sum_{i=1}^{l} \max(0, 1 - y_i \mathbf{h}_i), \tag{3}$$

where $\mathbf{Q}$ is assumed to be an invertible diagonal matrix to avoid degeneracies.[4] The values on the diagonal of $\mathbf{Q}$ depend on the particular choice of the kernel. The above optimization problem is essentially learning the predictions on all the examples by minimizing the so-called hinge loss and the regularization defined by the eigenspace of the graph Laplacian. The choice of the above formulation is due to its relation to the large margin learning framework given by the following theorem.

Theorem 1. *The optimization problem* (3) *is equivalent to*

$$\min_{\mathbf{w},b} \ \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{l} \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{Q}^{-\frac{1}{2}} \mathbf{v}_i + b)). \tag{4}$$

Proof. The predictions on all the examples (without the bias term) for the optimization problem (4) are given by $\mathbf{f} = \mathbf{V}\mathbf{Q}^{-\frac{1}{2}}\mathbf{w}$. Therefore $\mathbf{Q}^{\frac{1}{2}}\mathbf{V}^\top \mathbf{f} = \mathbf{Q}^{\frac{1}{2}}\mathbf{V}^\top\mathbf{V}\mathbf{Q}^{-\frac{1}{2}}\mathbf{w} = \mathbf{w}$ since $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$. Substituting this expression for $\mathbf{w}$ in (4), the optimization problem becomes,

$$\min_{\mathbf{f},b} \ \frac{1}{2}\mathbf{f}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{f} + C\sum_{i=1}^{l} \max(0, 1 - y_i(\mathbf{f}_i + b)).$$

Let $\mathbf{h} = \mathbf{f} + b\mathbf{1}$ and consider the first term in the objective above,

$$\begin{aligned}&(\mathbf{h} - b\mathbf{1})^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top (\mathbf{h} - b\mathbf{1})\\ =&\mathbf{h}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{h} + 2\mathbf{h}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{1} + \mathbf{1}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{1} = \mathbf{h}^\top \mathbf{V}\mathbf{Q}\mathbf{V}^\top \mathbf{h},\end{aligned}$$

where we have used the fact that $\mathbf{V}^\top\mathbf{1} = \mathbf{0}$ since the eigenvectors in $\mathbf{V}$ are orthogonal to $\mathbf{1}$. This is because $\mathbf{1}$ is always an eigenvector of $\mathbf{L}$ and other eigenvectors are orthogonal to it. Thus, the optimization problem (3) follows. □

The above theorem[5] thus implies that learning predictions with Laplacian regularization in (3) is equivalent to learning in a large margin setting (4). It

[4] In practice, $\mathbf{Q}$ can be non-invertible, but we consider an invertible $\mathbf{Q}$ to elucidate the main point.

[5] Although we excluded ϕ_1 in the definition of $\mathbf{V}$ in these derivation, typically we would include it in practice and allow the weight on it to vary freely as in the kernel target alignment approach. However, experiments show that the algorithms typically choose a negligible weight on this eigenvector.

is easy to see that the implicit kernel for the learning algorithm (4) (over both labeled and unlabeled examples) is given by $\mathbf{V}\mathbf{Q}^{-1}\mathbf{V}^\top$. Thus, computing predictions on all examples with $\mathbf{V}\mathbf{Q}\mathbf{V}^\top$ as the regularizer in (3) is equivalent to large margin learning with the kernel obtained by inverting the spectrum $\mathbf{Q}$. However, it is not clear why inverting the spectrum of a Laplacian is the right choice for a kernel. The parametric methods presented in the previous section construct this kernel by exploring specific parametric forms. On the other hand, the kernel target alignment approach constructs this kernel by maximizing alignment with labels while maintaining an ordering on the spectrum. The spectral graph transducer in Section 2 uses[6] the transformation i^2 on the Laplacian for regularization. In this paper, we explore a family of transformations and allow the algorithm to choose the one that best conforms to a large (relative) margin criterion. Instead of relying on parametric forms or using a surrogate criteria, this paper presents approaches that jointly obtain a transformation and a large margin classifier.

4 Relative margin machines

Relative margin machines (RMM) [7] measure the margin relative to the data spread; this approach has yielded significant improvement over SVMs and has enjoyed theoretical guarantees as well. In its primal form, the RMM solves the following optimization problem:[7]

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{s.t. } & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \ |\mathbf{w}^\top\mathbf{x}_i + b| \leq B \qquad \forall 1 \leq i \leq l. \end{aligned} \tag{5}$$

Note that when $B = \infty$, the above formulation gives back the support vector machine formulation. For values of B below a threshold, the RMM gives solutions that differ from SVM solutions. The dual of the above optimization problem can be shown to be:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}} \ & -\frac{1}{2}\boldsymbol{\gamma}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\gamma} + \boldsymbol{\alpha}^\top\mathbf{1} - B\left(\boldsymbol{\beta}^\top\mathbf{1} + \boldsymbol{\eta}^\top\mathbf{1}\right) \\ \text{s.t. } & \boldsymbol{\alpha}^\top\mathbf{y} - \boldsymbol{\beta}^\top\mathbf{1} + \boldsymbol{\eta}^\top\mathbf{1} = 0,\ \ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},\ \ \boldsymbol{\beta} \geq \mathbf{0},\ \ \boldsymbol{\eta} \geq \mathbf{0}. \end{aligned} \tag{6}$$

In the dual, we have defined $\boldsymbol{\gamma} := \mathbf{Y}\boldsymbol{\alpha} - \boldsymbol{\beta} + \boldsymbol{\eta}$ for brevity. Note that $\boldsymbol{\alpha} \in \mathbb{R}^l$, $\boldsymbol{\beta} \in \mathbb{R}^l$ and $\boldsymbol{\eta} \in \mathbb{R}^l$ are the Lagrange multipliers corresponding to the constraints in (5).

[6] Strictly speaking, the spectral graph transducer has additional constraints and a different motivation.

[7] The constraint $|\mathbf{w}^\top\mathbf{x}_i + b| \leq B$ is typically implemented as two linear constraints.

4.1 RMM on Laplacian eigenmaps

Based on the motivation from earlier sections, we consider the problem of jointly learning a classifier and weights on various eigenvectors in the RMM setup. We restrict the family of weights to be the same as that in (2) in the following problem:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,\boldsymbol{\Delta}} \ & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{l}\xi_i & (7)\\
\text{s.t. } & y_i(\mathbf{w}^\top\boldsymbol{\Delta}^{\frac{1}{2}}\mathbf{u}_i + b) \geq 1-\xi_i,\ \xi_i \geq 0 & \forall 1 \leq i \leq l\\
& |\mathbf{w}^\top\boldsymbol{\Delta}^{\frac{1}{2}}\mathbf{u}_i + b| \leq B & \forall 1 \leq i \leq l\\
& \delta_i \geq \delta_{i+1} \ \ \forall 2 \leq i \leq q-1, \ \ \delta_1 \geq 0, \ \ \delta_q \geq 0,\\
& \operatorname{trace}(\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top) = 1.
\end{aligned}$$

By writing the dual of the above problem over $\mathbf{w}$, b and ξ, we get:

$$\begin{aligned}
\min_{\boldsymbol{\Delta}} \ \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}} \ & -\frac{1}{2}\boldsymbol{\gamma}^\top\bar{\mathbf{U}}\boldsymbol{\Delta}\bar{\mathbf{U}}^\top\boldsymbol{\gamma} + \boldsymbol{\alpha}^\top\mathbf{1} - B\left(\boldsymbol{\beta}^\top\mathbf{1} + \boldsymbol{\eta}^\top\mathbf{1}\right) & (8)\\
& \text{s.t. } \boldsymbol{\alpha}^\top\mathbf{y} - \boldsymbol{\beta}^\top\mathbf{1} + \boldsymbol{\eta}^\top\mathbf{1} = 0,\ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},\ \boldsymbol{\beta} \geq \mathbf{0},\ \boldsymbol{\eta} \geq \mathbf{0},\\
& \qquad \delta_i \geq \delta_{i+1} \ \ \forall 2 \leq i \leq q-1, \ \ \delta_1 \geq 0,\ \delta_q \geq 0,\ \sum_{i=1}^{q}\delta_i = 1.
\end{aligned}$$

where we exploited the fact that $\operatorname{trace}(\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top) = \sum_{i=1}^{q}\delta_i$. Clearly, the above optimization problem, without the ordering constraints (*i.e.,* $\delta_i \geq \delta_{i+1}$) is simply the multiple kernel learning[8] problem (using the RMM criterion instead of the standard SVM). A straightforward derivation–following the approach of [1]–results in the corresponding multiple kernel learning optimization. Even though the optimization problem (8) without the ordering on δ's is a more general problem, it may not produce smooth predictions over the entire graph. This is because, with a small number of labeled examples (*i.e.,* small l), it is unlikely that multiple kernel learning will maintain the spectrum ordering unless it is explicitly enforced. In fact, this phenomenon can frequently be observed in our experiments where multiple kernel learning fails to maintain a meaningful ordering on the spectrum.

5 STORM and STOAM

This section poses the optimization problem (8) in a more canonical form to obtain practical large-margin (denoted by STOAM) and large-relative-margin (denoted by STORM) implementations. These implementations achieve globally optimal joint estimates of the kernel and the classifier of interest. First, the min

[8] In this paper, we restrict our attention to convex combination multiple kernel learning algorithms.

and the max in (8) can be interchanged since the objective is concave in $\boldsymbol{\Delta}$ and convex in $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ and both are strictly feasible [2][9]. Thus, we can write:

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}} \min_{\boldsymbol{\Delta}} \; & -\frac{1}{2}\boldsymbol{\gamma}^\top \sum_{i=1}^{q} \delta_i \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \boldsymbol{\gamma} + \boldsymbol{\alpha}^\top \mathbf{1} - B\left(\boldsymbol{\beta}^\top \mathbf{1} + \boldsymbol{\eta}^\top \mathbf{1}\right) \\ \text{s.t. } & \boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\beta}^\top \mathbf{1} + \boldsymbol{\eta}^\top \mathbf{1} = 0, \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},\ \boldsymbol{\beta} \geq \mathbf{0},\ \boldsymbol{\eta} \geq \mathbf{0}, \\ & \delta_i \geq \delta_{i+1} \quad \forall 2 \leq i \leq q-1, \quad \delta_1 \geq 0,\ \delta_q \geq 0,\ \sum_{i=1}^{q} \delta_i = 1.\end{aligned} \tag{9}$$

5.1 An unsuccessful attempt

We first discuss a naive attempt to simplify the optimization that is not fruitful. Consider the inner optimization over $\boldsymbol{\Delta}$ in the above optimization problem (9):

$$\begin{aligned}\min_{\boldsymbol{\Delta}} \; & -\frac{1}{2}\sum_{i=1}^{q} \delta_i \boldsymbol{\gamma}^\top \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \boldsymbol{\gamma} \\ \text{s.t. } & \delta_i \geq \delta_{i+1} \quad \forall 2 \leq i \leq q-1, \quad \delta_1 \geq 0, \quad \delta_q \geq 0, \quad \sum_{i=1}^{q} \delta_i = 1.\end{aligned} \tag{10}$$

Lemma 1. *The dual of the above formulation is:*

$$\max_{\tau,\boldsymbol{\lambda}} \; -\tau \quad s.t. \qquad \frac{1}{2}\boldsymbol{\gamma}^\top \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \boldsymbol{\gamma} = \lambda_{i-1} - \lambda_i + \tau,\ \lambda_i \geq 0 \quad \forall 1 \leq i \leq q.$$

where $\lambda_0 = 0$ is a dummy variable.

Proof. Start by writing the Lagrangian of the optimization problem:

$$\mathcal{L} = -\frac{1}{2}\sum_{i=1}^{q} \delta_i \boldsymbol{\gamma}^\top \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \boldsymbol{\gamma} - \sum_{i=2}^{q-1} \lambda_i(\delta_i - \delta_{i+1}) - \lambda_q \delta_q - \lambda_1 \delta_1 + \tau(\sum_{i=1}^{q} \delta_i - 1),$$

where $\lambda_i \geq 0$ and τ are Lagrange multipliers. The dual follows after differentiating $\mathcal{L}$ with respect to δ_i and equating the resulting expression to zero. □

Caveat While the above dual is independent of δ's, the constraints $\frac{1}{2}\boldsymbol{\gamma}^\top \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \boldsymbol{\gamma} = \lambda_{i-1} - \lambda_i + \tau$ involve a quadratic term in an equality. It is not possible to simply leave out λ_i to make this constraint an inequality since the same λ_i occurs in two equations. This is non-convex in $\boldsymbol{\gamma}$ and is problematic since, after all, we eventually want an optimization problem that is jointly convex in $\boldsymbol{\gamma}$ and the other variables. Thus, a reformulation is necessary to pose relative margin kernel learning as a jointly convex optimization problem.

[9] It is trivial to construct such $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\eta}$ and $\boldsymbol{\Delta}$ when not all the labels are the same.

5.2 A refined approach

We proceed by instead considering the following optimization problem:

$$\min_{\boldsymbol{\Delta}} -\frac{1}{2}\sum_{i=1}^{q}\delta_i\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_i\bar{\mathbf{u}}_i^\top\boldsymbol{\gamma} \tag{11}$$

$$\text{s.t. } \delta_i-\delta_{i+1}\geq\epsilon \quad \forall 2\leq i\leq q-1, \quad \delta_1\geq\epsilon, \quad \delta_q\geq\epsilon, \quad \sum_{i=1}^{q}\delta_i=1$$

where we still maintain the ordering of the eigenvalues but require that they are separated by at least ϵ. Note that $\epsilon>0$ is not like other typical machine learning algorithm parameters (such as the parameter C in SVMs), since it can be *arbitrarily* small. The only requirement here is that ϵ remains positive. Thus, we are not really adding an extra parameter to the algorithm in posing it as a QCQP. The following theorem shows that a change of variables can be done in the above optimization problem so that its dual is in a particularly convenient form; *note, however, that directly deriving the dual of* (11) *fails to give the desired property and form.*

Theorem 2. *The dual of the optimization problem* (11) *is:*

$$\max_{\boldsymbol{\lambda}\geq 0,\tau} \quad -\tau+\epsilon\sum_{i=1}^{q}\lambda_i \tag{12}$$

$$s.t. \ \frac{1}{2}\boldsymbol{\gamma}^\top\sum_{j=2}^{i}\bar{\mathbf{u}}_j\bar{\mathbf{u}}_j^\top\boldsymbol{\gamma}=\tau(i-1)-\lambda_i \qquad \forall 2\leq i\leq q$$

$$\frac{1}{2}\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_1\bar{\mathbf{u}}_1^\top\boldsymbol{\gamma}=\tau-\lambda_1.$$

Proof. Start with the following change of variables:

$$\kappa_i:=\begin{cases}\delta_1 & \text{for } i=1,\\ \delta_i-\delta_{i+1} & \text{for } 2\leq i\leq q-1,\\ \delta_q & \text{for } i=q.\end{cases}$$

This gives:

$$\delta_i=\begin{cases}\kappa_1 & \text{for } i=1,\\ \sum_{j=i}^{q}\kappa_j & \text{for } 2\leq i\leq q.\end{cases} \tag{13}$$

Thus, (11) can be stated as,

$$\min_{\kappa} -\frac{1}{2}\sum_{i=2}^{q}\sum_{j=i}^{q}\kappa_j\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_i\bar{\mathbf{u}}_i^\top\boldsymbol{\gamma}+\kappa_1\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_1\bar{\mathbf{u}}_1^\top\boldsymbol{\gamma} \tag{14}$$

$$\text{s.t. } \kappa_i\geq\epsilon \quad \forall 1\leq i\leq q, \text{ and } \sum_{i=2}^{q}\sum_{j=i}^{q}\kappa_j+\kappa_1=1.$$

Consider simplifying the following term within the above formulation:

$$\sum_{i=2}^{q}\sum_{j=i}^{q}\kappa_j\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_i\bar{\mathbf{u}}_i^\top\boldsymbol{\gamma}=\sum_{i=2}^{q}\kappa_i\sum_{j=2}^{i}\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_j\bar{\mathbf{u}}_j^\top\boldsymbol{\gamma}\quad\text{and}\quad\sum_{i=2}^{q}\sum_{j=i}^{q}\kappa_j=\sum_{i=2}^{q}(i-1)\kappa_i.$$

It is now straightforward to write the Lagrangian to obtain the dual. □

Even though the above optimization appears to have non-convexity problems mentioned after Lemma 1, these can be avoided. This is facilitated by the following helpful property.

Lemma 2. *For $\epsilon > 0$, all the inequality constraints are active at the optimum of the following optimization problem:*

$$\begin{aligned}\max_{\boldsymbol{\lambda}\geq\mathbf{0},\tau}\quad&-\tau+\epsilon\sum_{i=1}^{q}\lambda_i\\ s.t.\ \ &\frac{1}{2}\boldsymbol{\gamma}^\top\sum_{j=2}^{i}\bar{\mathbf{u}}_j\bar{\mathbf{u}}_j^\top\boldsymbol{\gamma}\leq\tau(i-1)-\lambda_i\qquad\forall 2\leq i\leq q\\ &\frac{1}{2}\boldsymbol{\gamma}^\top\bar{\mathbf{u}}_1\bar{\mathbf{u}}_1^\top\boldsymbol{\gamma}\leq\tau-\lambda_1.\end{aligned}\tag{15}$$

Proof. Assume that $\boldsymbol{\lambda}^*$ is the optimum for the above problem and constraint i (corresponding to λ_i) is not active. Then, clearly, the objective can be further maximized by increasing λ_i^*. This contradicts the fact that $\boldsymbol{\lambda}^*$ is the optimum. □

In fact, it is not hard to show that the Lagrange multipliers of the constraints in problem (15) are equal to the κ_i's. Thus, replacing the inner optimization over δ's in (9), by (15), we get the following optimization problem, which we call STORM (Spectrum Transformations that Optimize the Relative Margin):

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta},\boldsymbol{\lambda},\tau}\quad&\boldsymbol{\alpha}^\top\mathbf{1}-\tau+\epsilon\sum_{i=1}^{q}\lambda_i-B\left(\boldsymbol{\beta}^\top\mathbf{1}+\boldsymbol{\eta}^\top\mathbf{1}\right)\\ \text{s.t. }&\frac{1}{2}(\mathbf{Y}\boldsymbol{\alpha}-\boldsymbol{\beta}+\boldsymbol{\eta})^\top\sum_{j=2}^{i}\bar{\mathbf{u}}_j\bar{\mathbf{u}}_j^\top(\mathbf{Y}\boldsymbol{\alpha}-\boldsymbol{\beta}+\boldsymbol{\eta})\leq(i-1)\tau-\lambda_i\quad\forall 2\leq i\leq q\\ &\frac{1}{2}(\mathbf{Y}\boldsymbol{\alpha}-\boldsymbol{\beta}+\boldsymbol{\eta})^\top\bar{\mathbf{u}}_1\bar{\mathbf{u}}_1^\top(\mathbf{Y}\boldsymbol{\alpha}-\boldsymbol{\beta}+\boldsymbol{\eta})\leq\tau-\lambda_1\\ &\boldsymbol{\alpha}^\top\mathbf{y}-\boldsymbol{\beta}^\top\mathbf{1}+\boldsymbol{\eta}^\top\mathbf{1}=0,\ \mathbf{0}\leq\boldsymbol{\alpha}\leq C\mathbf{1},\ \boldsymbol{\beta}\geq\mathbf{0},\ \boldsymbol{\eta}\geq\mathbf{0},\ \boldsymbol{\lambda}\geq\mathbf{0}.\end{aligned}\tag{16}$$

The above optimization problem has a linear objective with quadratic constraints. This equation now falls into the well-known family of quadratically constrained quadratic optimization (QCQP) problems whose solution is straightforward in practice. Thus, we have proposed a novel QCQP for large relative margin spectrum learning. Since the relative margin machine is strictly more general than the support vector machine, we obtain STOAM (Spectrum Transformations that Optimize the Absolute Margin) by simply setting $B = \infty$.

Obtaining $\boldsymbol{\delta}$ values Interior point methods obtain both primal and dual solutions of an optimization problem simultaneously. We can use equation (13) to obtain the weight on each eigenvector to construct the kernel.

Computational complexity STORM is a standard QCQP with q quadratic constraints of dimensionality l. This can be solved in time $\mathcal{O}(ql^3)$ with an interior point solver. We point out that, typically, the number of labeled examples l is much smaller than the total number of examples (which is n). Moreover, q is typically a fixed constant. Thus the runtime of the proposed QCQP compares favorably with the $\mathcal{O}(n^3)$ time for the initial eigendecomposition of $\mathbf{L}$ which is required for all the spectral methods described in this paper.

6 Experiments

To study the empirical performance of STORM and STOAM with respect to previous work, we performed experiments on both text and digit classification problems. Five binary classification problems were chosen from the 20-newsgroups text dataset (separating categories like baseball-hockey (b-h), pc-mac (p-m), religion-atheism (r-a), windows-xwindows (w-x), and politics.mideast-politics.misc (m-m)). Similarly, five different problems were considered from the MNIST dataset (separating digits 0-9, 1-2, 3-8, 4-7, and 5-6). One thousand randomly sampled examples were used for each task.

A mutual nearest neighbor graph was first constructed using five nearest neighbors and then the graph Laplacian was computed. The elements of the weight matrix $\mathbf{W}$ were all binary. In the case of MNIST digits, raw pixel values (note that each feature was normalized to zero-mean and unit variance) were used as features. For digits, nearest neighbors were determined by Euclidean distance, whereas, for text, the cosine similarity and tf-idf was used. In the experiments, the number of eigenvalues q was set to 200. This was a uniform choice for all methods which would not yield any unfair advantages for one approach over any other. In the case of STORM and STOAM, ϵ was set to a negligible value of 10^{-6}.

The entire dataset was randomly divided into labeled and unlabeled examples. The number of labeled examples was varied in steps of 20; the rest of the examples served as the test examples (as well as the unlabeled examples in graph construction). We then ran KTA to obtain a kernel; the estimated kernel was then fed into an SVM (this was referred to as KTA-S in the Tables) as well as to an RMM (referred to as KTA-R). To get an idea of the extent to which the ordering constraints matter, we also ran multiple kernel learning optimization which are similar to STOAM and STORM but without any ordering constraints. We refer to the multiple kernel learning with the SVM objective as MKL-S and with the RMM objective as MKL-R. We also included the spectral graph transducer (SGT) and the approach of [9] (described in the Appendix) in the experiments. Predictions on all the unlabeled examples were obtained for all the methods. Error rates were evaluated on the unlabeled examples. Twenty such runs were

done for various values of hyper-parameters (such as C,B) for all the methods. The values of the hyper-parameters that resulted in minimum average error rate over unlabeled examples were selected for all the approaches. Once the hyper-parameter values were fixed, the entire dataset was again divided into labeled and unlabeled examples. Training was then done but with fixed values of various hyper-parameters. Error rates on unlabeled examples were then obtained for all the methods over hundred runs of random splits of the dataset.

DATA	l	[9]	MKL-S	MKL-R	SGT	KTA-S	KTA-R	STOAM	STORM
r-a	30	$44.89_{\pm 5.2}$	$37.14_{\pm 5.6}$	$37.14_{\pm 5.6}$	$19.46_{\pm 1.4}$	$22.98_{\pm 4.8}$	$22.99_{\pm 4.8}$	$25.81_{\pm 6.1}$	$25.81_{\pm 6.1}$
	50	$42.18_{\pm 3.8}$	$29.93_{\pm 5.1}$	$30.01_{\pm 5.2}$	$18.92_{\pm 1.1}$	$19.87_{\pm 3.1}$	$19.87_{\pm 3.1}$	$21.49_{\pm 4.0}$	$21.49_{\pm 4.0}$
	70	$40.15_{\pm 2.5}$	$25.18_{\pm 4.4}$	$25.43_{\pm 4.3}$	$18.44_{\pm 1.0}$	$18.30_{\pm 2.4}$	$18.30_{\pm 2.4}$	$18.48_{\pm 3.1}$	$18.48_{\pm 3.1}$
	90	$38.86_{\pm 2.5}$	$22.33_{\pm 3.3}$	$22.67_{\pm 3.3}$	$18.22_{\pm 0.9}$	$17.32_{\pm 1.5}$	$17.32_{\pm 1.5}$	$17.21_{\pm 1.8}$	$17.23_{\pm 1.9}$
	110	$37.74_{\pm 2.3}$	$20.43_{\pm 2.4}$	$20.41_{\pm 2.4}$	$18.10_{\pm 1.0}$	$16.46_{\pm 1.3}$	$16.46_{\pm 1.3}$	$16.40_{\pm 1.2}$	$16.41_{\pm 1.2}$
w-m	30	$46.98_{\pm 2.4}$	$22.74_{\pm 8.7}$	$22.74_{\pm 8.7}$	$41.88_{\pm 8.5}$	$16.03_{\pm 8.8}$	$16.08_{\pm 8.8}$	$14.26_{\pm 5.9}$	$14.26_{\pm 5.9}$
	50	$45.47_{\pm 3.5}$	$15.08_{\pm 3.8}$	$15.08_{\pm 3.8}$	$35.63_{\pm 9.3}$	$13.54_{\pm 3.4}$	$13.56_{\pm 3.4}$	$11.49_{\pm 3.4}$	$11.52_{\pm 3.4}$
	70	$43.62_{\pm 4.0}$	$13.03_{\pm 1.6}$	$13.04_{\pm 1.6}$	$29.03_{\pm 7.8}$	$12.75_{\pm 4.8}$	$12.89_{\pm 5.0}$	$10.72_{\pm 0.9}$	$10.76_{\pm 1.0}$
	90	$42.85_{\pm 3.6}$	$12.20_{\pm 1.6}$	$12.20_{\pm 1.6}$	$22.55_{\pm 6.3}$	$11.30_{\pm 1.5}$	$11.41_{\pm 1.7}$	$10.43_{\pm 0.6}$	$10.43_{\pm 0.6}$
	110	$41.91_{\pm 3.8}$	$11.84_{\pm 1.0}$	$11.85_{\pm 1.0}$	$18.16_{\pm 5.0}$	$10.87_{\pm 1.4}$	$10.99_{\pm 1.7}$	$10.31_{\pm 0.6}$	$10.28_{\pm 0.6}$
p-m	30	$46.48_{\pm 2.7}$	$41.21_{\pm 4.9}$	$40.99_{\pm 5.0}$	$39.58_{\pm 3.8}$	$28.00_{\pm 5.8}$	$28.05_{\pm 5.8}$	$30.58_{\pm 6.6}$	$30.58_{\pm 6.6}$
	50	$44.08_{\pm 3.5}$	$35.98_{\pm 5.3}$	$35.94_{\pm 4.9}$	$37.46_{\pm 3.8}$	$24.34_{\pm 4.8}$	$24.34_{\pm 4.8}$	$25.72_{\pm 4.6}$	$25.72_{\pm 4.6}$
	70	$42.05_{\pm 3.5}$	$31.48_{\pm 4.6}$	$31.18_{\pm 4.3}$	$35.52_{\pm 3.4}$	$22.14_{\pm 3.6}$	$22.14_{\pm 3.6}$	$22.33_{\pm 4.9}$	$22.33_{\pm 4.9}$
	90	$39.54_{\pm 3.2}$	$28.15_{\pm 3.8}$	$28.30_{\pm 3.8}$	$33.57_{\pm 3.4}$	$20.58_{\pm 2.8}$	$20.59_{\pm 2.7}$	$20.44_{\pm 3.0}$	$20.77_{\pm 3.2}$
	110	$38.10_{\pm 3.2}$	$25.88_{\pm 3.1}$	$26.16_{\pm 2.9}$	$32.16_{\pm 3.2}$	$19.53_{\pm 2.2}$	$19.56_{\pm 2.2}$	$19.74_{\pm 2.4}$	$19.70_{\pm 2.4}$
b-h	30	$47.04_{\pm 2.1}$	$4.35_{\pm 0.8}$	$4.35_{\pm 0.8}$	$3.95_{\pm 0.2}$	$3.91_{\pm 0.4}$	$3.80_{\pm 0.3}$	$3.90_{\pm 0.3}$	$3.87_{\pm 0.3}$
	50	$46.11_{\pm 2.2}$	$3.90_{\pm 0.1}$	$3.91_{\pm 0.1}$	$3.93_{\pm 0.2}$	$3.81_{\pm 0.3}$	$3.80_{\pm 0.4}$	$3.87_{\pm 0.3}$	$3.73_{\pm 0.3}$
	70	$45.92_{\pm 2.4}$	$3.91_{\pm 0.2}$	$3.90_{\pm 0.2}$	$3.90_{\pm 0.2}$	$3.76_{\pm 0.3}$	$3.76_{\pm 0.3}$	$3.78_{\pm 0.3}$	$3.68_{\pm 0.3}$
	90	$45.30_{\pm 2.5}$	$3.88_{\pm 0.2}$	$3.89_{\pm 0.2}$	$3.85_{\pm 0.3}$	$3.69_{\pm 0.3}$	$3.67_{\pm 0.3}$	$3.75_{\pm 0.3}$	$3.61_{\pm 0.3}$
	110	$44.99_{\pm 2.6}$	$3.88_{\pm 0.2}$	$3.88_{\pm 0.2}$	$3.83_{\pm 0.3}$	$3.71_{\pm 0.4}$	$3.66_{\pm 0.3}$	$3.67_{\pm 0.3}$	$3.56_{\pm 0.3}$
m-m	30	$48.11_{\pm 4.7}$	$12.35_{\pm 5.2}$	$12.35_{\pm 5.2}$	$41.30_{\pm 3.5}$	$7.35_{\pm 3.6}$	$7.36_{\pm 3.8}$	$7.60_{\pm 3.9}$	$6.88_{\pm 2.9}$
	50	$46.36_{\pm 3.3}$	$7.47_{\pm 3.1}$	$7.25_{\pm 2.9}$	$31.18_{\pm 7.5}$	$6.25_{\pm 2.8}$	$6.19_{\pm 2.9}$	$5.45_{\pm 1.0}$	$5.39_{\pm 1.2}$
	70	$45.31_{\pm 5.7}$	$6.05_{\pm 1.3}$	$5.98_{\pm 1.4}$	$22.30_{\pm 7.5}$	$5.43_{\pm 1.0}$	$5.35_{\pm 1.1}$	$5.20_{\pm 0.7}$	$4.90_{\pm 0.6}$
	90	$42.52_{\pm 5.0}$	$5.71_{\pm 1.0}$	$5.68_{\pm 1.0}$	$15.39_{\pm 5.9}$	$5.13_{\pm 0.9}$	$5.14_{\pm 1.1}$	$5.09_{\pm 0.6}$	$4.76_{\pm 0.6}$
	110	$41.94_{\pm 5.2}$	$5.44_{\pm 0.7}$	$5.16_{\pm 0.6}$	$10.96_{\pm 3.9}$	$4.97_{\pm 0.8}$	$4.92_{\pm 0.9}$	$4.95_{\pm 0.5}$	$4.65_{\pm 0.5}$

Table 1. Mean and std. deviation of percentage error rates on text datasets. In each row, the method with minimum error rate is shown in dark gray. All the other algorithms whose performance is not significantly different from the best (at 5% significance level by a paired t-test) are shown in light gray.

The results are presented in Table 1 and Table 2. It can be seen that STORM and STOAM perform much better than all the methods. Results in the two tables are further summarized in Table 3. It can be seen that both STORM and STOAM have significant advantages over all the other methods. Moreover, the

formulation of [9] gives very poor results since the learned spectrum is independent of $\boldsymbol{\alpha}$.

DATA	l	[9]	MKL-S	MKL-R	SGT	KTA-S	KTA-R	STOAM	STORM
0-9	30	$46.45_{\pm 1.5}$	$0.89_{\pm 0.1}$	$0.89_{\pm 0.1}$	$0.83_{\pm 0.1}$	$0.90_{\pm 0.1}$	$0.90_{\pm 0.1}$	$0.88_{\pm 0.1}$	$0.88_{\pm 0.1}$
	50	$45.83_{\pm 1.9}$	$0.89_{\pm 0.1}$	$0.90_{\pm 0.1}$	$0.85_{\pm 0.1}$	$0.91_{\pm 0.1}$	$0.91_{\pm 0.1}$	$0.89_{\pm 0.1}$	$0.89_{\pm 0.1}$
	70	$45.55_{\pm 2.0}$	$0.88_{\pm 0.1}$	$0.87_{\pm 0.1}$	$0.87_{\pm 0.1}$	$0.89_{\pm 0.1}$	$0.93_{\pm 0.2}$	$0.88_{\pm 0.1}$	$0.88_{\pm 0.1}$
	90	$45.68_{\pm 1.6}$	$0.90_{\pm 0.1}$	$0.85_{\pm 0.2}$	$0.86_{\pm 0.1}$	$0.91_{\pm 0.2}$	$0.91_{\pm 0.2}$	$0.87_{\pm 0.1}$	$0.86_{\pm 0.1}$
	110	$45.40_{\pm 2.0}$	$0.85_{\pm 0.2}$	$0.90_{\pm 0.2}$	$0.87_{\pm 0.1}$	$0.89_{\pm 0.1}$	$0.89_{\pm 0.1}$	$0.92_{\pm 0.3}$	$0.86_{\pm 0.1}$
1-2	30	$47.22_{\pm 2.0}$	$3.39_{\pm 3.3}$	$4.06_{\pm 5.9}$	$11.81_{\pm 6.8}$	$2.92_{\pm 0.6}$	$2.92_{\pm 0.6}$	$2.88_{\pm 0.5}$	$2.85_{\pm 0.4}$
	50	$46.02_{\pm 2.0}$	$2.85_{\pm 0.5}$	$2.58_{\pm 0.4}$	$3.57_{\pm 2.7}$	$2.78_{\pm 0.4}$	$2.84_{\pm 0.5}$	$2.80_{\pm 0.7}$	$2.80_{\pm 0.7}$
	70	$45.56_{\pm 2.4}$	$2.64_{\pm 0.3}$	$2.34_{\pm 0.3}$	$2.72_{\pm 0.5}$	$2.74_{\pm 0.3}$	$2.76_{\pm 0.4}$	$2.61_{\pm 0.3}$	$2.70_{\pm 0.3}$
	90	$45.00_{\pm 2.7}$	$2.71_{\pm 0.3}$	$2.35_{\pm 0.3}$	$2.60_{\pm 0.2}$	$2.76_{\pm 0.3}$	$2.73_{\pm 0.4}$	$2.70_{\pm 0.4}$	$2.70_{\pm 0.3}$
	110	$44.97_{\pm 2.3}$	$2.77_{\pm 0.3}$	$2.36_{\pm 0.3}$	$2.61_{\pm 0.2}$	$2.61_{\pm 0.6}$	$2.61_{\pm 0.6}$	$2.51_{\pm 0.3}$	$2.51_{\pm 0.3}$
3-8	30	$45.42_{\pm 3.0}$	$13.02_{\pm 3.7}$	$12.63_{\pm 3.6}$	$9.86_{\pm 0.9}$	$8.54_{\pm 2.7}$	$7.58_{\pm 2.2}$	$7.93_{\pm 2.2}$	$7.68_{\pm 1.8}$
	50	$43.72_{\pm 3.0}$	$9.54_{\pm 2.3}$	$9.04_{\pm 2.2}$	$8.76_{\pm 0.9}$	$6.93_{\pm 1.8}$	$6.61_{\pm 1.6}$	$6.42_{\pm 1.5}$	$6.37_{\pm 1.4}$
	70	$42.77_{\pm 3.1}$	$7.98_{\pm 2.1}$	$7.39_{\pm 1.7}$	$8.00_{\pm 0.8}$	$6.31_{\pm 1.6}$	$6.07_{\pm 1.4}$	$5.85_{\pm 1.3}$	$5.85_{\pm 1.1}$
	90	$41.28_{\pm 3.4}$	$7.02_{\pm 1.6}$	$6.60_{\pm 1.3}$	$7.33_{\pm 0.8}$	$5.69_{\pm 1.1}$	$5.69_{\pm 1.1}$	$5.45_{\pm 1.0}$	$5.40_{\pm 0.9}$
	110	$41.09_{\pm 3.5}$	$6.56_{\pm 1.2}$	$6.15_{\pm 1.0}$	$6.91_{\pm 0.9}$	$5.35_{\pm 0.9}$	$5.43_{\pm 0.9}$	$5.25_{\pm 0.8}$	$5.24_{\pm 0.9}$
4-7	30	$44.85_{\pm 3.5}$	$5.74_{\pm 3.4}$	$5.54_{\pm 3.3}$	$5.60_{\pm 1.2}$	$4.27_{\pm 1.9}$	$4.09_{\pm 1.9}$	$3.64_{\pm 1.4}$	$3.57_{\pm 1.1}$
	50	$43.65_{\pm 3.3}$	$4.31_{\pm 1.2}$	$3.97_{\pm 0.9}$	$4.50_{\pm 0.5}$	$3.50_{\pm 0.9}$	$3.40_{\pm 0.8}$	$3.24_{\pm 0.7}$	$3.17_{\pm 0.6}$
	70	$44.05_{\pm 3.3}$	$3.66_{\pm 0.8}$	$3.31_{\pm 0.6}$	$4.04_{\pm 0.4}$	$3.38_{\pm 0.8}$	$3.23_{\pm 0.7}$	$3.11_{\pm 0.6}$	$3.04_{\pm 0.5}$
	90	$42.04_{\pm 3.3}$	$3.46_{\pm 0.8}$	$3.13_{\pm 0.6}$	$3.77_{\pm 0.4}$	$3.12_{\pm 0.6}$	$3.00_{\pm 0.6}$	$2.92_{\pm 0.5}$	$2.89_{\pm 0.5}$
	110	$41.85_{\pm 3.1}$	$3.28_{\pm 0.7}$	$3.00_{\pm 0.5}$	$3.60_{\pm 0.4}$	$2.99_{\pm 0.6}$	$2.98_{\pm 0.6}$	$2.92_{\pm 0.5}$	$2.91_{\pm 0.5}$
5-6	30	$46.75_{\pm 2.6}$	$5.18_{\pm 2.7}$	$4.91_{\pm 3.2}$	$2.49_{\pm 0.2}$	$3.48_{\pm 1.3}$	$3.32_{\pm 1.1}$	$3.19_{\pm 1.4}$	$2.96_{\pm 0.9}$
	50	$45.98_{\pm 3.1}$	$3.30_{\pm 1.3}$	$2.93_{\pm 0.8}$	$2.46_{\pm 0.2}$	$2.94_{\pm 0.7}$	$2.86_{\pm 0.5}$	$2.73_{\pm 0.4}$	$2.67_{\pm 0.4}$
	70	$45.75_{\pm 3.5}$	$2.80_{\pm 0.5}$	$2.62_{\pm 0.3}$	$2.49_{\pm 0.2}$	$2.70_{\pm 0.4}$	$2.65_{\pm 0.4}$	$2.63_{\pm 0.3}$	$2.83_{\pm 0.6}$
	90	$45.19_{\pm 3.8}$	$2.68_{\pm 0.3}$	$2.60_{\pm 0.3}$	$2.49_{\pm 0.2}$	$2.62_{\pm 0.4}$	$2.60_{\pm 0.4}$	$2.60_{\pm 0.3}$	$2.52_{\pm 0.4}$
	110	$43.59_{\pm 2.8}$	$2.62_{\pm 0.3}$	$2.52_{\pm 0.3}$	$2.51_{\pm 0.2}$	$2.57_{\pm 0.4}$	$2.53_{\pm 0.4}$	$2.55_{\pm 0.4}$	$2.49_{\pm 0.4}$

Table 2. Mean and std. deviation of percentage error rates on digits datasets. In each row, the method with minimum error rate is shown in dark gray. All the other algorithms whose performance is not significantly different from the best (at 5% significance level by a paired t-test) are shown in light gray.

To gain further intuition, we visualized the learned spectrum in each problem to see if the algorithms yield significant differences in spectra. We present four typical plots in Figure 1. We show the spectra obtained by KTA, STORM and MKL-R (the difference between the spectra obtained by STOAM (MKL-S) was much closer to that obtained by STORM (MKL-R) compared to other methods). Typically KTA puts significantly more weight on the top few eigenvectors. By not maintaining the order among the eigenvectors, MKL seems to put haphazard weights on the eigenvectors. However, STORM is less aggressive and its eigenspectrum decays at a slower rate. This shows that STORM obtains a markedly different spectrum compared to KTA and MKL and is recovering a qualitatively different kernel. It is important to point out that MKL-R (MKL-S)

solves a more general problem than STORM (STOAM). Thus, it can always achieve a better objective value compared to STORM (STOAM). However, this causes over-fitting and the experiments show that the error rate on the unlabeled examples actually increases when the order of the spectrum is not preserved. In fact, MKL obtained competitive results in only one case (digits:1-2) which could be attributed to chance.

	[9]	MKL-S	MKL-R	SGT	KTA-S	KTA-R	STOAM	STORM
#dark gray	0	1	5	9	5	2	8	22
#light gray	0	1	2	4	8	12	16	13
#total	0	2	7	13	13	14	24	35

Table 3. Summary of results in Tables 1 & 2. For each method, we enumerate the number of times it performed best (dark gray), the number of times it was not significantly worse than the best performing method (light gray) and the total number of times it was either best or not significantly worse from best.

7 Conclusions

We proposed a large relative margin formulation for transforming the eigenspectrum of a graph Laplacian. A family of kernels was explored which maintains smoothness properties on the graph by enforcing an ordering on the eigenvalues of the kernel matrix. Unlike the previous methods which used two distinct criteria at each phase of the learning process, we demonstrated how jointly optimizing the spectrum of a Laplacian while learning a classifier can result in improved performance. The resulting kernels, learned as part of the optimization, showed improvements on a variety of experiments. The formulation (3) shows that we can learn predictions as well as the spectrum of a Laplacian jointly by convex programming. This opens up an interesting direction for further investigation. By learning weights on an appropriate number of matrices, it is possible to explore all graph Laplacians. Thus, it seems possible to learn both a graph structure and a large (relative) margin solution jointly.

Acknowledgments The authors acknowledge support from DHS Contract N66001-09-C-0080—"Privacy Preserving Sharing of Network Trace Data (PPSNTD) Program" and "NetTrailMix" Google Research Award.

References

1. F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *International Conference on Machine Learning*, 2004.
2. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

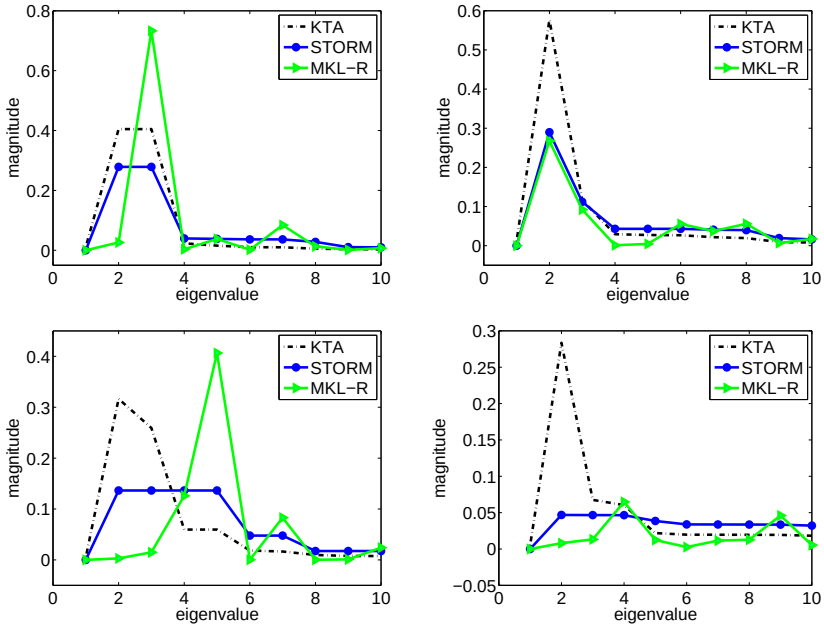

Fig. 1. Magnitudes of the top 15 eigenvalues recovered by the different algorithms. Top: problems 1-2 and 3-8. Bottom: m-m and p-m. The plots show average eigenspectra over all runs for each problem.

3. N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola. On kernel-target alignment. In *NIPS*, pages 367–373, 2001.
4. T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, pages 290–297, 2003.
5. R. I. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, pages 315–322, 2002.
6. B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
7. P. Shivaswamy and T. Jebara. Maximum relative margin and data-dependent regularization. *Journal of Machine Learning Research*, 11:747–788, 2010.
8. A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT*, pages 144–158, 2003.
9. Z. Xu, J. Zhu, M. R. Lyu, and I. King. Maximum margin based semi-supervised spectral kernel learning. In *IJCNN*, pages 418–423, 2007.
10. T. Zhang and R. Ando. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*, pages 1601–1608, 2006.
11. X. Zhu, J. S. Kandola, Z. Ghahramani, and J. D. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*, 2004.
12. X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, Carnegie Mellon University, 2003.

A Approach of Xu *et al.* [9]

It is important to note that, in a previously published article [9], other authors attempted to solve a problem related to STOAM. While this section is not the main focus of our paper, it is helpful to point out that the method in [9] is completely different from our formulation and contains serious flaws. The previous approach attempted to learn a kernel of the form $\mathbf{K} = \sum_{i=1}^{q} \delta_i \mathbf{u}_i \mathbf{u}_i^\top$ while maximizing the margin in the SVM dual. They start with the problem (Equation (13) in [9] but using our notation):

$$\max_{\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \boldsymbol{\alpha}^\top \mathbf{y} = 0} \boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{Y} \mathbf{K}^{\text{tr}} \mathbf{Y} \boldsymbol{\alpha} \tag{17}$$

$$\text{s.t } \delta_i \geq w\delta_{i+1} \quad \forall 1 \leq i \leq q-1, \quad \delta_i \geq 0, \quad \mathbf{K} = \sum_{i=1}^{q} \delta_i \mathbf{u}_i \mathbf{u}_i^\top, \quad \text{trace}(\mathbf{K}) = \mu$$

which is the SVM dual with a particular choice of kernel. Here $\mathbf{K}^{\text{tr}} = \sum_{i=1}^{q} \delta_i \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top$. It is assumed that μ, w and C are fixed parameters. The authors discuss optimizing the above problem while exploring $\mathbf{K}$ by adjusting the δ_i values. The authors then claim, without proof, that the following QCQP (Equation (14) of [9]) can jointly optimize δ's while learning a classifier:

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\delta}, \rho} \; 2\boldsymbol{\alpha}^\top \mathbf{1} - \mu\rho \tag{18}$$

$$\text{s.t. } \mu = \sum_{i=1}^{q} \delta_i t_i, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \quad \boldsymbol{\alpha}^\top \mathbf{y} = 0, \quad \delta_i \geq \mathbf{0} \; \forall 1 \leq i \leq q$$

$$\frac{1}{t_i} \boldsymbol{\alpha}^\top \mathbf{Y} \bar{\mathbf{u}}_i \bar{\mathbf{u}}_i^\top \mathbf{Y} \boldsymbol{\alpha} \leq \rho \quad \forall 1 \leq i \leq q, \quad \delta_i \geq w\delta_{i+1} \quad \forall 1 \leq i \leq q-1$$

where t_i are *fixed* scalar values (whose values are irrelevant in this discussion). The only constraints on δ's are: non-negativity, $\delta_i \geq w\delta_{i+1}$, and $\sum_{i=1}^{q} \delta_i t_i = \mu$ where w and μ are fixed parameters. Clearly, in this problem, δ's can be set independently of $\boldsymbol{\alpha}$! Further, since μ is also a fixed constant, δ no longer has any effect on the objective. Thus, δ's can be set without affecting either the objective or the other variables ($\boldsymbol{\alpha}$ and ρ). Therefore, the formulation (18) certainly does not maximize the margin while learning the spectrum. This conclusion is further supported by empirical evidence in our experiments. Throughout all the experiments, the optimization problem proposed by [9] produced extremely weak results.