Journal of Machine Learning Research 13 (2012) 2339-2365 Submitted 11/12; Published 9/12

A Local Spectral Method for Graphs: With Applications to Improving Graph Partitions and Exploring Data Graphs Locally

Michael W. Mahoney MMAHONEY@CS.STANFORD.EDU
Department of Mathematics
Stanford University
Stanford, CA 94305

Lorenzo Orecchia ORECCHIA@EECS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720

Nisheeth K. Vishnoi NISHEETH.VISHNOI@GMAIL.COM
Microsoft Research India
#9, Lavelle Road
Bangalore 560 025
India

Editor: Mikhail Belkin

Abstract

The second eigenvalue of the Laplacian matrix and its associated eigenvector are fundamental features of an undirected graph, and as such they have found widespread use in scientific computing, machine learning, and data analysis. In many applications, however, graphs that arise have several *local* regions of interest, and the second eigenvector will typically fail to provide information fine-tuned to each local region. In this paper, we introduce a locally-biased analogue of the second eigenvector, and we demonstrate its usefulness at highlighting local properties of data graphs in a semi-supervised manner. To do so, we first view the second eigenvector as the solution to a constrained optimization problem, and we incorporate the local information as an additional constraint; we then characterize the optimal solution to this new problem and show that it can be interpreted as a generalization of a Personalized PageRank vector; and finally, as a consequence, we show that the solution can be computed in nearly-linear time. In addition, we show that this locally-biased vector can be used to compute an approximation to the best partition *near* an input seed set in a manner analogous to the way in which the second eigenvector of the Laplacian can be used to obtain an approximation to the best partition in the entire input graph. Such a primitive is useful for identifying and refining clusters locally, as it allows us to focus on a local region of interest in a semi-supervised manner. Finally, we provide a detailed empirical evaluation of our method by showing how it can applied to finding locally-biased sparse cuts around an input vertex seed set in social and information networks.

Keywords: spectral graph partitioning, local spectral algorithms, Laplacian matrix, semi-supervised learning, personalized pagerank

1. Introduction

Spectral methods are popular in machine learning, data analysis, and applied mathematics due to their strong underlying theory and their good performance in a wide range of applications. In the study of undirected graphs, in particular, spectral techniques play an important role, as many fundamental structural properties of a graph depend directly on spectral quantities associated with matrices representing the graph. Two fundamental objects of study in this area are the second smallest eigenvalue of the graph Laplacian and its associated eigenvector. These quantities determine many features of the graph, including the behavior of random walks and the presence of sparse cuts. This relationship between the graph structure and an easily-computable quantity has been exploited in data clustering, community detection, image segmentation, parallel computing, and many other applications.

A potential drawback of using the second eigenvalue and its associated eigenvector is that they are inherently *global* quantities, and thus they may not be sensitive to very *local* information. For instance, a sparse cut in a graph may be poorly correlated with the second eigenvector (and even with all the eigenvectors of the Laplacian) and thus invisible to a method based only on eigenvector analysis. Similarly, based on domain knowledge one might have information about a specific target region in the graph, in which case one might be interested in finding clusters only near this prespecified local region, for example, in a semi-supervised manner; but this local region might be essentially invisible to a method that uses only global eigenvectors. For these and related reasons, standard global spectral techniques can have substantial difficulties in semi-supervised settings, where the goal is to learn more about a locally-biased target region of the graph.

In this paper, we provide a methodology to construct a locally-biased analogue of the second eigenvalue and its associated eigenvector, and we demonstrate both theoretically and empirically that this localized vector inherits many of the good properties of the global second eigenvector. Our approach is inspired by viewing the second eigenvector as the optimum of a constrained global quadratic optimization program. To model the localization step, we modify this program by adding a natural locality constraint. This locality constraint requires that any feasible solution have sufficient correlation with the target region, which we assume is given as input in the form of a set of nodes or a distribution over vertices. The resulting optimization problem, which we name LocalSpectral and which is displayed in Figure 1, is the main object of our work.

The main advantage of our formulation is that an optimal solution to LocalSpectral captures many of the same structural properties as the global eigenvector, except in a locally-biased setting. For example, as with the global optimization program, our locally-biased optimization program has an intuitive geometric interpretation. Similarly, as with the global eigenvector, an optimal solution to LocalSpectral is efficiently computable. To show this, we characterize the optimal solutions of LocalSpectral and show that such a solution can be constructed in nearly-linear time by solving a system of linear equations. In applications where the eigenvectors of the graph are pre-computed and only a small number of them are needed to describe the data, the optimal solution to our program can be obtained by performing a small number of inner product computations. Finally, the optimal solution to LocalSpectral can be used to derive bounds on the mixing time of random walks that start near the local target region as well as on the existence of sparse cuts near the locally-biased target region. In particular, it lower bounds the conductance of cuts as a function of how well-correlated they are with the seed vector. This will allow us to exploit the analogy between global eigenvectors and our localized analogue to design an algorithm for discovering sparse cuts near an input seed set of vertices.

In order to illustrate the empirical behavior of our method, we will describe its performance on the problem of finding locally-biased sparse cuts in real data graphs. Subsequent to the dissemination of the initial technical report version of this paper, our methodology was applied to the problem of finding, given a small number of "ground truth" labels that correspond to known segments in an image, the segments in which those labels reside (Maji, Vishnoi, and Malik, 2011). This computer vision application will be discussed briefly. Then, we will describe in detail how our algorithm for discovering sparse cuts near an input seed set of vertices may be applied to the problem of exploring data graphs locally and to identifying locally-biased clusters and communities in a more difficult-to-visualize social network application. In addition to illustrating the performance of the method in a practical application related to the one that initially motivated this work (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010), this social graph application will illustrate how the various "knobs" of our method can be used in practice to explore the structure of data graphs in a locally-biased manner.

Our method uses ideas from spectral graph theory; for a detailed introduction to this topic, see Chung (1997). Recent theoretical work has focused on using spectral ideas to find good clusters nearby an input seed set of nodes (Spielman and Teng, 2004; Andersen, Chung, and Lang, 2006; Chung, 2007). These methods are based on running a number of local random walks around the seed set and using the resulting distributions to extract information about clusters in the graph. This line of work grew out of attempts to develop linear equation solvers that run in time nearly linear in the number of edges in the graph (Spielman and Teng, 2004), and work subsequent to the initial technical report version of this paper has provided implementations of these ideas (Koutis, Miller, and Peng, 2010). The connections with our work described in this article remain to be explored.

Recent empirical work has used Personalized PageRank, a particular variant of a local random walk, to characterize very finely the clustering and community structure in a wide range of very large social and information networks (Andersen and Lang, 2006; Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010). In contrast with previous methods, our local spectral method is the first to be derived in a direct way from an explicit optimization problem inspired by the global spectral problem. Interestingly, our characterization also shows that optimal solutions to LocalSpectral are generalizations of Personalized PageRank, providing an additional insight to why local random walk methods work well in practice.

In the next section, we will describe relevant background and notation; and then, in Section 3, we will present our formulation of a locally-biased spectral optimization program, the solution of which will provide a locally-biased analogue of the second eigenvector of the graph Laplacian. Then, in Section 4 we will describe how our method may be applied to identifying and refining locally-biased partitions in a graph; and in Section 5 we will provide a detailed empirical evaluation of our algorithm. Finally, in Section 6, we will conclude with a discussion of our results in a broader context.

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 weight w_{ij}. For a set of vertices $S\subseteq V$ in a graph, the *volume of S* is $\mathrm{vol}(S)\stackrel{\mathrm{def}}{=}\sum_{i\in S}d_i$, in which case the *volume of the graph G* is $\mathrm{vol}(G)\stackrel{\mathrm{def}}{=}\mathrm{vol}(V)=2m$. 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, that is, $D_G(i,i) = d_i = \sum_{\{i,j\}\in E} w_{ij}$, the weighted degree of vertex 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 $x^T L_G x = \frac{1}{2}\sum_{ij\in E} w_{ij}(x_i - x_j)^2$, 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. For a symmetric matrix A, we will use $A \succeq 0$ to denote that it is positive semi-definite. Moreover, given two symmetric matrices A and B, the expression $A \succeq B$ will mean $A - B \succeq 0$. Further, for two $n \times n$ matrices A and B, we let $A \circ B$ denote $\mathrm{Tr}\,(A^T B)$. 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 $x^T D_G y \stackrel{\text{def}}{=} \sum_{i=1}^n x_i y_i d_i$. 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. We consider the following definition of the complete graph K_n on the vertex set V: $A_{K_n} \stackrel{\text{def}}{=} \frac{1}{\mathrm{vol}(G)} D_G 1 1^T D_G$. Note that this is not the standard complete graph, but a weighted version of it, where the weights depend on D_G. With this scaling we have $D_{K_n} = D_G$. Hence, the Laplacian of the complete graph defined in this manner becomes $L_{K_n} = D_G - \frac{1}{\mathrm{vol}(G)} D_G 1 1^T D_G$.

In this paper, the *conductance* $\phi(S)$ *of a cut* $(S, \bar{S})$ is $\phi(S) \stackrel{\text{def}}{=} \mathrm{vol}(G) \cdot \frac{|E(S,\bar{S})|}{\mathrm{vol}(S)\cdot \mathrm{vol}(\bar{S})}$. A sparse cut, also called a good-conductance partition, is one for which $\phi(S)$ is small. The *conductance of the graph* G is then $\phi(G) = \min_{S\subseteq V} \phi(S)$. Note that the conductance of a set S, or equivalently a cut $(S, \bar{S})$, is often defined as $\phi'(S) = |E(S,\bar{S})| / \min\{\mathrm{vol}(S), \mathrm{vol}(\bar{S})\}$. This notion is equivalent to that $\phi(S)$, in that the value $\phi(G)$ thereby obtained for the conductance of the graph G differs by no more than a factor of 2 times the constant $\mathrm{vol}(G)$, depending on which notion we use for the conductance of a set.

3. The LocalSpectral Optimization Program

In this section, we introduce the local spectral optimization program $\mathsf{LocalSpectral}(G, s, \kappa)$ as a strengthening of the usual global spectral program $\mathsf{Spectral}(G)$. To do so, we will augment $\mathsf{Spectral}(G)$ with a locality constraint of the form $(x^T D_G s)^2 \geq \kappa$, for a seed vector s and a correlation parameter κ. Both these programs are homogeneous quadratic programs, with optimization variable the vector $x \in \mathbb{R}^V$, and thus any solution vector x is essentially equivalent to $-x$ for the purpose of these optimizations. Hence, in the following we do not differentiate between x and $-x$, and we assume a suitable direction is chosen in each instance.

3.1 Motivation for the Program

Recall that the second eigenvalue $\lambda_2(G)$ of the Laplacian L_G can be viewed as the optimum of the standard optimization problem $\mathsf{Spectral}(G)$ described in Figure 1. In matrix terminology, the corresponding optimal solution v_2 is a generalized eigenvector of L_G with respect to D_G. For our purposes, however, it is best to consider the geometric meaning of this optimization formulation. To do so, suppose we are operating in a vector space $\mathbb{R}^V$, where the ith dimension is stretched by a factor of d_i, so that the natural identity operator is D_G and the inner product between two vectors

$$
\begin{aligned}
\min \quad & x^T L_G x \\
\text{s.t.} \quad & x^T D_G x = 1 \\
& (x^T D_G 1)^2 = 0 \\
& x \in \mathbb{R}^V
\end{aligned}
\qquad\qquad
\begin{aligned}
\min \quad & x^T L_G x \\
\text{s.t.} \quad & x^T D_G x = 1 \\
& (x^T D_G 1)^2 = 0 \\
& (x^T D_G s)^2 \geq \kappa \\
& x \in \mathbb{R}^V
\end{aligned}
$$

Figure 1: Global and local spectral optimization programs. Left: The usual spectral program $\mathsf{Spectral}(G)$. Right: Our new locally-biased spectral program $\mathsf{LocalSpectral}(G,s,\kappa)$. In both cases, the optimization variable is the vector $x \in \mathbb{R}^n$.

x and y is given by $\sum_{i\in V} d_i x_i y_i = x^T D_G y$. In this representation, $\mathsf{Spectral}(G)$ is seeking the vector $x \in \mathbb{R}^V$ that is orthogonal to the all-one vector, lies on the unit sphere, and minimizes the Laplacian quadratic form. Note that such an optimum v_2 may lie anywhere on the unit sphere.

Our goal here is to modify $\mathsf{Spectral}(G)$ to incorporate a bias towards a target region which we assume is given to us as an input vector s. 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. We obtain $\mathsf{LocalSpectral}(G,s,\kappa)$ from $\mathsf{Spectral}(G)$ by requiring that a feasible solution also have a sufficiently large correlation with the vector s. This is achieved by the addition of the constraint $(x^T D_G s)^2 \geq \kappa$, which ensures that the projection of x onto the direction s is at least $\sqrt{\kappa}$ in absolute value, where the parameter κ is also an input parameter ranging between 0 and 1. Thus, we would like the solution to be well-connected with or to lie near the *seed* vector s. In particular, as displayed pictorially in Figure 2, x must lie within the spherical cap centered at s that contains all vectors at an angle of at most $\arccos(\sqrt{\kappa})$ from s. Thus, higher values of κ demand a higher correlation with s and, hence, a stronger localization. Note that in the limit $\kappa = 0$, the spherical cap constituting the feasible region of the program is guaranteed to include v_2 and $\mathsf{LocalSpectral}(G,s,\kappa)$ is equivalent to $\mathsf{Spectral}(G)$. In the rest of this paper, we refer to s as the *seed vector* and to κ as the *correlation parameter* for a given $\mathsf{LocalSpectral}(G,s,\kappa)$ optimization problem. Moreover, we denote the objective value of the program $\mathsf{LocalSpectral}(G,s,\kappa)$ by the number $\lambda(G,s,\kappa)$.

3.2 Characterization of the Optimal Solutions of LocalSpectral

Our first theorem is a characterization of the optimal solutions of LocalSpectral. Although LocalSpectral is a non-convex program (as, of course, is Spectral), the following theorem states that solutions to it can be expressed as the solution to a system of linear equations which has a natural interpretation. The proof of this theorem (which may be found in Section 3.4) will involve a relaxation of the non-convex program LocalSpectral to a convex semidefinite program (SDP), that is, the variables in the optimization program will be distributions over vectors rather than the vectors themselves. For the statement of this theorem, recall that A^+ denotes the (uniquely defined) Moore-Penrose pseudoinverse of the matrix A.

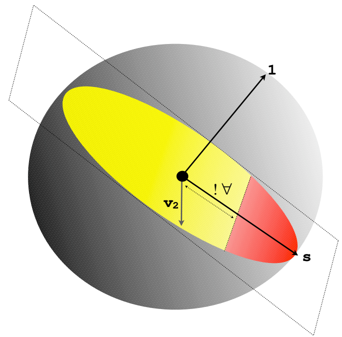


Figure 2: (Best seen in color.) Pictorial representation of the feasible regions of the optimization programs $\mathsf{Spectral}(G)$ and $\mathsf{LocalSpectral}(G,s,\kappa)$ that are defined in Figure 1. See the text for a discussion.

Theorem 1 (Solution Characterization) *Let $s \in \mathbb{R}^V$ be a seed vector such that $s^T D_G 1 = 0$, $s^T D_G s = 1$, and $s^T D_G v_2 \neq 0$, where v_2 is the second generalized eigenvector of L_G with respect to D_G. In addition, let $1 > \kappa \geq 0$ be a correlation parameter, and let $x^\star$ be an optimal solution to $\mathsf{LocalSpectral}(G,s,\kappa)$. Then, there exists some $\gamma \in (-\infty, \lambda_2(G))$ and a $c \in [0,\infty]$ such that*

$$x^\star = c(L_G - \gamma D_G)^+ D_G s. \tag{1}$$

There are several parameters (such as s, κ, γ, and c) in the statement of Theorem 1, and understanding their relationship is important: s and κ are the parameters of the program; c is a normalization factor that rescales the norm of the solution vector to be 1 (and that can be computed in linear time, given the solution vector); and γ is implicitly defined by κ, G, and s. The correct setting of γ ensures that $(s^T D_G x^\star)^2 = \kappa$, that is, that $x^\star$ is found exactly on the boundary of the feasible region. At this point, it is important to notice the behavior of $x^\star$ and γ as κ changes. As κ goes to 1, γ tends to $-\infty$ and $x^\star$ approaches s; conversely, as κ goes to 0, γ goes to $\lambda_2(G)$ and $x^\star$ tends towards v_2, the global eigenvector. We will discuss how to compute γ and $x^\star$, given a specific κ, in Section 3.3.

Finally, we should note that there is a close connection between the solution vector $x^\star$ and the popular PageRank procedure. Recall that PageRank refers to a method to determine a global rank or global notion of importance for a node in a graph such as the web that is based on the link structure of the graph (Brin and Page, 1998; Langville and Meyer, 2004; Berkhin, 2005). There have been several extensions to the basic PageRank concept, including Topic-Sensitive PageRank (Haveliwala, 2003) and Personalized PageRank (Jeh and Widom, 2003). In the same way that PageRank can be viewed as a way to express the quality of a web page over the entire web, Personalized PageRank expresses a link-based measure of page quality around user-selected pages. In particular, given a vector $s \in \mathbb{R}^V$ and a *teleportation* constant $\alpha > 0$, the Personalized PageRank vector can be written as $\mathrm{pr}_{\alpha,s} = \left(L_G + \frac{1-\alpha}{\alpha} D_G\right)^{-1} D_G s$ (Andersen, Chung, and Lang, 2006). By setting $\gamma = -\frac{1-\alpha}{\alpha}$, the optimal solution to $\mathsf{LocalSpectral}$ is proved to be a generalization of Personalized PageRank. In particular, this means that for high values of the correlation parameter κ, for which the corresponding γ in Theorem 1 is negative, the optimal solution to $\mathsf{LocalSpectral}$ takes the form of a Personalized PageRank vector. On the other hand, when $\gamma \geq 0$, the optimal solution to $\mathsf{LocalSpectral}$ provides a smooth way of transitioning from the Personalized PageRank vector to the global second eigenvector v_2.

3.3 Computation of the Optimal Solutions of LocalSpectral

Here, we discuss how to compute efficiently an optimal solution for $\mathsf{LocalSpectral}(G,s,\kappa)$, for a fixed choice of the parameters G, s, and κ. The following theorem is our main result.

Theorem 2 (Solution Computation) *For any $\varepsilon > 0$, a solution to $\mathsf{LocalSpectral}(G,s,\kappa)$ of value at most $(1+\varepsilon)\cdot\lambda(G,s,\kappa)$ can be computed in time $\tilde{O}({}^{m}/\sqrt{\lambda_2(G)}\cdot\log({}^{1}/\varepsilon))$ using the Conjugate Gradient Method (Golub and Loan, 1996). Alternatively, such a solution can be computed in time $\tilde{O}(m\log({}^{1}/\varepsilon))$ using the Spielman-Teng linear-equation solver (Spielman and Teng, 2004).*

Proof By Theorem 1, we know that the optimal solution $x^\star$ must be a unit-scaled version of $y(\gamma) = (L_G - \gamma D_G)^+ D_G s$, for an appropriate choice of $\gamma \in (-\infty, \lambda_2(G))$. Notice that, given a fixed γ, the task of computing $y(\gamma)$ is equivalent to solving the system of linear equations $(L_G - \gamma D_G)y = D_G s$ for the unknown y. This operation can be performed, up to accuracy ε, in time $\tilde{O}({}^{m}/\sqrt{\lambda_2(G)}\cdot\log({}^{1}/\varepsilon))$ using the Conjugate Gradient Method, or in time $\tilde{O}(m\log({}^{1}/\varepsilon))$ using the Spielman-Teng linear-equation solver. To find the correct setting of γ, it suffices to perform a binary search over the possible values of γ in the interval $(-\mathrm{vol}(G), \lambda_2(G))$, until $(s^T D_G x)^2$ is sufficiently close to κ. ■

We should note that, depending on the application, other methods of computing a solution to $\mathsf{LocalSpectral}(G,s,\kappa)$ might be more appropriate. In particular, if an eigenvector decomposition of L_G has been pre-computed, as is the case in certain machine learning and data analysis applications, then this computation can be modified as follows. Given an eigenvector decomposition of L_G as $L_G = \sum_{i=2}^n \lambda_i D_G^{1/2} u_i u_i^T D_G^{1/2}$, then $y(\gamma)$ must take the form

$$y(\gamma) = (L_G - \gamma D_G)^+ D_G s = \sum_{i=2}^{n} \frac{1}{\lambda_i - \gamma}(s^T D_G^{1/2} u_i)^2,$$

for the same choice of c and γ, as in Theorem 1. Hence, given the eigenvector decomposition, each guess $y(\gamma)$ of the binary search can be computed by expanding the above series, which requires a linear number of inner product computations. While this may yield a worse running time than Theorem 2 in the worst case, in the case that the graph is well-approximated by a small number k of dominant eigenvectors, then the computation is reduced to only k straightforward inner product computations.

3.4 Proof of Theorem 1

We start with an outline of the proof. Although the program $\mathsf{LocalSpectral}(G,s,\kappa)$ is *not* convex, it can be relaxed to the convex semidefinite program $\mathsf{SDP}_p(G,s,\kappa)$ of Figure 3. Then, one can observe that strong duality holds for this SDP relaxation. Using strong duality and the related complementary slackness conditions, one can argue that the primal $\mathsf{SDP}_p(G,s,\kappa)$ has a rank one unique optimal solution under the conditions of the theorem. This implies that the optimal solution of $\mathsf{SDP}_p(G,s,\kappa)$ is the same as the optimal solution of $\mathsf{LocalSpectral}(G,s,\kappa)$. Moreover, combining this fact with the complementary slackness condition obtained from the dual $\mathsf{SDP}_d(G,s,\kappa)$ of Figure 3, one can derive that the optimal rank one solution is of the form promised by Theorem 1.

$$
\begin{array}{lrlr}
\text{minimize} & L_G \circ X & \text{maximize} & \alpha + \kappa\beta \\
\text{s.t.} & L_{K_n} \circ X = 1 & \text{s.t.} & L_G \succeq \alpha L_{K_n} + \beta (D_G s)(D_G s)^T \\
 & (D_G s)(D_G s)^T \circ X \geq \kappa & & \beta \geq 0 \\
 & X \succeq 0 & & \alpha \in \mathbb{R}
\end{array}
$$

Figure 3: Left: Primal SDP relaxation of $\mathsf{LocalSpectral}(G,s,\kappa)$: $\mathsf{SDP}_p(G,s,\kappa)$; for this primal, the optimization variable is $X \in \mathbb{R}^{V\times V}$ such that X is symmetric and positive semidefinite. Right: Dual SDP relaxation of $\mathsf{LocalSpectral}(G,s,\kappa)$: $\mathsf{SDP}_d(G,s,\kappa)$; for this dual, the optimization variables are $\alpha, \beta \in \mathbb{R}$. Recall that $L_{K_n} \stackrel{\text{def}}{=} D_G - \frac{1}{\mathrm{vol}(G)} D_G 1 1^T D_G$.

Before proceeding with the details of the proof, we pause to make several points that should help to clarify our approach.

- First, since it may seem to some readers to be unnecessarily complex to relax LocalSpectral as an SDP, we emphasize that the motivation for relaxing it in this way is that we would like to prove Theorem 1. To prove this theorem, we must understand the form of the optimal solutions to the non-convex program LocalSpectral. Thus, in order to overcome the non-convexity, we relax LocalSpectral to $\mathsf{SDP}_p(G,s,\kappa)$ (of Figure 3) by “lifting” the rank-1 condition implicit in LocalSpectral. Then, strong duality applies; and it implies a set of sufficient optimality conditions. By combining these conditions, we will be able to establish that an optimal solution $X^\star$ to $\mathsf{SDP}_p(G,s,\kappa)$ has rank 1, that is, it has the form $X^\star = x^\star x^{\star T}$ for some vector $x^\star$; and thus it yields an optimal solution to LocalSpectral, that is, the vector $x^\star$.

- Second, in general, the value of a relaxation like $\mathsf{SDP}_p(G,s,\kappa)$ may be strictly less than that of the original program (LocalSpectral, in this case). Our characterization and proof will imply that the relaxation is tight, that is, that the optimum of $\mathsf{SDP}_p(G,s,\kappa)$ equals that of LocalSpectral. The reason is that one can find a rank-1 optimal solution to $\mathsf{SDP}_p(G,s,\kappa)$, which then yields an optimal solution of the same value for LocalSpectral. Note that this also implies that strong duality holds for the non-convex LocalSpectral, although this observation is not needed for our proof.

That is, although it may be possible to prove Theorem 1 in some other way that does not involve SDPs, we chose this proof since it is simple and intuitive and correct; and we note that Appendix B in the textbook of Boyd and Vandenberghe (2004) proves a similar statement by the same SDP-based approach.

Returning to the details of the proof, we will proceed to prove the theorem by establishing a sequence of claims. First, consider $\mathsf{SDP}_p(G,s,\kappa)$ and its dual $\mathsf{SDP}_d(G,s,\kappa)$ (as shown in Figure 3). The following claim uses the fact that, given $X = xx^T$ for $x \in \mathbb{R}^V$, and for any matrix $A \in \mathbb{R}^{V\times V}$, we have that $A \circ X = x^T A x$. In particular, $L_G \circ X = x^T L_G x$, for any graph G, and $(x^T D_G s)^2 = x^T D_G s s^T D_G x = D_G s s^T D_G \circ X$.

Claim 1 *The primal* $\mathsf{SDP}_p(G,s,\kappa)$ *is a relaxation of the vector program* *LocalSpectral*(G,s,κ).

Proof Consider a vector x that is a feasible solution to $\mathsf{LocalSpectral}(G,s,\kappa)$, and note that $X = xx^T$ is a feasible solution to $\mathsf{SDP}_p(G,s,\kappa)$. ■

Next, we establish the strong duality of $\mathsf{SDP}_p(G,s,\kappa)$. (Note that the feasibility conditions and complementary slackness conditions stated below may not suffice to establish the optimality, in the absence of this claim; hence, without this claim, we could not prove the subsequent claims, which are needed to prove the theorem.)

Claim 2 *Strong duality holds between* $\mathsf{SDP}_p(G,s,\kappa)$ *and* $\mathsf{SDP}_d(G,s,\kappa)$.

Proof Since $\mathsf{SDP}_p(G,s,\kappa)$ is convex, it suffices to verify that Slater's constraint qualification condition (Boyd and Vandenberghe, 2004) is true for this primal SDP. Consider $X = ss^T$. Then, $(D_G s)(D_G s)^T \circ ss^T = (s^T D_G s)^2 = 1 > \kappa$. ■

Next, we use this result to establish the following two claims. In particular, strong duality allows us to prove the following claim showing the KKT-conditions, that is, the feasibility conditions and complementary slackness conditions stated below, suffice to establish optimality.

Claim 3 *The following feasibility and complementary slackness conditions are sufficient for a primal-dual pair* $X^\star,\alpha^\star,\beta^\star$ *to be an optimal solution. The feasibility conditions are:*

$$\begin{aligned} L_{K_n} \circ X^\star &= 1, \\ (D_G s)(D_G s)^T \circ X^\star &\geq \kappa, \\ L_G - \alpha^\star L_{K_n} - \beta^\star (D_G s)(D_G s)^T &\succeq 0, \text{ and} \\ \beta^\star &\geq 0, \end{aligned} \tag{2}$$

and the complementary slackness conditions are:

$$\begin{aligned} \alpha^\star (L_{K_n} \circ X^\star - 1) &= 0, \\ \beta^\star ((D_G s)(D_G s)^T \circ X^\star - \kappa) &= 0, \text{ and} \end{aligned} \tag{3}$$

$$X^\star \circ (L_G - \alpha^\star L_{K_n} - \beta^\star (D_G s)(D_G s)^T) = 0. \tag{4}$$

Proof This follows from the convexity of $\mathsf{SDP}_p(G,s,\kappa)$ and Slater's condition (Boyd and Vandenberghe, 2004). ■

Claim 4 *These feasibility and complementary slackness conditions, coupled with the assumptions of the theorem, imply that* $X^\star$ *must be rank* 1 *and* $\beta^\star > 0$.

Proof Plugging in v_2 in Equation (2), we obtain that $v_2^T L_G v_2 - \alpha^\star - \beta^\star (v_2^T D_G s)^2 \geq 0$. But $v_2^T L_G v_2 = \lambda_2(G)$ and $\beta^\star \geq 0$. Hence, $\lambda_2(G) \geq \alpha^\star$. Suppose $\alpha^\star = \lambda_2(G)$. As $s^T D_G v_2 \neq 0$, it must be the case that $\beta^\star = 0$. Hence, by Equation (4), we must have $X^\star \circ L(G) = \lambda_2(G)$, which implies that $X^\star = v_2 v_2^T$, that is, the optimum for $\mathsf{LocalSpectral}$ is the global eigenvector v_2. This corresponds to a choice of $\gamma = \lambda_2(G)$ and c tending to infinity.

Otherwise, we may assume that $\alpha^\star < \lambda_2(G)$. Hence, since G is connected and $\alpha^\star < \lambda_2(G)$, $L_G - \alpha^\star L_{K_n}$ has rank exactly $n-1$ and kernel parallel to the vector 1. From the complementary slackness condition (4) we can deduce that the image of $X^\star$ is in the kernel of $L_G - \alpha^\star L_{K_n} - \beta^\star (D_G s)(D_G s)^T$. If

$\beta^\star > 0$, we have that $\beta^\star(D_G s)(D_G s)^T$ is a rank one matrix and, since $s^T D_G 1 = 0$, it reduces the rank of $L_G - \alpha^\star L_{K_n}$ by one precisely. If $\beta^\star = 0$ then $X^\star$ must be 0 which is not possible if $\mathsf{SDP}_p(G,s,\kappa)$ is feasible. Hence, the rank of $L_G - \alpha^\star L_{K_n} - \beta^\star(D_G s)(D_G s)^T$ must be exactly $n-2$. As we may assume that 1 is in the kernel of $X^\star$, $X^\star$ must be of rank one. This proves the claim. ■

Now we complete the proof of the theorem. From the claim it follows that, $X^\star = x^\star x^{\star T}$ where $x^\star$ satisfies the equation $(L_G - \alpha^\star L_{K_n} - \beta^\star(D_G s)(D_G s)^T)x^\star = 0$. From the second complementary slackness condition, Equation (3), and the fact that $\beta^\star > 0$, we obtain that $(x^\star)^T D_G s = \pm\sqrt{\kappa}$. Thus, $x^\star = \pm\beta^\star\sqrt{\kappa}(L_G - \alpha^\star L_{K_n})^+ D_G s$, as required.

4. Application to Partitioning Graphs Locally

In this section, we describe the application of LocalSpectral to finding locally-biased partitions in a graph, that is, to finding sparse cuts around an input seed vertex set in the graph. For simplicity, in this part of the paper, we let the instance graph G be unweighted.

4.1 Background on Global Spectral Algorithms for Partitioning Graphs

We start with a brief review of global spectral graph partitioning. Recall that the basic global graph partitioning problem is: given as input a graph $G = (V,E)$, find a set of nodes $S \subseteq V$ to solve

$$\phi(G) = \min_{S\subseteq V} \phi(S).$$

Spectral methods approximate the solution to this intractable global problem by solving the relaxed problem $\mathsf{Spectral}(G)$ presented in Figure 1. To understand this optimization problem, recall that $x^T L_G x$ counts the number of edges crossing the cut and that $x^T D_G x = 1$ encodes a variance constraint; thus, the goal of $\mathsf{Spectral}(G)$ is to minimize the number of edges crossing the cut subject to a given variance. Recall that for $T \subseteq V$, we let $1_T \in \{0,1\}^V$ be a vector which is 1 for vertices in T and 0 otherwise. Then for a cut $(S,\bar{S})$, if we define the vector $v_S \stackrel{\text{def}}{=} \sqrt{\frac{\mathrm{vol}(S)\cdot\mathrm{vol}(\bar{S})}{\mathrm{vol}(G)}} \cdot \left(\frac{1_S}{\mathrm{vol}(S)} - \frac{1_{\bar{S}}}{\mathrm{vol}\bar{S}}\right)$, it can be checked that v_S satisfies the constraints of Spectral and has objective value $\phi(S)$. Thus, $\lambda_2(G) \le \min_{S\subseteq V}\phi(S) = \phi(G)$.

Hence, $\mathsf{Spectral}(G)$ is a relaxation of the minimum conductance problem. Moreover, this program is a good relaxation in that a good cut can be recovered by considering a truncation, that is, a sweep cut, of the vector v_2 that is the optimal solution to $\mathsf{Spectral}(G)$. (That is, for example, consider each of the n cuts defined by the vector v_2, and return the cut with minimum conductance value.) This is captured by the following celebrated result often referred to as Cheeger's Inequality.

Theorem 3 (Cheeger's Inequality) *For a connected graph G, $\phi(G) \le O(\sqrt{\lambda_2(G)})$.*

Although there are many proofs known for this theorem (see, e.g., Chung, 1997), a particularly interesting proof was found by Mihail (1989); this proof involves rounding any *test vector* (rather than just the optimal vector), and it achieves the same guarantee as Cheeger's Inequality.

Theorem 4 (Sweep Cut Rounding) *Let x be a vector such that $x^T D_G 1 = 0$. Then there is a t for which the set of vertices $S := \mathsf{SweepCut}_t(x) \stackrel{\text{def}}{=} \{i : x_i \ge t\}$ satisfies $\frac{x^T L_G x}{x^T D_G x} \ge \phi^2(S)/8$.*

It is the form of Cheeger's Inequality provided by Theorem 4 that we will use below.

4.2 Locally-Biased Spectral Graph Partitioning

Here, we will exploit the analogy between Spectral and LocalSpectral by applying the global approach just outlined to the following locally-biased graph partitioning problem: given as input a graph $G = (V,E)$, an input node u, and a positive integer k, find a set of nodes $T \subseteq V$ achieving

$$\phi(u,k) = \min_{T \subseteq V : u \in T, \mathrm{vol}(T) \leq k} \phi(T).$$

That is, the problem is to find the best conductance set of nodes of volume no greater than k that contains the input node v.

As a first step, we show that we can choose the seed set and correlation parameters s and κ such that LocalSpectral(G,s,κ) is a relaxation for this locally-biased graph partitioning problem.

Lemma 5 *For $u \in V$, LocalSpectral$(G, v_{\{u\}}, 1/k)$ is a relaxation of the problem of finding a minimum conductance cut T in G which contains the vertex u and is of volume at most k. In particular, $\lambda(G, v_{\{u\}}, 1/k) \leq \phi(u,k)$.*

Proof If we let $x = v_T$ in LocalSpectral$(G, v_{\{u\}}, 1/k)$, then $v_T^T L_G v_T = \phi(T)$, $v_T^T D_G 1 = 0$, and $v_T^T D_G v_T = 1$. Moreover, we have that $(v_T^T D_G v_{\{u\}})^2 = \frac{d_u(2m-\mathrm{vol}(T))}{\mathrm{vol}(T)(2m-d_u)} \geq 1/k$, which establishes the lemma. ■

Next, we can apply Theorem 4 to the optimal solution for LocalSpectral$(G, v_{\{u\}}, 1/k)$ and obtain a cut T whose conductance is quadratically close to the optimal value $\lambda(G, v_{\{u\}}, 1/k)$. By Lemma 5, this implies that $\phi(T) \leq O(\sqrt{\phi(u,k)})$. This argument proves the following theorem.

Theorem 6 (Finding a Cut) *Given an unweighted graph $G = (V,E)$, a vertex $u \in V$ and a positive integer k, we can find a cut in G of conductance at most $O(\sqrt{\phi(u,k)})$ by computing a sweep cut of the optimal vector for LocalSpectral$(G, v_{\{u\}}, 1/k)$. Moreover, this algorithm runs in nearly-linear time in the size of the graph.*

That is, this theorem states that we can perform a sweep cut over the vector that is the solution to LocalSpectral$(G, v_{\{u\}}, 1/k)$ in order to obtain a locally-biased partition; and that this partition comes with quality-of-approximation guarantees analogous to that provided for the global problem Spectral(G) by Cheeger's inequality.

Our final theorem shows that the optimal value of LocalSpectral also provides a lower bound on the conductance of *other cuts*, as a function of how well-correlated they are with the input seed vector. In particular, when the seed vector corresponds to a cut U, this result allows us to lower bound the conductance of an arbitrary cut T, in terms of the correlation between U and T. The proof of this theorem also uses in an essential manner the duality properties that were used in the proof of Theorem 1.

Theorem 7 (Cut Improvement) *Let G be a graph and $s \in \mathbb{R}^n$ be such that $s^T D_G 1 = 0$, where D_G is the degree matrix of G. In addition, let $\kappa \geq 0$ be a correlation parameter. Then, for all sets $T \subseteq V$ such that $\kappa' \stackrel{\text{def}}{=} (s^T D_G v_T)^2$, we have that*

$$\phi(T) \geq \begin{cases} \lambda(G,s,\kappa) & \text{if } \kappa \leq \kappa' \\ \kappa'/\kappa \cdot \lambda(G,s,\kappa) & \text{if } \kappa' \leq \kappa. \end{cases}$$

Proof It follows from Theorem 1 that $\lambda(G,s,\kappa)$ is the same as the optimal value of $\mathsf{SDP}_p(G,s,\kappa)$ which, by strong duality, is the same as the optimal value of $\mathsf{SDP}_d(G,s,\kappa)$. Let $\alpha^\star,\beta^\star$ be the optimal dual values to $\mathsf{SDP}_d(G,s,\kappa)$. Then, from the dual feasibility constraint $L_G - \alpha^\star L_{K_n} - \beta^\star (D_G s)(D_G s)^T \succeq 0$, it follows that

$$s_T^T L_G s_T - \alpha^\star s_T^T L_{K_n} s_T - \beta^\star (s^T D_G s_T)^2 \geq 0.$$

Notice that since $s_T^T D_G 1 = 0$, it follows that $s_T^T L_{K_n} s_T = s_T^T D_G s_T = 1$. Further, since $s_T^T L_G s_T = \phi(T)$, we obtain, if $\kappa \leq \kappa'$, that

$$\phi(T) \geq \alpha^\star + \beta^\star (s^T D_G s_T)^2 \geq \alpha^\star + \beta^\star \kappa = \lambda(G,s,\kappa).$$

If on the other hand, $\kappa' \leq \kappa$, then

$$\phi(T) \geq \alpha^\star + \beta^\star (s^T D_G s_T)^2 \geq \alpha^\star + \beta^\star \kappa \geq \kappa'/\kappa \cdot (\alpha^\star + \beta^\star \kappa) = \kappa'/\kappa \cdot \lambda(G,s,\kappa).$$

Note that strong duality was used here. ■

Thus, although the relaxation guarantees of Lemma 5 only hold when the seed set is a single vertex, we can use Theorem 7 to consider the following problem: given a graph G and a cut $(T,\bar{T})$ in the graph, find a cut of minimum conductance in G which is well-correlated with T or certify that there is none. Although one can imagine many applications of this primitive, the main application that motivated this work was to explore clusters nearby or around a given *seed set* of nodes in data graphs. This will be illustrated in our empirical evaluation in Section 5.

4.3 Our Geometric Notion of Correlation Between Cuts

Here we pause to make explicit the geometric notion of correlation between cuts (or partitions, or sets of nodes) that is used by LocalSpectral, and that has already been used in various guises in previous sections. Given a cut $(T,\bar{T})$ in a graph $G=(V,E)$, a natural vector in $\mathbb{R}^V$ to associate with it is its characteristic vector, in which case the correlation between a cut $(T,\bar{T})$ and another cut $(U,\bar{U})$ can be captured by the inner product of the characteristic vectors of the two cuts. A somewhat more refined vector to associate with a cut is the vector obtained after removing from the characteristic vector its projection along the all-ones vector. In that case, again, a notion of correlation is related to the inner product of two such vectors for two cuts. More precisely, given a set of nodes $T \subseteq V$, or equivalently a cut $(T,\bar{T})$, one can define the unit vector s_T as

$$s_T(i) = \begin{cases} \sqrt{\mathrm{vol}(T)\mathrm{vol}(\bar{T})/2m} \cdot 1/\mathrm{vol}(T) & \text{if } i \in T \\ -\sqrt{\mathrm{vol}(T)\mathrm{vol}(\bar{T})/2m} \cdot 1/\mathrm{vol}(\bar{T}) & \text{if } i \in \bar{T}. \end{cases}$$

That is, $s_T \stackrel{\text{def}}{=} \sqrt{\frac{\mathrm{vol}(T)\mathrm{vol}(\bar{T})}{2m}} \left(\frac{1_T}{\mathrm{vol}(T)} - \frac{1_{\bar{T}}}{\mathrm{vol}(\bar{T})} \right)$, which is exactly the vector defined in Section 4.1. It is easy to check that this is well defined: one can replace s_T by $s_{\bar{T}}$ and the correlation remains the same with any other set. Moreover, several observations are immediate. First, defined this way, it immediately follows that $s_T^T D_G 1 = 0$ and that $s_T^T D_G s_T = 1$. Thus, $s_T \in \mathcal{S}_D$ for $T \subseteq V$, where we denote by $\mathcal{S}_D$ the set of vectors $\{x \in \mathbb{R}^V : x^T D_G 1 = 0\}$; and s_T can be seen as an appropriately normalized version of the vector consisting of the uniform distribution over T minus the uniform distribution over $\bar{T}$.[1] Second, one can introduce the following measure of correlation between two sets of nodes, or equivalently between two cuts, say a cut $(T,\bar{T})$ and a cut $(U,\bar{U})$:

$$K(T,U) \stackrel{\text{def}}{=} (s_T D_G s_U)^2.$$

The proofs of the following simple facts regarding $K(T,U)$ are omitted: $K(T,U) \in [0,1]$; $K(T,U) = 1$ if and only if $T = U$ or $\bar{T} = U$; $K(T,U) = K(\bar{T},U)$; and $K(T,U) = K(T,\bar{U})$. Third, although we have described this notion of geometric correlation in terms of vectors of the form $s_T \in \mathcal{S}_D$ that represent partitions $(T,\bar{T})$, this correlation is clearly well-defined for other vectors $s \in \mathcal{S}_D$ for which there is not such a simple interpretation in terms of cuts. Indeed, in Section 3 we considered the case that s was an arbitrary vector in $\mathcal{S}_D$, while in the first part of Section 4.2 we considered the case that s was the seed set of a single node. In our empirical evaluation in Section 5, we will consider both of these cases as well as the case that s encodes the correlation with cuts consisting of multiple nodes.

5. Empirical Evaluation

In this section, we provide an empirical evaluation of LocalSpectral by illustrating its use at finding and evaluating locally-biased low-conductance cuts, that is, sparse cuts or good clusters, around an input seed set of nodes in a data graph. We start with a brief discussion of a very recent and pictorially-compelling application of our method to a computer vision problem; and then we discuss in detail how our method can be applied to identify clusters and communities in a more heterogeneous and more difficult-to-visualize social network application.

5.1 Semi-Supervised Image Segmentation

Subsequent to the initial dissemination of the technical report version of this paper, Maji, Vishnoi, and Malik (2011) applied our methodology to the problem of finding locally-biased cuts in a computer vision application. Recall that image segmentation is the problem of partitioning a digital image into segments corresponding to significant objects and areas in the image. A standard approach consists in converting the image data into a similarity graph over the the pixels and applying a graph partitioning algorithm to identify relevant segments. In particular, spectral methods have been popular in this area since the work of Shi and Malik (2000), which used the second eigenvector of the graph to approximate the so-called normalized cut (which, recall, is an objective measure for image segmentation that is practically equivalent to conductance). However, a difficulty in applying the normalized cut method is that in many cases global eigenvectors may fail to capture important local segments of the image. The reason for this is that they aggressively optimize a global objective function and thus they tend to combine multiple segments together; this is illustrated pictorially in the first row of Figure 4.

This difficulty can be overcome in a semi-supervised scenario by using our LocalSpectral method. Specifically, one often has a small number of "ground truth" labels that correspond to known segments, and one is interested in extracting and refining the segments in which those labels reside. In this case, if one considers an input seed corresponding to a small number of pixels within a target

1. Notice also that $s_T = -s_{\bar{T}}$. Thus, since we only consider quadratic functions of s_T, we can consider both s_T and $s_{\bar{T}}$ to be representative vectors for the cut $(T,\bar{T})$.

object, then LocalSpectral can recover the corresponding segment with high precision. This is illustrated in the second row of Figure 4. This computer vision application of our methodology was motivated by a preliminary version of this paper, and it was described in detail and evaluated against competing algorithms by Maji, Vishnoi, and Malik (2011). In particular, they show that LocalSpectral achieves a performance superior to that of other semi-supervised segmentation algorithms (Yu and Shi, 2002; Eriksson, Olsson, and Kahl, 2007); and they also show how LocalSpectral can be incorporated in an unsupervised segmentation pipeline by using as input seed distributions obtained by an object-detector algorithm (Bourdev, Maji, Brox, and Malik, 2010).

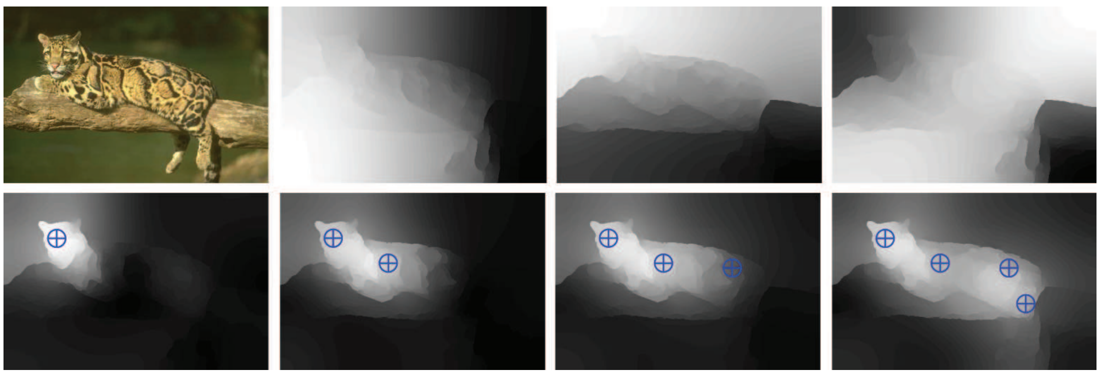


Figure 4: The first row shows the input image and the three smallest eigenvectors of the Laplacian of the corresponding similarity graph computed using the intervening contour cue (Maire, Arbelaez, Fowlkes, and Malik, 2008). Note that no sweep cut of these eigenvectors reveals the leopard. The second row shows the results of LocalSpectral with a setting of $\gamma = -10\lambda_2(G)$ with the seed pixels highlighted by crosshairs. Note how one can to recover the leopard by using a seed vector representing a set of only 4 pixels. In addition, note how the first seed pixel allows us to capture the head of the animal, while the other seeds help reveal other parts of its body.

5.2 Detecting Communities in Social Networks

Finding local clusters and meaningful locally-biased communities is also of interest in the analysis of large social and information networks. A standard approach to finding clusters and communities in many network analysis applications is to formalize the idea of a good community with an "edge counting" metric such as conductance or modularity and then to use a spectral relaxation to optimize it approximately (Newman, 2006b,a). For many very large social and information networks, however, there simply do not exist good large global clusters, but there do exist small meaningful local clusters that may be thought of as being nearby prespecified seed sets of nodes (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010). In these cases, a local version of the global spectral partitioning problem is of interest, as was shown by Leskovec, Lang, and Mahoney (2010). Typical networks are very large and, due to their expander-like properties, are not easily-visualizable (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009). Thus, in order to illustrate the empirical behavior of our LocalSpectral methodology in a "real" network application related to the one that motivated this work (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010), we examined a small "coauthorship network" of scientists. This network was previously used by Newman (2006b) to study community structure in small social and information networks.

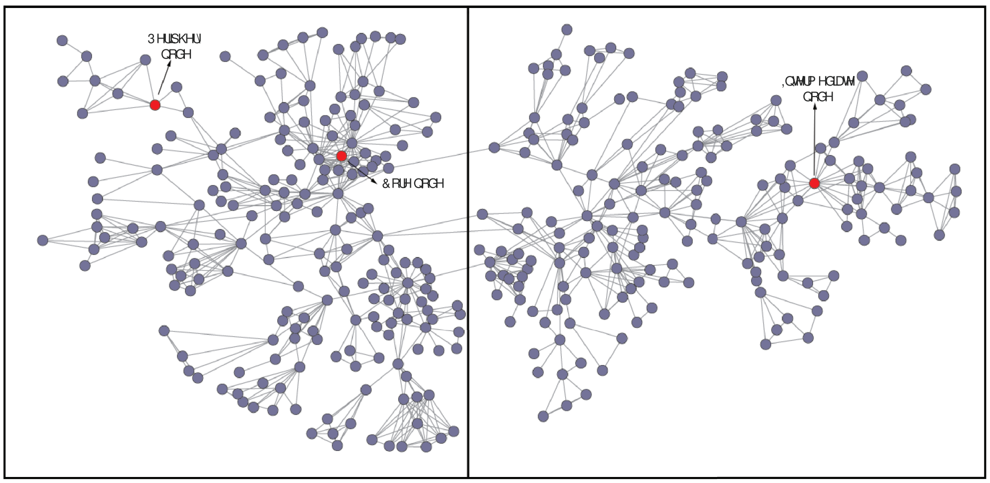


Figure 5: The coauthorship network of Newman (2006b). This layout was obtained in the Pajek (Batagelj and Mrvar, 2001) visualization software, using the Kamada-Kawai method (Kamada and Kawai, 1989) on each component of a partition provided by LocalCut and tiling the layouts at the end. Boxes show the two main global components of the network, which are displayed separately in subsequent figures.

The corresponding graph G is illustrated in Figure 5 and consists of 379 nodes and 914 edges, where each node represents an author and each unweighted edge represents a coauthorship relationship. The spectral gap $\lambda_2(G) = 0.0029$; and a sweep cut of the eigenvector corresponding to this second eigenvalue yields the globally-optimal spectral cut separating the graph into two well-balanced partitions, corresponding to the left half and the right half of the network, as shown in Figure 5. Our main empirical observations, described in detail in the remainder of this section, are the following.

- First, we show how varying the teleportation parameter allows us to detect low-conductance cuts of different volumes that are locally-biased around a prespecified seed vertex; and how this information, aggregated over multiple choices of teleportation, can improve our understanding of the network structure in the neighborhood of the seed.

- Second, we demonstrate the more general usefulness of our definition of a *generalized* Personalized PageRank vector (where the γ parameter in Equation (1) can be $\gamma \in (-\infty, \lambda_2(G))$ by displaying specific instances in which that vector is more effective than the usual Personalized PageRank (where only positive teleportation probabilities are allowed and thus where γ must be negative). We do this by detecting a wider range of low-conductance cuts at a given volume and by interpolating smoothly between very locally-biased solutions to LocalSpectral and the global solution provided by the Spectral program.

- Third, we demonstrate how our method can find low-conductance cuts that are well-correlated to more general input seed vectors by demonstrating an application to the detection of sparse peripheral regions, for example, regions of the network that are well-correlated with low-degree nodes. This suggests that our method may find applications in leveraging feature data, which are often associated with the vertices of a data graph, to find interesting and meaningful cuts.

We emphasize that the goal of this empirical evaluation is to illustrate how our proposed methodology can be applied in real applications; and thus we work with a relatively easy-to-visualize example of a small social graph. This will allow us to illustrate how the "knobs" of our proposed method can be used in practice. In particular, the goal is not to illustrate that our method or heuristic variants of it or other spectral-based methods scale to much larger graphs—this latter fact is by now well-established (Andersen and Lang, 2006; Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010).

5.2.1 Algorithm Description and Implementation

We refer to our cut-finding algorithm, which will be used to guide our empirical study of finding and evaluating cuts around an input seed set of nodes and which is a straightforward extension of the algorithm referred to in Theorem 6, as LocalCut. In addition to the graph, the input parameters for LocalCut are a seed vector s (e.g., corresponding to a single vertex v), a teleportation parameter γ, and (optionally) a size factor c. Then, LocalCut performs the following steps.

- First, compute the vector $x^\star$ of Equation (1) with seed s and teleportation γ.
- Second, either perform a sweep of the vector $x^\star$, for example, consider each of the n cuts defined by the vector and return the the minimum conductance cut found along the sweep; or consider only sweep cuts along the vector $x^\star$ of volume at most $c \cdot k_\gamma$, where $k_\gamma = 1/\kappa_\gamma$, that contain the input vertex v, and return the minimum conductance cut among such cuts.

By Theorem 1, the vector computed in the first step of LocalCut, $x^\star$, is an optimal solution to LocalSpectral(G,s,κ_γ) for some choice of κ_γ. (Indeed, by fixing the above parameters, the κ parameter is fixed implicitly.) Then, by Theorem 6, when the vector $x^\star$ is rounded (to, for example, $\{-1,+1\}$) by performing the sweep cut, provably-good approximations are guaranteed. In addition, when the seed vector corresponds to a single vertex v, it follows from Lemma 5 that $x^\star$ yields a lower bound to the conductance of cuts that contain v and have less than a certain volume k_γ.

Although the full sweep-cut rounding does not give a specific guarantee on the volume of the output cut, empirically we have found that it is often possible to find small low-conductance cuts in the range dictated by k_γ. Thus, in our empirical evaluation, we also consider volume-constrained sweep cuts (which departs slightly from the theory but can be useful in practice). That is, we also introduce a new input parameter, a *size factor* $c > 0$, that regulates the maximum volume of the sweep cuts considered when s represents a single vertex. In this case, LocalCut does not consider all n cuts defined by the vector $x^\star$, but instead it considers only sweep cuts of volume at most $c \cdot k_\gamma$ that contain the vertex v. (Note that it is a simple consequence of our optimization characterization that the optimal vector has sweep cuts of volume at most k_γ containing v.) This new input parameter turns out to be extremely useful in exploring cuts at different sizes, as it neglects sweep cuts of low conductance at large volume and allows us to pick out more local cuts around the seed vertex.

In our first two sets of experiments, summarized in Sections 5.2.2 and 5.2.3, we used single-vertex seed vectors, and we analyzed the effects of varying the parameters γ and c, as a function of the location of the seed vertex in the input graph. In the last set of experiments, presented in Section 5.2.4, we considered more general seed vectors, including both seed vectors that correspond to multiple nodes, that is, to cuts or partitions in the graph, as well as seed vectors that do not have an obvious interpretation in terms of input cuts. We implemented our code in a combination of MATLAB and C++, solving linear systems using the Stabilized Biconjugate Gradient Method (van der Vorst, 1992) provided in MATLAB 2006b. On this particular coauthorship network, and on a Dell PowerEdge 1950 machine with 2.33 GHz and 16GB of RAM, the algorithm ran in less than a few seconds.

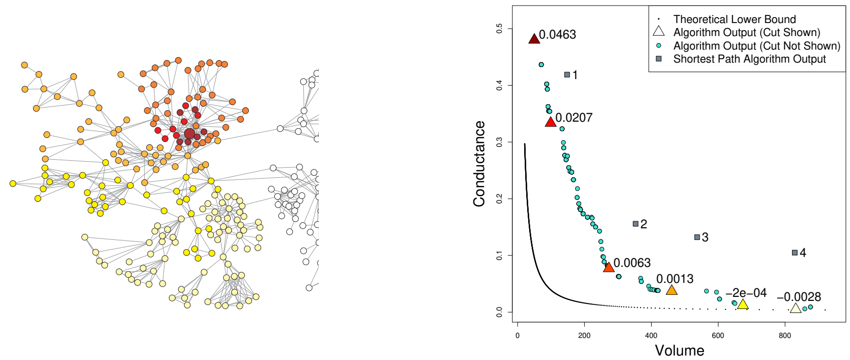
5.2.2 Varying the Teleportation Parameter

Here, we evaluate the effect of varying the teleportation parameter $\gamma \in (-\infty, \lambda_2(G))$, where recall $\lambda_2(G) = 0.0029$. Since it is known that large social and information networks are quite heterogeneous and exhibit a very strong "nested core-periphery" structure (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010), we perform this evaluation by considering the behavior of LocalCut when applied to three types of seed nodes, examples of which are the highlighted vertices in Figure 5. These three nodes were chosen to represent three different types of nodes seen in larger networks: a *periphery-like node*, which belongs to a lower-degree and less expander-like part of the graph, and which tends to be surrounded by lower-conductance cuts of small volume; a *core-like node*, which belongs to a denser and higher-conductance or more expander-like part of the graph; and an *intermediate node*, which belongs to a regime between the core-like and the periphery-like regions.

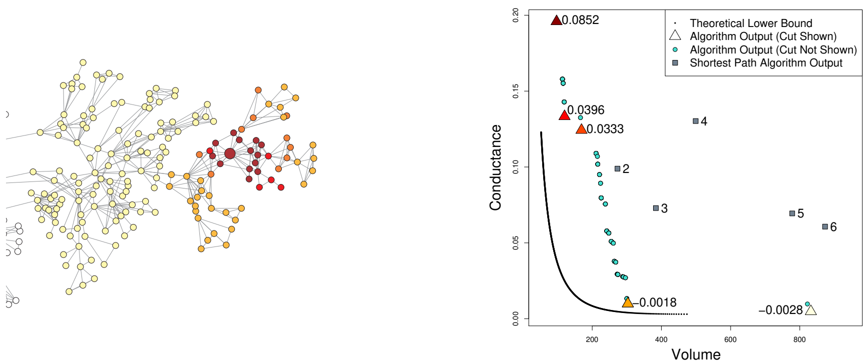
For each of the three representative seed nodes, we executed 1000 runs of LocalCut with $c = 2$ and γ varying by 0.001 increments. Figure 6 displays, for each of these three seeds, a plot of the conductance as a function of volume of the cuts found by each run of LocalCut. We refer to this type of plot as a *local profile plot* since it is a specialization of the *network community profile plot* (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010) to cuts around the specified seed vertex. In addition, Figure 6 also plots several other quantities of interest: first, the volume and conductance of the theoretical lower bound yielded by each run; second, the volume and conductance of the cuts defined by the shortest-path balls (in squares and numbered according to the length of the path) around each seed (which should and do provide a sanity-check upper bound); third, next to each of the plots, we present a color-coded/grayscale image of representative cuts detected by LocalCut; and fourth, for each of the cuts illustrated on the left, a color-coded/grayscale triangle and the numerical value of $-\gamma$ is shown on the right.

Several points about the behavior of the LocalCut algorithm as a function of the location of the input seed node and that are illustrated in Figure 6 are worth emphasizing.

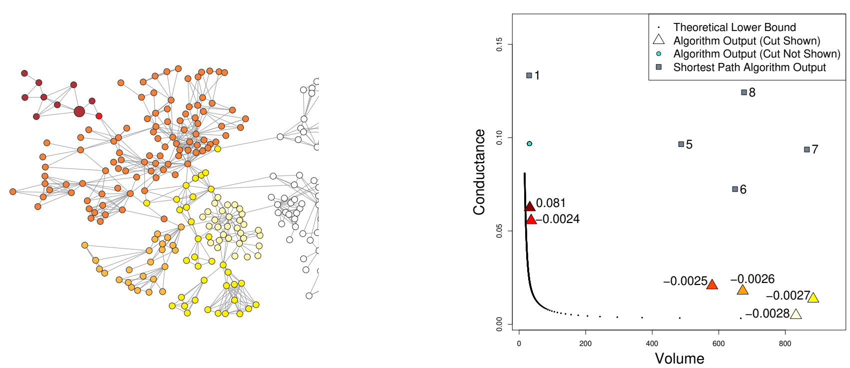
- First, for the core-like node, whose profile plot is shown in Figure 6(a), the volume of the output cuts grows relatively smoothly as γ is increased (i.e., as $-\gamma$ is decreased). For small



(a) Selected cuts and profile plot for the *core-like node*.



(b) Selected cuts and profiles plot for the *intermediate node*.



(c) Selected cuts and profile plot for the *periphery-like node*.

Figure 6: Selected cuts and local profile plots for varying γ. The cuts on the left are displayed by assigning to each vertex a color corresponding to the smallest selected cut in which the vertex was included. Smaller cuts are darker, larger cuts are lighter; and the seed vertex is shown slightly larger. Each profile plot on the right shows results from 1000 runs of LocalCut, with $c = 2$ and γ decreasing in 0.001 increments starting at 0.0028. For each color-coded/grayscale triangle, corresponding to a cut on the left, $-\gamma$ is also listed.

γ, for example, $\gamma = -0.0463$ or $\gamma = -0.0207$, the output cuts are forced to be small and hence display high conductance, as the region around the node is somewhat expander-like. By decreasing the teleportation, the conductance progressively decreases, as the rounding starts to hit nodes in peripheral regions, whose inclusion only improves conductance (since it increases the cut volume without adding many additional cut edges). In this case, this phenomena ends at $\gamma = -0.0013$, when a cut of conductance value close to that of the global optimum is found. (After that, larger and slightly better conductance cuts can still be found, but, as discussed below, they require $\gamma > 0$.)

- Second, a similar interpretation applies to the profile plot of the intermediate node, as shown in Figure 6(b). Here, however, the global component of the network containing the seed has smaller volume, around 300, and a very low conductance (again, requiring $\gamma > 0$). Thus, the profile plot *jumps* from this cut to the much larger eigenvector sweep cut, as will be discussed below.

- Third, a more extreme case is that of the periphery-like node, whose profile plot is displayed in Figure 6(c). In this case, an initial increase in γ does *not* yield larger cuts. This vertex is contained in a small-volume cut of low conductance, and thus diffusion-based methods get "stuck" on the small side of the cut. The only cuts of lower conductance in the network are those separating the global components, which can only be accessed when $\gamma > 0$. Hence, the teleportation must be greatly decreased before the algorithm starts outputting cuts at larger volumes. (As an aside, this behavior is also often seen with so-called "whiskers" in much larger social and information networks (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010).)

In addition, several general points that are illustrated in Figure 6 are worth emphasizing about the behavior of our algorithm.

- First, LocalCut found low-conductance cuts of different volumes around each seed vertex, outperforming the shortest-path algorithm (as it should) by a factor of roughly 4 in most cases. However, the results of LocalCut still lie away from the lower bound, which is also a factor of roughly 4 smaller at most volumes.

- Second, consider the range of the teleportation parameter necessary for the LocalCut algorithm to discover the well-balanced globally-optimal spectral partition. In all three cases, it was necessary to make γ positive (i.e., $-\gamma$ negative) to detect the well-balanced global spectral cut. Importantly, however, the quantitative details depend strongly on whether the seed is core-like, intermediate, or periphery-like. That is, by *formally* allowing "negative teleportation" probabilities, which correspond to $\gamma > 0$, the use of *generalized* Personalized PageRank vectors as an exploratory tool is much stronger than the usual Personalized PageRank (Andersen, Chung, and Lang, 2006; Andersen and Lang, 2006), in that it permits one to find a larger class of clusters, up to and including the global partition found by the solution to the global Spectral program. Relatedly, it provides a smooth interpolation between Personalized PageRank and the second eigenvector of the graph. Indeed, for $\gamma = 0.0028 \approx \lambda_2(G)$, LocalCut outputs the same cut as the eigenvector sweep cut for all three seeds.

- Third, recall that, given a teleportation parameter γ, the rounding step selects the cut of smallest conductance along the sweep cut of the solution vector. (Alternatively, if volume-constrained sweeps are considered, then it selects the cut of smallest conductance among sweep cuts of volume at most $c \cdot k_\gamma$, where k_γ is the lower bound obtained from the optimization program.) In either case, increasing γ can lead LocalCut to pick out larger cuts, but it does not *guarantee* this will happen. In particular, due to the local topology of the graph, in many instances there may *not* be a way of slightly increasing the volume of a cut while slightly decreasing its conductance. In those cases, LocalCut may output the same small sweep cut for a range of teleportation parameters until a much larger, much lower-conductance cut is then found. The presence such horizontal and vertical *jumps* in the local profile plot conveys useful information about the structure of the network in the neighborhood of the seed at different size scales, illustrating that the practice follows the theory quite well.

5.2.3 Varying the Output-Size Parameter

Here, we evaluate the effect of varying the size factor c, for a fixed choice of teleportation parameter γ. (In the previous section, c was fixed at $c = 2$ and γ was varied.) We have observed that varying c, like varying γ, tends to have the effect of producing low-conductance cuts of different volumes around the seed vertex. Moreover, it is possible to obtain low-conductance large-volume cuts, even at lower values of the teleportation parameter, by increasing c to a sufficiently large value. This is illustrated in Figure 7, which shows the result of varying c with the core-like node as the seed and $-\gamma = 0.02$. Figure 6(a) illustrated that when $c = 2$, this setting only yielded a cut of volume close to 100 (see the red/darker triangle with $-\gamma = 0.0207$); but the yellow/lighter crosses in Figure 7 illustrate that by allowing larger values of c, better conductance cuts of larger volume can be obtained.

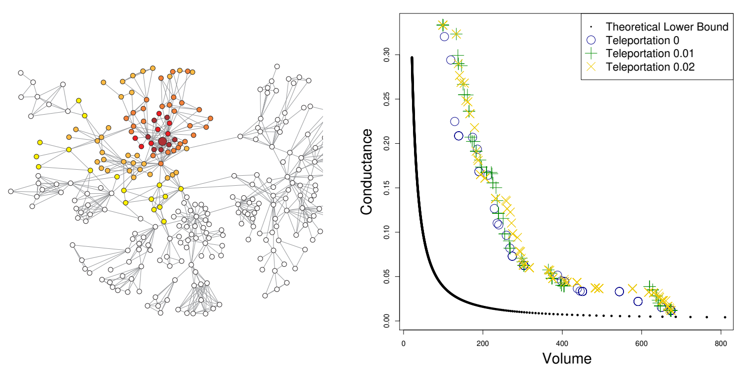


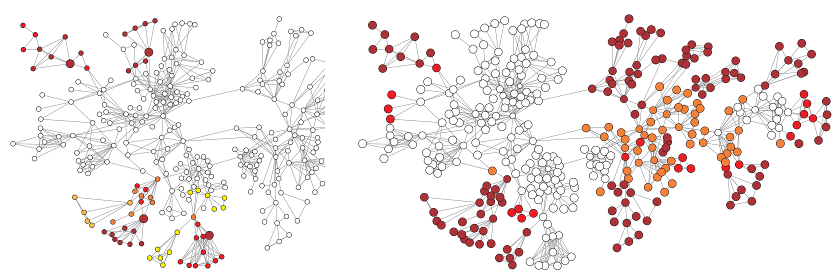
Figure 7: Selected cuts and local profile plots for varying c with the core-like node as the seed. The cuts are displayed by assigning to each vertex a color corresponding to the smallest selected cut in which the vertex was included. Smaller cuts are darker, larger are lighter. The seed vertex is shown larger. The profile plot shows results from 1000 runs of LocalCut, with varying c and $-\gamma \in \{0, 0.01, 0.02\}$.

While many of these cuts tend to have conductance slightly worse than the best found by varying the teleportation parameter, the observation that cuts of a wide range of volumes can be obtained with a single value of γ leaves open the possibility that there exists a single choice of teleportation parameter γ that produces good low-conductance cuts at all volumes simply by varying c. (This would allow us to only solve a single optimization problem and still find cuts of different volumes.) To address (and rule out) this possibility, we selected three choices of the teleportation parameter for each of the three seed nodes, and then we let c vary. The resulting output cuts for the core-like node as the seed are plotted in Figure 7. (The plots for the other seeds are similar and are not displayed.) Clearly, no single teleportation setting dominates the others: in particular, at volume 200 the lowest-conductance cut was produced with $-\gamma = 0.02$; at volume 400 it was produced with $-\gamma = 0.01$; and at volume 600 with it was produced with $\gamma = 0$. The highest choice of $\gamma = 0$ performed marginally better overall, recording lowest conductance cuts at both small and large volumes. That being said, the results of all three settings roughly track each other, and cuts of a wide range of volumes were able to be obtained by varying the size parameter c.

These and other empirical results suggest that the best results are achieved when we vary both the teleportation parameter and the size factor. In addition, the use of multiple teleportation choices have the side-effect advantage of yielding multiple lower bounds at different volumes.

5.2.4 MULTIPLE SEEDS AND CORRELATION

Here, we evaluate the behavior of LocalCut on more general seed vectors. We consider two examples—for the first example, there is an interpretation as a cut or partition consisting of multiple nodes; while the second example does not have any immediate interpretation in terms of cuts or partitions.



(a) Seed set of four seed nodes. (b) A more general seed vector.

Figure 8: Multiple seeds and correlation. 8(a) shows selected cuts for varying γ with the seed vector corresponding to a subset of 4 vertices lying in the periphery-like region of the network. 8(b) shows selected cuts for varying γ with the seed vertex equal to a normalized version of the degree vector. In both cases, the cuts are displayed by assigning to each vertex a color corresponding to the smallest selected cut in which the vertex was included. Smaller cuts are darker, larger are lighter.

In our first example, we consider a seed vector representing a subset of four nodes, located in different peripheral branches of the left half of the global partition of the the network: see the four slightly larger (and darker) vertices in Figure 8(a). This is of interest since, depending on the size-scale at which one is interested, such sets of nodes can be thought of as either "nearby" or "far apart." For example, when viewing the entire graph of 379 nodes, these four nodes are all close, in that they are all on the left side of the optimal global spectral partition; but when considering smaller clusters such as well-connected sets of 10 or 15 nodes, these four nodes are much farther apart. In Figure 8(a), we display a selection of the cuts found by varying the teleportation, with $c = 2$. The smaller cuts tend to contain the branches in which each seed node is found, while larger cuts start to incorporate nearby branches. Not shown in the color-coding/grayscale is that the optimal global spectral partition is eventually recovered. Identifying peripheral areas that are well-separated from the rest of the graph is a useful primitive in studying the structure of social networks (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010); and thus, this shows how LocalCut may be used in this context, when some periphery-like seed nodes of the graph are known.

In our second example, we consider a seed vector that represents a feature vector on the vertices but that does not have an interpretation in terms of cuts. In particular, we consider a seed vector that is a normalized version of the degree distribution vector. Since nodes that are periphery-like tend to have lower degree than those that are core-like (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010), this choice of seed vector biases LocalCut towards cuts that are well-correlated with periphery-like and low-degree vertices. A selection of the cuts found on this seed vector when varying the teleportation with $c = 2$ is displayed in Figure 8(b). These cuts partition the network naturally into three well-separated regions: a sparser periphery-like region in darker colors, a lighter-colored intermediate region, and a white dense core-like region, where higher-degree vertices tend to lie. Clearly, this approach could be applied more generally to find low-conductance cuts that are well-correlated with a known feature of the node vector.

6. Discussion

In this final section, we provide a brief discussion of our results in a broader context.

6.1 Relationship to Local Graph Partitioning

Recent theoretical work has focused on using spectral ideas to find good clusters nearby an input seed set of nodes (Spielman and Teng, 2004; Andersen, Chung, and Lang, 2006; Chung, 2007). In particular, local graph partitioning—roughly, the problem of finding a low-conductance cut in a graph in time depending only on the volume of the output cut—was introduced by Spielman and Teng (2004). They used random walk based methods; and they used this as a subroutine to give a nearly linear-time algorithm for outputting balanced cuts that match the Cheeger Inequality up to polylog factors. In our language, a local graph partitioning algorithm would start a random walk at a seed node, truncating the walk after a suitably chosen number of steps, and outputting the nodes visited by the walk. This result was improved by Andersen, Chung, and Lang (2006) by performing a truncated Personalized PageRank computation. These and subsequent papers building on them were motivated by local graph partitioning (Chung, 2007), but they do not address the problem of discovering cuts near general seed vectors, as do we, or of generalizing the second eigenvector of the Laplacian. Moreover, these approaches are more operationally-defined, while ours is axiomatic and optimization-based. For more details on these issues as well as the relationship between these issues and implementing regularization implicitly via approximate computation, see Mahoney and Orecchia (2011) and Mahoney (2012).

6.2 Relationship to Empirical Work on Community Structure

Recent empirical work has used Personalized PageRank, a particular variant of a local random walk, to characterize very finely the clustering and community structure in a wide range of very large social and information networks (Andersen and Lang, 2006; Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010). In particular, Andersen and Lang used local spectral methods to identify communities in certain informatics graphs using an input set of nodes as a seed set (Andersen and Lang, 2006). Subsequently, Leskovec, Lang, Dasgupta, and Mahoney used related methods to characterize the small-scale and large-scale clustering and community structure in a wide range of large social and information networks (Leskovec, Lang, Dasgupta, and Mahoney, 2008, 2009; Leskovec, Lang, and Mahoney, 2010). Our optimization program and empirical results suggest that this line of work can be extended to ask in a theoretically principled manner much more refined questions about graph structure near prespecified seed vectors.

6.3 Relationship to Cut-improvement Algorithms

Many recently-popular algorithms for finding minimum-conductance cuts, such as those in Khandekar, Rao, and Vazirani (2006) and in Orecchia, Schulman, Vazirani, and Vishnoi (2008), use as a crucial building block a primitive that takes as input a cut $(T,\bar{T})$ and attempts to find a lower-conductance cut that is *well correlated* with $(T,\bar{T})$. This primitive is referred to as a *cut-improvement algorithm* (Lang and Rao, 2004; Andersen and Lang, 2008), as its original purpose was limited to post-processing cuts output by other algorithms. Recently, cut-improvement algorithms have also been used to find low conductance cuts in specific regions of large graphs (Leskovec, Lang, and Mahoney, 2010). Given a notion of correlation between cuts, cut-improvement algorithms typically produce approximation guarantees of the following form: for any cut $(C,\bar{C})$ that is ε-correlated with the input cut, the cut output by the algorithm has conductance upper-bounded by a function of the conductance of $(C,\bar{C})$ and ε. This line of work has typically used flow-based techniques. For example, Gallo, Grigoriadis, and Tarjan (1989) were the first to show that one can find a subset of an input set $T \subseteq V$ with minimum conductance in polynomial time. Similarly, Lang and Rao (2004) implement a closely related algorithm and demonstrate its effectiveness at refining cuts output by other methods. Finally, Andersen and Lang (2008) give a more general algorithm that uses a small number of single-commodity maximum-flows to find low-conductance cuts not only inside the input subset T, but among all cuts which are well-correlated with $(T,\bar{T})$. Viewed from this perspective, our work may be seen as a spectral analogue of these flow-based techniques, since Theorem 7 provides lower bounds on the conductance of other cuts as a function of how well-correlated they are with the seed vector.

6.4 Alternate Interpretation of Our Main Optimization Program

There are a few interesting ways to view our local optimization problem of Figure 1 which would like to point out here. Recall that LocalSpectral may be interpreted as augmenting the standard spectral optimization program with a constraint that the output cut be well-correlated with the input seed set. To understand this program from the perspective of the dual, recall that the dual of

LocalSpectral is given by the following.

$$\begin{aligned} \text{maximize} \quad & \alpha + \beta\kappa \\ \text{s.t.} \quad & L_G \succeq \alpha L_{K_n} + \beta\Omega_T \\ & \beta \geq 0, \end{aligned}$$

where $\Omega_T = D_G s_T s_T^T D_G$. Alternatively, by subtracting the second constraint of LocalSpectral from the first constraint, it follows that

$$x^T \left(L_{K_n} - L_{K_n} s_T s_T^T L_{K_n}\right) x \leq 1 - \kappa.$$

It can be shown that

$$L_{K_n} - L_{K_n} s_T s_T^T L_{K_n} = \frac{L_{K_T}}{\mathrm{vol}(\bar{T})} + \frac{L_{K_{\bar{T}}}}{\mathrm{vol}(T)},$$

where L_{K_T} is the D_G-weighted complete graph on the vertex set T. Thus, LocalSpectral is clearly equivalent to

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t.} \quad & x^T L_{K_n} x = 1 \\ & x^T \left(\frac{L_{K_T}}{\mathrm{vol}(\bar{T})} + \frac{L_{K_{\bar{T}}}}{\mathrm{vol}(T)}\right) x \leq 1 - \kappa. \end{aligned}$$

The dual of this program is given by the following.

$$\begin{aligned} \text{maximize} \quad & \alpha - \beta(1-\kappa) \\ \text{s.t.} \quad & L_G \succeq \alpha L_{K_n} - \beta\left(\frac{L_{K_T}}{\mathrm{vol}(\bar{T})} + \frac{L_{K_{\bar{T}}}}{\mathrm{vol}(T)}\right) \\ & \beta \geq 0. \end{aligned}$$

From the perspective of this dual, this can be viewed as "embedding" a combination of a complete graph K_n and a weighted combination of complete graphs on the sets T and $\bar{T}$, that is, K_T and $K_{\bar{T}}$. Depending on the value of β, the latter terms clearly discourage cuts that substantially cut into T or $\bar{T}$, thus encouraging partitions that are well-correlated with the input cut $(T, \bar{T})$.

6.5 Bounding the Size of the Output Cut

Readers familiar with the spectral method may recall that given a graph with a small balanced cut, it is not possible, in general, to guarantee that the sweep cut procedure of Theorem 4 applied to the optimal of Spectral outputs a balanced cut. One may have to iterate several times before one gets a balanced cut. Our setting, building up on the spectral method, also suffers from this; we cannot hope, in general, to bound the size of the output cut (which is a sweep cut) in terms of the correlation parameter κ. This was the reason for considering volume-constrained sweep cuts in our empirical evaluation.

References

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.

R. Andersen and K. Lang. An algorithm for improving graph partitions. In *SODA '08: Proceedings of the 19th ACM-SIAM Symposium on Discrete algorithms*, pages 651–660, 2008.

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.

V. Batagelj and A. Mrvar. Pajek—analysis and visualization of large networks. In *Proceedings of Graph Drawing*, pages 477–478, 2001.

P. Berkhin. A survey on PageRank computing. *Internet Mathematics*, 2(1):73–120, 2005.

L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.

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

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, pages 107–117, 1998.

F.R.K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

F.R.K Chung. Four proofs of Cheeger inequality and graph partition algorithms. In *Proceedings of ICCM*, 2007.

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

G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

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.

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.

T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

R. Khandekar, S. Rao, and U. Vazirani. Graph partitioning using single commodity flows. In *STOC '06: Proceedings of the 38th annual ACM Symposium on Theory of Computing*, pages 385–390, 2006.

I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving SDD linear systems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 235–244, 2010.

K. Lang and S. Rao. A flow-based method for improving the expansion or conductance of graph cuts. In *IPCO '04: Proceedings of the 10th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 325–337, 2004.

A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.

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.

J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1): 29–123, 2009. Also available at: arXiv:0810.1355.

J. Leskovec, K.J. Lang, and M.W. Mahoney. Empirical comparison of algorithms for network community detection. In *WWW '10: Proceedings of the 19th International Conference on World Wide Web*, pages 631–640, 2010.

M. W. Mahoney. Approximate computation and implicit regularization for very large-scale data analysis. In *Proceedings of the 31st ACM Symposium on Principles of Database Systems*, pages 143–154, 2012.

M. W. Mahoney and L. Orecchia. Implementing regularization implicitly via approximate eigenvector computation. In *Proceedings of the 28th International Conference on Machine Learning*, pages 121–128, 2011.

M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

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.

M. Mihail. Conductance and convergence of Markov chains—a combinatorial treatment of expanders. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 526–531, 1989.

M.E.J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, 2006a.

M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006b.

L. Orecchia, L. Schulman, U.V. Vazirani, and N.K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 461–470, 2008.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transcations of Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

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.

H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. and Stat. Comput.*, 13(2):631–644, 1992.

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.