Semi-supervised Eigenvectors for Locally-biased Learning

Toke Jansen Hansen
Section for Cognitive Systems
DTU Informatics
Technical University of Denmark
tjha@imm.dtu.dk

Michael W. Mahoney
Department of Mathematics
Stanford University
Stanford, CA 94305
mmahoney@cs.stanford.edu

Abstract

In many applications, one has side information, *e.g.*, labels that are provided in a semi-supervised manner, about a specific target region of a large data set, and one wants to perform machine learning and data analysis tasks "nearby" that pre-specified target region. Locally-biased problems of this sort are particularly challenging for popular eigenvector-based machine learning and data analysis tools. At root, the reason is that eigenvectors are inherently global quantities. In this paper, we address this issue by providing a methodology to construct *semi-supervised eigenvectors* of a graph Laplacian, and we illustrate how these locally-biased eigenvectors can be used to perform *locally-biased machine learning*. These semi-supervised eigenvectors capture successively-orthogonalized directions of maximum variance, conditioned on being well-correlated with an input seed set of nodes that is assumed to be provided in a semi-supervised manner. We also provide several empirical examples demonstrating how these semi-supervised eigenvectors can be used to perform locally-biased learning.

1 Introduction

We consider the problem of finding a set of locally-biased vectors that inherit many of the "nice" properties that the leading nontrivial global eigenvectors of a graph Laplacian have—for example, that capture "slowly varying" modes in the data, that are fairly-efficiently computable, that can be used for common machine learning and data analysis tasks such as kernel-based and semi-supervised learning, etc.—so that we can perform what we will call *locally-biased machine learning* in a principled manner.

By *locally-biased machine learning*, we mean that we have a very large data set, *e.g.*, represented as a graph, and that we have information, *e.g.*, given in a semi-supervised manner, that certain "regions" of the data graph are of particular interest. In this case, we may want to focus predominantly on those regions and perform data analysis and machine learning, *e.g.*, classification, clustering, ranking, etc., that is "biased toward" those pre-specified regions. Examples of this include the following.

- *Locally-biased community identification.* In social and information network analysis, one might have a small "seed set" of nodes that belong to a cluster or community of interest [2, 13]; in this case, one might want to perform link or edge prediction, or one might want to "refine" the seed set in order to find other nearby members.
- *Locally-biased image segmentation.* In computer vision, one might have a large corpus of images along with a "ground truth" set of pixels as provided by a face detection algorithm [7, 14, 15]; in this case, one might want to segment entire heads from the background for all the images in the corpus in an automated manner.

- *Locally-biased neural connectivity analysis.* In functional magnetic resonance imaging applications, one might have small sets of neurons that "fire" in response to some external experimental stimulus [16]; in this case, one might want to analyze the subsequent temporal dynamics of stimulation of neurons that are "nearby," either in terms of connectivity topology or functional response.

These examples present considerable challenges for spectral techniques and traditional eigenvector-based methods. At root, the reason is that eigenvectors are inherently global quantities, thus limiting their applicability in situations where one is interested in very local properties of the data.

In this paper, we provide a methodology to construct what we will call *semi-supervised eigenvectors* of a graph Laplacian; and we illustrate how these locally-biased eigenvectors inherit many of the properties that make the leading nontrivial global eigenvectors of the graph Laplacian so useful in applications. To achieve this, we will formulate an optimization ansatz that is a variant of the usual global spectral graph partitioning optimization problem that includes a natural locality constraint as well as an orthogonality constraint, and we will iteratively solve this problem.

In more detail, assume that we are given as input a (possibly weighted) data graph $G = (V, E)$, an indicator vector s of a small "seed set" of nodes, a *correlation parameter* $\kappa \in [0, 1]$, and a positive integer k. Then, informally, we would like to construct k vectors that satisfy the following bicriteria: first, each of these k vectors is well-correlated with the input seed set; and second, those k vectors describe successively-orthogonalized directions of maximum variance, in a manner analogous to the leading k nontrivial global eigenvectors of the graph Laplacian. (We emphasize that the seed set s of nodes, the integer k, and the correlation parameter κ are part of the input; and thus they should be thought of as being available in a semi-supervised manner.) Somewhat more formally, our main algorithm, Algorithm 1 in Section 3, returns as output k semi-supervised eigenvectors; each of these is the solution to an optimization problem of the form of GENERALIZED LOCALSPECTRAL in Figure 1, and thus each "captures" (say) κ/k of the correlation with the seed set. Our main theoretical result states that these vectors define successively-orthogonalized directions of maximum variance, conditioned on being κ/k-well-correlated with an input seed set s; and that each of these k semi-supervised eigenvectors can be computed quickly as the solution to a system of linear equations.

From a technical perspective, the work most closely related to ours is that of Mahoney *et al.* [14]. The original algorithm of Mahoney *et al.* [14] introduced a methodology to construct a locally-biased version of the leading nontrivial eigenvector of a graph Laplacian and showed (theoretically and empirically in a social network analysis application) that the resulting vector could be used to partition a graph in a locally-biased manner. From this perspective, our extension incorporates a natural orthogonality constraint that successive vectors need to be orthogonal to previous vectors. Subsequent to the work of [14], [15] applied the algorithm of [14] to the problem of finding locally-biased cuts in a computer vision application. Similar ideas have also been applied somewhat differently. For example, [2] use locally-biased random walks, *e.g.*, short random walks starting from a small seed set of nodes, to find clusters and communities in graphs arising in Internet advertising applications; [13] used locally-biased random walks to characterize the local and global clustering structure of a wide range of social and information networks; [11] developed the Spectral Graph Transducer (SGT), that performs transductive learning via spectral graph partitioning. The objectives in both [11] and [14] are considered constrained eigenvalue problems, that can be solved by finding the smallest eigenvalue of an asymmetric generalized eigenvalue problem, but in practice this procedure can be highly unstable [8]. The SGT reduces the instabilities by performing all calculations in a subspace spanned by the d smallest eigenvectors of the graph Laplacian, whereas [14] perform a binary search, exploiting the monotonic relationship between a control parameter and the corresponding Lagrange multiplier.

In parallel, [3] and a large body of subsequent work including [6] used eigenvectors of the graph Laplacian to perform dimensionality reduction and data representation, in unsupervised and semi-supervised settings. Many of these methods have a natural interpretation in terms of kernel-based learning [18]. Many of these diffusion-based spectral methods also have a natural interpretation in terms of spectral ranking [21]. "Topic sensitive" and "personalized" versions of these spectral ranking methods have also been studied [9, 10]; and these were the motivation for diffusion-based methods to find locally-biased clusters in large graphs [19, 1, 14]. Our optimization ansatz is a generalization of the linear equation formulation of the PageRank procedure [17, 14, 21], and the solution involves Laplacian-based linear equation solving, which has been suggested as a primitive

of more general interest in large-scale data analysis [20]. Finally, the form of our optimization problem has similarities to other work in computer vision applications: *e.g.*, [23] and [7] find good conductance clusters subject to a set of linear constraints.

2 Background and Notation

Let $G = (V, E, w)$ be a connected undirected graph with $n = |V|$ vertices and $m = |E|$ edges, in which edge $\{i, j\}$ has non-negative weight w_{ij}. In the following, $A_G \in \mathbb{R}^{V \times V}$ will denote the adjacency matrix of G, while $D_G \in \mathbb{R}^{V \times V}$ will denote the diagonal degree matrix of G, *i.e.*, $D_G(i,i) = d_i = \sum_{\{i,j\} \in E} w_{ij}$, the weighted degree of vertex i. Moreover, for a set of vertices $S \subseteq V$ in a graph, the *volume of* S is $\mathrm{vol}(S) \stackrel{\text{def}}{=} \sum_{i \in S} d_i$. The Laplacian of G is defined as $L_G \stackrel{\text{def}}{=} D_G - A_G$. (This is also called the combinatorial Laplacian, in which case the normalized Laplacian of G is $\mathcal{L}_G \stackrel{\text{def}}{=} D_G^{-1/2} L_G D_G^{-1/2}$.)

The Laplacian is the symmetric matrix having quadratic form $x^T L_G x = \sum_{ij \in E} w_{ij}(x_i - x_j)^2$, for $x \in \mathbb{R}^V$. This implies that L_G is positive semidefinite and that the all-one vector $1 \in \mathbb{R}^V$ is the eigenvector corresponding to the smallest eigenvalue 0. The generalized eigenvalues of $L_G x = \lambda_i D_G x$ are $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$. We will use v_2 to denote smallest non-trivial eigenvector, *i.e.*, the eigenvector corresponding to λ_2; v_3 to denote the next eigenvector; and so on. Finally, for a matrix A, let A^+ denote its (uniquely defined) Moore-Penrose pseudoinverse. For two vectors $x, y \in \mathbb{R}^n$, and the degree matrix D_G for a graph G, we define the degree-weighted inner product as $x^T D_G y \stackrel{\text{def}}{=} \sum_{i=1}^n x_i y_i d_i$. In particular, if a vector x has unit norm, then $x^T D_G x = 1$. Given a subset of vertices $S \subseteq V$, we denote by 1_S the indicator vector of S in $\mathbb{R}^V$ and by 1 the vector in $\mathbb{R}^V$ having all entries set equal to 1.

3 Optimization Approach to Semi-supervised Eigenvectors

3.1 Motivation for the Program

Recall the optimization perspective on how one computes the leading nontrivial global eigenvectors of the normalized Laplacian $\mathcal{L}_G$. The first nontrivial eigenvector v_2 is the solution to the problem GLOBALSPECTRAL that is presented on the left of Figure 1. Equivalently, although GLOBALSPECTRAL is a non-convex optimization problem, strong duality holds for it and it's solution may be computed as v_2, the leading nontrivial generalized eigenvector of L_G. The next eigenvector v_3 is the solution to GLOBALSPECTRAL, augmented with the constraint that $x^T D_G v_2 = 0$; and in general the t^{th} generalized eigenvector of L_G is the solution to GLOBALSPECTRAL, augmented with the constraints that $x^T D_G v_i = 0$, for $i \in \{2, \ldots, t-1\}$. Clearly, this set of constraints and the constraint $x^T D_G 1 = 0$ can be written as $x^T D_G Q = 0$, where 0 is a $(t-1)$-dimensional all-zeros vector, and where Q is an $n \times (t-1)$ orthogonal matrix whose i^{th} column equals v_i (where $v_1 = 1$, the all-ones vector, is the first column of Q).

Also presented in Figure 1 is LOCALSPECTRAL, which includes a constraint requiring the solution to be well-correlated with an input seed set. This LOCALSPECTRAL optimization problem was introduced in [14], where it was shown that the solution to LOCALSPECTRAL may be interpreted as a locally-biased version of the second eigenvector of the Laplacian. In particular, although LOCALSPECTRAL is not convex, it's solution can be computed efficiently as the solution to a set of linear equations that generalize the popular Personalized PageRank procedure; in addition, by performing a sweep cut and appealing to a variant of Cheeger's inequality, this locally-biased eigenvector can be used to perform locally-biased spectral graph partitioning [14].

3.2 Our Main Algorithm

We will formulate the problem of computing semi-supervised vectors in terms of a primitive optimization problem of independent interest. Consider the GENERALIZED LOCALSPECTRAL optimization problem, as shown in Figure 1. For this problem, we are given a graph $G = (V, E)$, with associated Laplacian matrix L_G and diagonal degree matrix D_G; an indicator vector s of a small

GLOBALSPECTRAL	LOCALSPECTRAL	GENERALIZED LOCALSPECTRAL
minimize $x^T L_G x$	minimize $x^T L_G x$	minimize $x^T L_G x$
s.t $x^T D_G x = 1$	s.t $x^T D_G x = 1$	s.t $x^T D_G x = 1$
$x^T D_G 1 = 0$	$x^T D_G 1 = 0$	$x^T D_G Q = 0$
	$x^T D_G s \geq \sqrt{\kappa}$	$x^T D_G s \geq \sqrt{\kappa}$

Figure 1: Left: The usual GLOBALSPECTRAL partitioning optimization problem; the vector achieving the optimal solution is v_2, the leading nontrivial generalized eigenvector of L_G with respect to D_G. Middle: The LOCALSPECTRAL optimization problem, which was originally introduced in [14]; for $\kappa = 0$, this coincides with the usual global spectral objective, while for $\kappa > 0$, this produces solutions that are biased toward the seed vector s. Right: The GENERALIZED LOCALSPECTRAL optimization problem we introduce that includes both the locality constraint and a more general orthogonality constraint. Our main algorithm for computing semi-supervised eigenvectors will iteratively compute the solution to GENERALIZED LOCALSPECTRAL for a sequence of Q matrices. In all three cases, the optimization variable is $x \in \mathbb{R}^n$.

"seed set" of nodes; a *correlation parameter* $\kappa \in [0, 1]$; and an $n \times \nu$ constraint matrix Q that may be assumed to be an orthogonal matrix. We will assume (without loss of generality) that s is properly normalized and orthogonalized so that $s^T D_G s = 1$ and $s^T D_G 1 = 0$. While s can be a general unit vector orthogonal to 1, it may be helpful to think of s as the indicator vector of one or more vertices in V, corresponding to the target region of the graph.

In words, the problem GENERALIZED LOCALSPECTRAL asks us to find a vector $x \in \mathbb{R}^n$ that minimizes the variance $x^T L_G x$ subject to several constraints: that x is unit length; that x is orthogonal to the span of Q; and that x is $\sqrt{\kappa}$-well-correlated with the input seed set vector s. In our application of GENERALIZED LOCALSPECTRAL to the computation of semi-supervised eigenvectors, we will iteratively compute the solution to GENERALIZED LOCALSPECTRAL, updating Q to contain the already-computed semi-supervised eigenvectors. That is, to compute the first semi-supervised eigenvector, we let $Q = 1$, *i.e.*, the n-dimensional all-ones vector, which is the trivial eigenvector of L_G, in which case Q is an $n \times 1$ matrix; and to compute each subsequent semi-supervised eigenvector, we let the columns of Q consist of 1 and the other semi-supervised eigenvectors found in each of the previous iterations.

To show that GENERALIZED LOCALSPECTRAL is efficiently-solvable, note that it is a quadratic program with only one quadratic constraint and one linear equality constraint. In order to remove the equality constraint, which will simplify the problem, let's change variables by defining the $n \times (n-\nu)$ matrix F as $\{x : Q^T D_G x = 0\} = \{x : x = Fy\}$. That is, F is a span for the null space of Q^T; and we will take F to be an orthogonal matrix. Then, with respect to the y variable, GENERALIZED LOCALSPECTRAL becomes

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & y^T F^T L_G F y \\ \text{subject to} \quad & y^T F^T D_G F y = 1, \\ & y^T F^T D_G s \geq \sqrt{\kappa}. \end{aligned} \tag{1}$$

In terms of the variable x, the solution to this optimization problem is of the form

$$\begin{aligned} x^* &= cF \left(F^T \left(L_G - \gamma D_G\right) F\right)^+ F^T D_G s \\ &= c \left(FF^T \left(L_G - \gamma D_G\right) FF^T\right)^+ D_G s, \end{aligned} \tag{2}$$

for a normalization constant $c \in (0, \infty)$ and for some γ that depends on $\sqrt{\kappa}$. The second line follows from the first since F is an $n \times (n-\nu)$ orthogonal matrix. This so-called "S-procedure" is described in greater detail in Chapter 5 and Appendix B of [4]. The significance of this is that, although it is a non-convex optimization problem, the GENERALIZED LOCALSPECTRAL problem can be solved by solving a linear equation, in the form given in Eqn. (2).

Returning to our problem of computing semi-supervised eigenvectors, recall that, in addition to the input for the GENERALIZED LOCALSPECTRAL problem, we need to specify a positive integer k that indicates the number of vectors to be computed. In the simplest case, we would assume that

we would like the correlation to be "evenly distributed" across all k vectors, in which case we will require that each vector is $\sqrt{\kappa/k}$-well-correlated with the input seed set vector s; but this assumption can easily be relaxed, and thus Algorithm 1 is formulated more generally as taking a k-dimensional vector $\kappa = [\kappa_1, \ldots, \kappa_k]^T$ of correlation coefficients as input.

To compute the first semi-supervised eigenvector, we will let $Q = 1$, the all-ones vector, in which case the first nontrivial semi-supervised eigenvector is

$$x_1^* = c\,(L_G - \gamma_1 D_G)^+ D_G s, \tag{3}$$

where γ_1 is chosen to saturate the part of the correlation constraint along the first direction. (Note that the projections FF^T from Eqn. (2) are not present in Eqn. (3) since by design $s^T D_G 1 = 0$.) That is, to find the correct setting of γ_1, it suffices to perform a binary search over the possible values of γ_1 in the interval $(-\mathrm{vol}(G), \lambda_2(G))$ until the correlation constraint is satisfied, that is, until $(s^T D_G x)^2$ is sufficiently close to κ_1^2, see [8, 14].

To compute subsequent semi-supervised eigenvectors, *i.e.*, at steps $t = 2, \ldots, k$ if one ultimately wants a total of k semi-supervised eigenvectors, then one lets Q be the $n \times (t-1)$ matrix with first column equal to 1 and with j^{th} column, for $i = 2, \ldots, t-1$, equal to x_{j-1}^* (where we emphasize that x_{j-1}^* is a vector not an element of a vector). That is, Q is of the form $Q = [1, x_1^*, \ldots, x_{t-1}^*]$, where x_i^* are successive semi-supervised eigenvectors, and the projection matrix FF^T is of the form $FF^T = I - D_G Q (Q^T D_G D_G Q)^{-1} Q^T D_G$, due to the degree-weighted inner norm. Then, by Eqn. (2), the t^{th} semi-supervised eigenvector takes the form

$$x_t^* \quad = \quad c\left(FF^T (L_G - \gamma_t D_G) FF^T\right)^+ D_G s. \tag{4}$$

Algorithm 1 Semi-supervised eigenvectors

Input: $L_G, D_G, s, \kappa = [\kappa_1, \ldots, \kappa_k]^T, \epsilon$
Require: $s^T D_G 1 = 0$, $s^T D_G s = 1$, $\kappa^T 1 \leq 1$
$Q = [1]$
for $t = 1$ to k **do**
  $FF^T \leftarrow I - D_G Q (Q^T D_G D_G Q)^{-1} Q^T D_G$
  $\top \leftarrow \lambda_2$ where $FF^T L_G FF^T v_2 = \lambda_2 FF^T D_G FF^T v_2$
  $\perp \leftarrow -\mathrm{vol}(G)$
  repeat
    $\gamma_t \leftarrow (\perp + \top)/2$ (Binary search over γ_t)
    $x_t \leftarrow (FF^T (L_G - \gamma_t D_G) FF^T)^+ FF^T D_G s$
    Normalize x_t such that $x_t^T D_G x_t = 1$
    if $(x_t^T D_G s)^2 > \kappa_t$ **then** $\perp \leftarrow \gamma_t$ **else** $\top \leftarrow \gamma_t$ **end if**
  until $\|(x_t^T D_G s)^2 - \kappa_t\| \leq \epsilon$ **or** $\|(\perp + \top)/2 - \gamma_t\| \leq \epsilon$
  Augment Q with x_t^* by letting $Q = [Q, x_t^*]$.
end for

In more detail, Algorithm 1 presents pseudo-code for our main algorithm for computing semi-supervised eigenvectors. Several things should be noted about our implementation. First, note that we implicitly compute the projection matrix FF^T. Second, a naïve approach to Eqn. (2) does not immediately lead to an efficient solution, since $D_G s$ will not be in the span of $(FF^T(L_G - \gamma D_G)FF^T)$, thus leading to a large residual. By changing variables so that $x = FF^T y$, the solution becomes $x^* \propto FF^T (FF^T (L_G - \gamma D_G) FF^T)^+ FF^T D_G s$. Since FF^T is a projection matrix, this expression is equivalent to $x^* \propto (FF^T (L_G - \gamma D_G) FF^T)^+ FF^T D_G s$. Third, we exploit that $FF^T (L_G - \gamma_i D_G) FF^T$ is an SPSD matrix, and we apply the conjugate gradient method, rather than computing the explicit pseudoinverse. That is, in the implementation we never represent the dense matrix FF^T, but instead we treat it as an operator and we simply evaluate the result of applying a vector to it on either side. Fourth, we use that λ_2 can never decrease (here we refer to λ_2 as the smallest non-zero eigenvalue of the modified matrix), so we only recalculate the upper bound for the binary search when an iteration saturates without satisfying $\|(x_t^T D_G s)^2 - \kappa_t\| \leq \epsilon$. In case of saturation one can for instance recalculate λ_2 iteratively by using the inverse iteration method, $v_2^{k+1} \propto (FF^T L_G FF^T - \lambda_2^{\text{est}} FF^T D_G FF^T)^+ FF^T D_G FF^T v_2^k$, and normalizing such that $(v_2^{k+1})^T v_2^{k+1} = 1$.

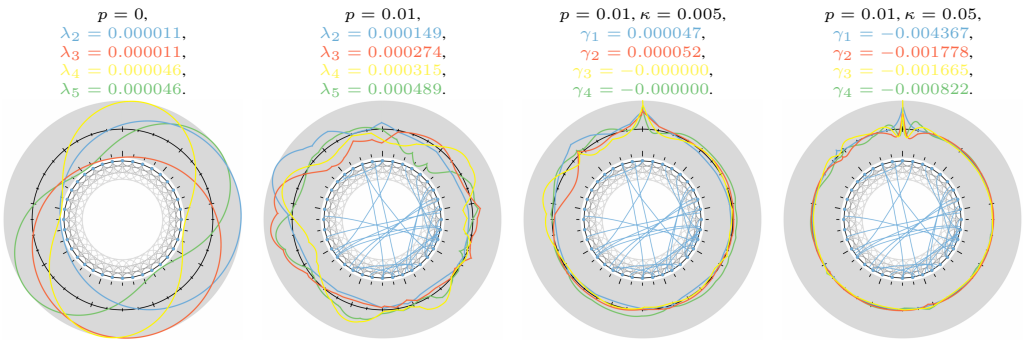
4 Illustrative Empirical Results

In this section, we will provide a detailed empirical evaluation of our method of semi-supervised eigenvectors and how they can be used for locally-biased machine learning. Our goal will be two-fold: first, to illustrate how the "knobs" of our method work; and second, to illustrate the usefulness of the method in a real application. To do so, we will consider:

- *Toy data.* In Section 4.1, we will consider one-dimensional examples of the popular "small world" model [22]. This is a parameterized family of models that interpolates between low-dimensional grids and random graphs; and, as such, it will allow us to illustrate the behavior of our method and it's various parameters in a controlled setting.
- *Handwritten image data.* In Section 4.2, we will consider the data from the MNIST digit data set [12]. These data have been widely-studied in machine learning and related areas and they have substantial "local heterogeneity"; and thus these data will allow us to illustrate how our method may be used to perform locally-biased versions of common machine learning tasks such as smoothing, clustering, and kernel construction.

4.1 Small-world Data

To illustrate how the "knobs" of our method work, and in particular how κ and γ interplay, we consider data constructed from the so-called small-world model. To demonstrate how semi-supervised eigenvectors can focus on specific target regions of a data graph to capture slowest modes of local variation, we plot semi-supervised eigenvectors around illustrations of (non-rewired and rewired) realizations of the small-world graph; see Figure 2.



(a) Global eigenvectors (b) Global eigenvectors (c) Semi-supervised eigenvectors (d) Semi-supervised eigenvectors

Figure 2: In each case, (a-d) the data consist of 3600 nodes, each connected to it's 8 nearest-neighbors. In the center of each subfigure, we show the nodes (blue) and edges (black and light gray are the local edges, and blue are the randomly-rewired edges). In each subfigure, we wrap a plot (black x-axis and gray background) visualizing the 4 smallest semi-supervised eigenvectors, allowing us to see the effect of random edges (different values of rewiring probability p) and degree of localization (different values of κ). Eigenvectors are color coded as blue, red, yellow, and green, starting with the one having the smallest eigenvalue. See the main text for more details.

In Figure 2.a, we show a graph with no randomly-rewired edges ($p = 0$) and a locality parameter κ such that the global eigenvectors are obtained. This yields a symmetric graph with eigenvectors corresponding to orthogonal sinusoids, *i.e.*, for all eigenvectors, except the all-ones with eigenvalue 0, the algebraic multiplicity is 2, *i.e.*, the first two capture the slowest mode of variation and correspond to a sine and cosine with equal random phase-shift (rotational ambiguity). In Figure 2.b, random edges have been added with probability $p = 0.01$ and the locality parameter κ is still chosen such that the global eigenvectors of the rewired graph are obtained. In particular, note small kinks in the eigenvectors at the location of the randomly added edges. Since the graph is no longer symmetric, all of the visualized eigenvectors have algebraic multiplicity 1. Moreover, note that the slow mode of variation in the interval on the top left; a normalized-cut based on the leading global eigenvector would extract this region since the remainder of the ring is more well-connected due to the degree of rewiring. In Figure 2.c, we see the same graph realization as in Figure 2.b, except that the semi-supervised eigenvectors have a seed node at the top of the circle and the correlation

parameter $\kappa_t = 0.005$. Note that, like the global eigenvectors, the local approach produces modes of increasing variation. In addition, note that the neighborhood around "11 o-clock" contains more mass, when compared with Figure 2.b; the reason for this is that this region is well-connected with the seed via a randomly added edge. Above the visualization we also show the γ_t that saturates κ_t, *i.e.*, γ_t is the Lagrange multiplier that defines the effective correlation κ_t. Not shown is that if we kept reducing κ, then γ_t would tend towards λ_{t+1}, and the respective semi-supervised eigenvector would tend towards the global eigenvector. Finally, in Figure 2.d, the desired correlation is increased to $\kappa = 0.05$ (thus decreasing the value of γ_t), making the different modes of variation more localized in the neighborhood of the seed. It should be clear that, in addition to being determined by the locality parameter, we can think of γ as a regularizer biasing the global eigenvectors towards the region near the seed set.

4.2 MNIST Digit Data

We now demonstrate the semi-supervised eigenvectors as a feature extraction preprocessing step in a machine learning setting. We consider the well-studied MNIST dataset containing 60000 training digits and 10000 test digits ranging from 0 to 9. We construct the complete 70000×70000 k-NN graph with $k = 10$ and with edge weights given by $w_{ij} = \exp(-\frac{4}{\sigma_i^2}\|x_i - x_j\|^2)$, where σ_i^2 being the Euclidean distance to it's nearest neighbor, and we define the graph Laplacian in the usual way. We evaluate the semi-supervised eigenvectors in a transductive learning setting by disregarding the majority of labels in the entire training data. We then use a few samples from each class to seed our semi-supervised eigenvectors, and a few others to train a downstream classification algorithm. Here we choose to apply the SGT of [11] for two main reasons. First, the transductive classifier is inherently designed to work on a subset of global eigenvectors of the graph Laplacian, making it ideal for validating that our localized basis constructed by the semi-supervised eigenvectors can be more informative when we are solely interested in the "local heterogeneity" near a seed set. Second, using the SGT based on global eigenvectors is a good point of comparison, because we are only interested in the effect of our subspace representation. (If we used one type of classifier in the local setting, and another in the global, the classification accuracy that we measure would obviously be biased.) As in [11], we normalize the spectrum of both global and semi-supervised eigenvectors by replacing the eigenvalues with some monotonically increasing function. We use $\lambda_i = \frac{i^2}{k^2}$, *i.e.*, focusing on ranking among smallest cuts; see [5]. Furthermore, we fix the regularization parameter of the SGT to $c = 3200$, and for simplicity we fix $\gamma = 0$ for all semi-supervised eigenvectors, implicitly defining the effective $\kappa = [\kappa_1, \ldots, \kappa_k]^T$. Clearly, other correlation distributions and values of γ may yield subspaces with even better discriminative properties[1].

	#Semi-supervised eigenvectors for SGT						#Global eigenvectors for SGT					
Labeled points	1	2	4	6	8	10	1	5	10	15	20	25
1 : 1	0.39	0.39	0.38	0.38	0.38	0.36	0.50	0.48	0.36	0.27	0.27	0.19
1 : 10	0.30	0.31	0.25	0.23	0.19	0.15	0.49	0.36	0.09	0.08	0.06	0.06
5 : 50	0.12	0.15	0.09	0.08	0.07	0.06	0.49	0.09	0.08	0.07	0.05	0.04
10 : 100	0.09	0.10	0.07	0.06	0.05	0.05	0.49	0.08	0.07	0.06	0.04	0.04
50 : 500	0.03	0.03	0.03	0.03	0.03	0.03	0.49	0.10	0.07	0.06	0.04	0.04

Table 1: Classification error for the SGT based on respectively semi-supervised and global eigenvectors. The first column from the left encodes the configuration, *e.g.*, 1:10 interprets as 1 seed and 10 training samples from each class (total of 22 samples - for the global approach these are all used for training). When the seed is well determined and the number of training samples moderate (50:500) a single semi-supervised eigenvector is sufficient, where for less data we benefit from using multiple semi-supervised eigenvectors. All experiments have been repeated 10 times.

Here, we consider the task of discriminating between *fours* and *nines*, as these two classes tend to overlap more than other combinations. (A closed *four* usually resembles *nine* more than an "open" *four*.) Hence, we expect localization on low order global eigenvectors, meaning that class separation will not be evident in the leading global eigenvector, but instead will be "buried" further down the spectrum. Thus, this will illustrate how semi-supervised eigenvectors can represent relevant heterogeneities in a local subspace of low dimensionality. Table 1 summarizes our classification results based on respectively semi-supervised and global eigenvectors. Finally, Figure 3 and 4 illustrates two realizations for the 1:10 configuration, where the training samples are fixed, but where we vary

[1] A thorough analysis regarding the importance of this parameter will appear in the journal version.

the seed nodes, to demonstrate the influence of the seed. See the caption in these figures for further details.

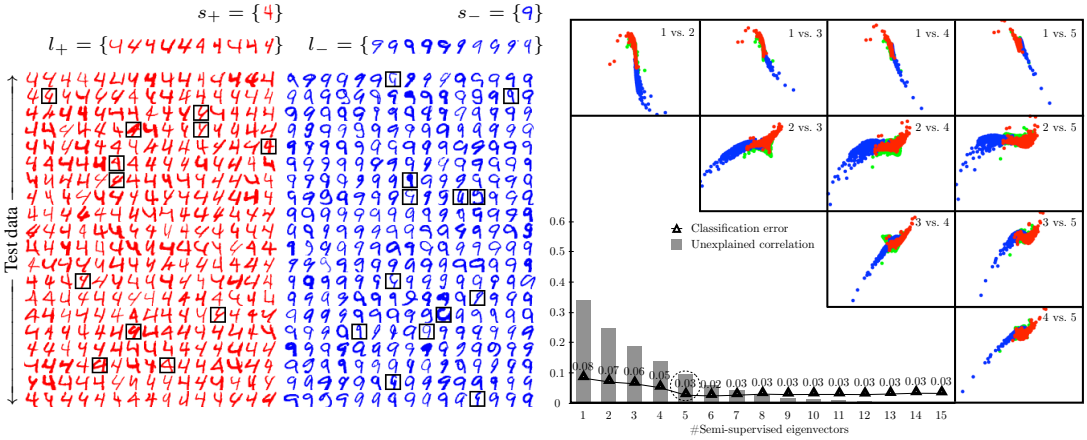


Figure 3: Left: Shows a subset of the classification results for the SGT based on 5 semi-supervised eigenvectors seeded in s_+ and s_-, and trained using samples l_+ and l_-. Misclassifications are marked with black frames. Right: Visualizes all test data spanned by the first 5 semi-supervised eigenvectors, by plotting each component as a function of the others. Red (blue) points correspond to 4 (9), whereas green points correspond to remaining digits. As the seed nodes are good representatives, we note that the eigenvectors provide a good class separation. We also plot the error as a function of local dimensionality, as well as the unexplained correlation, *i.e.*, initial components explain the majority of the correlation with the seed (effect of $\gamma = 0$). The particular realization based on the leading 5 semi-supervised eigenvectors yields an error of ≈ 0.03 (dashed circle).

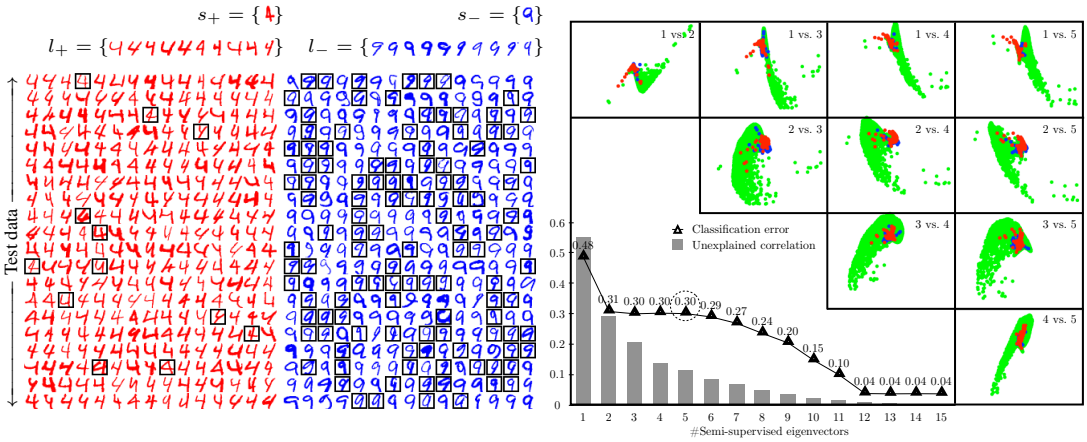


Figure 4: See the general description in Figure 3. Here we illustrate an instance where the s_+ shares many similarities with s_-, *i.e.*, s_+ is on the boundary of the two classes. This particular realization achieves a classification error of $\approx$ 0.30 (dashed circle). In this constellation we first discover localization on low order semi-supervised eigenvectors ($\approx$ 12 eigenvectors), which is comparable to the error based on global eigenvectors (see Table 1), *i.e.*, further down the spectrum we recover from the bad seed and pickup the relevant mode of variation.

In summary: We introduced the concept of semi-supervised eigenvectors that are biased towards local regions of interest in a large data graph. We demonstrated the feasibility on a well-studied dataset and found that our approach leads to more compact subspace representations by extracting desired local heterogeneities. Moreover, the algorithm is scalable as the eigenvectors are computed by the solution to a sparse system of linear equations, preserving the low $\mathcal{O}(m)$ space complexity. Finally, we foresee that the approach will prove useful in a wide range of data analysis fields, due to the algorithm's speed, simplicity, and stability.

References

[1] R. Andersen, F.R.K. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.

[2] R. Andersen and K. Lang. Communities from seed sets. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 223–232, 2006.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[5] O. Chapelle, J. Weston, and B. Schölkopf. Cluster Kernels for Semi-Supervised Learning. In Becker, editor, *NIPS 2002*, volume 15, pages 585–592, Cambridge, MA, USA, 2003.

[6] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition in data: Diffusion maps. *Proc. Natl. Acad. Sci. USA*, 102(21):7426–7431, 2005.

[7] A. P. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *Proceedings of th 11th International Conference on Computer Vision*, pages 1–8, 2007.

[8] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114/115:815–839, 1989.

[9] T.H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[10] G. Jeh and J. Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 271–279, 2003.

[11] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[12] Y. Lecun and C. Cortes. The MNIST database of handwritten digits.

[13] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW '08: Proceedings of the 17th International Conference on World Wide Web*, pages 695–704, 2008.

[14] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi. A local spectral method for graphs: with applications to improving graph partitions and exploring data graphs locally. Technical report, 2009. Preprint: arXiv:0912.0681.

[15] S. Maji, N. K. Vishnoi, and J. Malik. Biased normalized cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2057–2064, 2011.

[16] K.A. Norman, S.M. Polyn, G.J. Detre, and J.V. Haxby. Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in Cognitive Sciences*, 10(9):424–30, 2006.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[18] B. Scholkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[19] D.A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC '04: Proceedings of the 36th annual ACM Symposium on Theory of Computing*, pages 81–90, 2004.

[20] S.-H. Teng. The Laplacian paradigm: Emerging algorithms for massive graphs. In *Proceedings of the 7th Annual Conference on Theory and Applications of Models of Computation*, pages 2–14, 2010.

[21] S. Vigna. Spectral ranking. Technical report. Preprint: arXiv:0912.0238 (2009).

[22] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[23] S. X. Yu and J. Shi. Grouping with bias. In *Annual Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Conference*, pages 1327–1334, 2002.