Graph partitioning into isolated, high conductance clusters: theory, computation and applications to preconditioning

Ioannis Koutis, Gary L. Miller
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
ioannis.koutis,glmiller@cs.cmu.edu

April 11, 2008

Abstract

We consider the problem of decomposing a weighted graph with n vertices into a collection P of vertex disjoint clusters such that, for all clusters $C \in P$, the graph induced by the vertices in C and the edges leaving C, has conductance bounded below by ϕ. We show that for planar graphs we can compute a decomposition P such that $|P| < n/\rho$, where ρ is a constant, in $O(\log n)$ parallel time with $O(n)$ work. Slightly worse guarantees can be obtained in nearly linear time for graphs that have fixed size minors or bounded genus. We show how these decompositions can be used in the first known linear work parallel construction of provably good preconditioners for the important class of fixed degree graph Laplacians. On a more theoretical note, we present spectral inequalities that give upper bounds on the Euclidean distance of eigenvectors of the normalized Laplacian from the space of vectors which consists of the cluster-wise constant vectors scaled by the square roots of the total incident weights of the vertices.

1 Introduction

Partitioning weighted graphs into disjoint and dissimilar clusters of similar vertices is arguably one of the most important algorithmic problems with applications ranging from web clustering and text retrieval, to computer aided diagnosis and computational biology. Several flavors of clustering, with respect to disparate optimization targets, have been studied in the literature (e.g. [14, 5, 22, 23, 8]). Naturally, in applications, one is interested in obtaining good clusterings with as few clusters as possible, i.e. with a large *reduction factor* ρ, defined as the number of vertices in the given graph over the number of clusters.

A particularly appealing and in practice useful way of characterizing clustering was motivated and analyzed in [16]. It is based on expansion-like properties of the graph; the quality of a clustering is characterized by the minimum *conductance* ϕ over the clusters and the ratio γ of the weight going between clusters over the total weight of the edges in the graph. An interesting property of this bicriteria measure, that we will subsequently call (ϕ, γ^{avg}) decomposition, is its connection to the well studied sparsest cut problem. As shown in [16], assuming a two-way algorithm returns a cut of sparsity at most $\sigma\phi^{\nu}$ when there is cut of sparsity ϕ, its recursive application returns - up to a logarithmic factor- a $((\phi/\sigma)^{1/\nu}, [(\sigma\gamma)^{\nu}]^{avg})$ decomposition when the graph has a (ϕ, γ^{avg}) decomposition. The complexity of the recursive algorithm analyzed in [16] is at least a logarithmic factor slower than the two-way algorithm, but it can be considerably slower because in general the two-way algorithm is not expected to return balanced cuts.

A stronger type of clusterings, that we will subsequently call (ϕ, γ) decompositions, is implicit in the laminar decompositions constructed in the context of low congestion oblivious routing [25, 3, 13]. In a (ϕ, γ) decomposition all clusters have minimum conductance ϕ, but now the for *every vertex* v the total weight incident to v that stays within v's cluster is at least a fraction γ of the total weight incident to v. The algorithms of [3, 13] both use as their basic subroutine a two-way separator algorithm, but in a far more sophisticated and expensive way than the simple recursion analyzed in [16], making the complexity at least quadratic. In particular, the algorithm of [3] gives a hierarchy of graphs $G = G_1, .., G_{O(\log n)}$ and partitions P_i, such that G_{i+1} is the contraction of the vertices of G_i with respect to the clusters in P_i. Each P_i is a $(1/(\sigma^3 \log^2 n), 1/(\sigma \log n))$-decomposition for G_i, where σ is the approximation factor provided by the two-way algorithm. The vertex reduction factor is constant in average, but there are no guarantees for the reduction factor between subsequent levels of the decomposition.

While theoretical approaches seem to suggest a top-down approach to the construction of (ϕ, γ) decompositions, in real-world applications and implementations, the most successful two-way partition software follows bottom-up approaches; they are based on multilevel iterative contractions of the graph which are done using local heuristics [17, 24]. The situation is similar in related applications like the computation of low-fill orderings for the solution of linear systems; while the best theoretical bounds come from top-down algorithms like nested dissection [10, 19, 11], the state-of-the art direct linear system solvers use variations of the local and greedy minimum degree heuristic (e.g. [7]) which in general do not have theoretical guarantees even for planar graphs [2].

The bottom-up approach has recently found its way into algorithms with strict theoretical guarantees for the solution of linear systems involving graph Laplacians. The $O(n)$ work, parallel algorithm of [18] for planar Laplacians, is based on the computation of good multi-way vertex separators. The importance of (ϕ, γ^{avg}) decompositions for dense unweighted graphs was observed by Spielman and Teng in their work in graph sparsification [28], again in the context of the solution of linear systems. Their partitioning algorithm works in a *local* fashion, targeting subgraphs of high conductance, and is the basic building block of their nearly liner time sparsification algorithm, in yet another example of a bottom-up approach. As noted in [28], it is open whether good (ϕ, γ^{avg}) decompositions can be computed in nearly linear time for general dense graphs.

1.1 Our contributions

In this paper we introduce and consider the closely related $[\phi, \rho]$ decompositions, where we require that each of the -at most n/ρ- clusters satisfies the following property: the graph induced by the vertices of each cluster and the edges leaving the cluster, has conductance bounded below by ϕ. In Section 2 we show that constant degree graphs and planar graphs have a $[\Theta(1), \Theta(1)]$ decomposition. The decomposition can be computed in $O(\log n)$ time with linear work. We also show that graphs with no K_s minor or s^2 genus have an $[\Theta(1/(s^2 \log s \log^3 n)), \Theta(1)]$ decomposition. The decomposition can be computed in $O(n \log^2 n)$ time. The main idea of our approach is the reduction to the same problem in a sparser, tree-like, spanning subgraph of the given graph.

Our approach to the problem borrows ideas from combinatorial preconditioning, the area that -motivated by problem of simplifying linear systems- studies the approximation of graphs by other simpler graphs with respect to the *condition number* metric. While our approach draws techniques from the construction of subgraph preconditioners, our motivation is -to a large extent- the fast construction of preconditioners that use extra vertices, the so-called *Steiner* preconditioners. Steiner tree preconditioners were introduced in [12]. In [20] it was shown how the laminar decomposition of [3] can be used for the construction of provably good Steiner trees. In Section 3 we present new material that extends the results of [20] from Steiner trees to more general Steiner graphs. We also show that $[\phi, \rho]$ decompositions can be used to construct provably good Steiner preconditioners. In Section 3.1 we discuss how in the particular case of constant degree graphs our ideas have a strikingly simple and embarrassingly parallel implementation with a very small hidden constant. For fixed degree graphs, this gives the first known linear work parallel construction of combinatorial preconditioners with a constant condition number. The recursive computation of $[\phi, \rho]$ decompositions leads to a laminar decomposition and a corresponding hierarchy of Steiner preconditioners. We report on preliminary experimental results on graphs with very large weight variations, derived from 3D medical scans; our results show that besides their faster construction, the Steiner preconditioners produce in practice much better condition numbers comparing to subgraph preconditioners, as predicted by prior theoretical results [21, 20].

Perhaps not surprisingly, the local partitioning algorithm of [28] as well as other heuristic variants [30, 1], exploit the connection of (ϕ, γ) decompositions with random walks. A particle doing a random walk tends to get 'trapped' in clusters of high conductance when the vertices of the cluster are connected to the exterior with relatively light edges; then the probability distribution P_v^t after a small number t of steps of the random walk starting at a given vertex v is expected to provide information about the cluster where v belongs. While this 'local' intuition can be captured mathematically, obtaining a multi-way decomposition by computing independently several such probability distributions as done in [28], is a quite complicated task when the running time must be nearly linear. In contrast, computing arbitrary distribution mixtures of the form $\sum_{v \in V} w_v P_v^t$ is straightforward and can be done in time linear in t and the number of edges in the graph. This leads to a natural 'global' question: how these distribution mixtures look in terms of the clusters of a (ϕ, γ) decomposition? We study the closely related eigenvectors of the normalized Laplacian. In Section 4 we present spectral inequalities that give upper bounds on the Euclidean distance of these eigenvectors from the space of vectors which consists of the cluster-wise constant vectors scaled by the square roots of the total incident weights of the vertices. We anticipate that this characterization may find applications in the practical computation of (ϕ, γ) decompositions for general graphs.

2 Planar decompositions

Let $G = (V, E, w)$ be a weighted graph. The Laplacian of G is the matrix A_G defined by $A_{ij} = -w_{ij}$ and $A_{ii} = \sum_{j \neq i} A_{ij}$. If $G_1 = (V, E, w_1)$, $G_2 = (V, E, w_2)$ and $G = (V, E, w_1 + w_2)$, we have $A_G = A_{G_1} + A_{G_2}$. We will often identify graphs with their Laplacians using this natural one-to-one correspondence. The total incident weight $\sum_{u \in N(v)} w(u, v)$ of vertex v is denoted by $vol(v)$. For any $V' \subseteq V$ we let $vol(V') = \sum_{v \in V'} vol(v)$, and $out(V') = \sum_{v \in V', u \notin V'} w(u, v)$. We also let

$$cap(U, V) = \sum_{u \in U, v \in V} w(u, v)$$

denote the total weight connecting the nodes of the disjoint sets U, V. The *sparsity* of an edge cut into V' and $V - V'$ is defined as the ratio

$$\frac{cap(V, V - V')}{\min(vol(V'), vol(V - V'))}.$$

The *conductance* of the graph is the minimum sparsity value over all possible cuts. Let P be a partition of the vertices of a graph $G = (V, E, w)$ into disjoint sets V_i, $i = 1, \ldots, m$ and let G_i denote the graph induced by the vertices in V_i. We call n/m the *vertex reduction factor* of P and we denote it by ρ. We call P a (ϕ, γ) decomposition if the conductance of each G_i is bounded below by ϕ and for each vertex $v \in V_i$, $cap(v, V_i - v)/vol(v) \geq \gamma$.

In this paper we consider a variant of (ϕ, γ) decompositions. For each G_i in the partition, we introduce a vertex on each edge leaving G_i. If W_i is the set of newly introduced vertices for G_i, we say that P is $[\phi, \rho]$-decomposition if the *closure* graph G_i^o induced by the vertices in $V_i \cup W_i$ has conductance bounded below by ϕ and the vertex reduction factor of P is at least ρ. By definition, G_i^o is G_i with additional degree one vertices hanging off of it. Therefore, any edge cut in G_i induces a sparser cut in G_i^o, and thus the conductance of G_i must be lower bounded by ϕ. Also note that if G_i contains two vertices v_1, v_2 such that $cap(v_j, V_i - v_j)/vol(v_j) \leq \phi$ for $j = 1, 2$, the conductance of G_i^o is less than ϕ; this can be seen by considering the edge cut consisting of the edges incident to v_1 in G_i when $vol(v_2) \geq vol(v_1)$, and vice-versa. Hence there can be no more than one vertex violating the γ constraint, if $\gamma < \phi$. So a $[\phi, \rho]$-decomposition is "almost" a (ϕ, ϕ) decomposition. It turns out that the parallel computation of $[\phi, \rho]$ decompositions is not trivial even for trees, for which we will need some machinery from parallel tree contraction algorithms [26].

Theorem 2.1. *Trees have a $[1/2, 6/5]$-decomposition that can be computed with linear work in $O(\log n)$ parallel time.*

Proof. If the tree contains 2 or 3 vertices the decomposition consists of only one cluster. The basic step of the algorithm is to compute an appropriate vertex separator of T, the so-called 3-critical vertices [26]. Given a root for the tree, a vertex v with children w_i, is defined to be 3-critical if (i) it is not a leaf, and (ii) for all i, we have $\lceil |descendants(v)|/3 \rceil > \lceil |descendants(w_i)|/3 \rceil$. The 3-critical vertices can be seen as the shared vertex boundaries of edge-disjoint connected subtrees consisting otherwise of non-critical vertices. We call these subtrees 3-*bridges*. The computation of the 3-critical vertices can be done with linear work in $O(\log n)$ parallel time using the parallel tree contraction algorithms [26].

We now describe the computation of the decomposition P. We form one cluster per critical vertex, each containing initially only the critical vertex. Assuming that T has n vertices, the number of 3-critical vertices is at most $2n/3$. Although we will allow critical vertices to be singletons in P, we will not allow non-critical vertices to be singletons. This implies that after the contraction of the clusters the tree will have at most $2n/3 + n/6$ vertices, which gives the reduction factor. By their definition and properties, the 3-critical vertices decompose the edges of T into 3-bridges of two types. *External* 3-bridges contain only one critical vertex and *internal* 3-bridges contain two critical vertices.

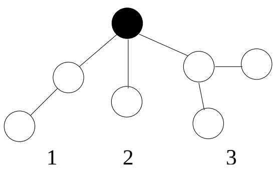

Figure 1: External 3-bridge with possible attachments.

Each external 3-bridge is formed by the critical vertex v which is the shared root of a number of trees T_i. The 4 possible cases of T_i are depicted in Figure 1, where the black vertex is the critical vertex. In cases 2,4 we form clusters with the non-critical vertices and we add them to P. The closure of these clusters has conductance 1. In case 1, we form a cluster of two vertices by cutting edge e_1 if

$w(e_1) \leq w(e_2)$. Otherwise, we form a cluster containing all three non-critical vertices. The closures of these clusters have conductance at least $1/2$. We finally add to the cluster of v its attached leaves (case 2), and the non-clustered non-critical vertices possibly left from case 1.

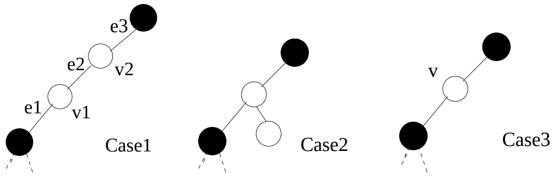

Figure 2: The possible internal 3-bridges.

An *internal* 3-bridge that is attached through two critical vertices contains at most 2 non-critical vertices. In Figure 2 we give the three possible 3-bridges of this type. In case 2, we form a cluster for the two non-critical vertices, the conductance of its closure is obviously 1. In case 3, we assign the non-critical vertex v to the cluster of the adjacent critical node which has the heaviest connection to v. Finally for case 1 we have the following subcases: (i) if $w(e_2) \leq w(e_1)$ and $w(e_2) \leq w(e_3)$, we assign v_1, v_2 to the clusters of their adjacent critical vertices, otherwise (ii) we form a cluster with v_1, v_2 and we add it to P. The closure of the cluster has conductance at least $1/2$.

We are left with the clusters of the critical vertices which we add to P. By the construction in the previous step, the closure of each cluster has the critical vertex v as a root shared by a number of edges and a number paths of the form (v, u_1, u_2), where $w(v, u_1) \geq w(u_1, u_2)$. It is then easy to see that the conductance of the closure is at least $1/2$. Finally note that after the computation of the 3-critical nodes the clustering can be done in $O(1)$ parallel time. □

We are now ready to give our results for planar graphs.

Theorem 2.2. *Planar graphs have a $[\phi, \rho]$-decomposition such that $\phi\rho$ is constant. The decomposition can be constructed with linear work in $O(\log n)$ parallel time.*

Proof. Let $A = (V, E, w)$ be any planar graph. In [18], relying on the computation of a multiway vertex separator for A, we showed that for any large enough constant k, there is a subgraph B of A with $n - 1 + cn \log^3 k/k$ edges, for some fixed constant c. Furthermore for all vectors x, we have $x^T Ax < kx^T Bx$. It is well known that a process of greedily removing degree one vertices and then replacing each path of degree two vertices by an edge, results in a graph that contains at most $4cn \log^3 k/k$ vertices (see for example [18]). Let $W \subset V$ denote the set of these vertices.

The vertices in $V - W$ either (i) lie on paths between vertices $w_i, w_j \in W$, or (ii) they belong to trees that are attached to the rest of B through a vertex $w \in W$ or through a vertex $v \in V - W$ of the first kind. This is illustrated in Figure 3.

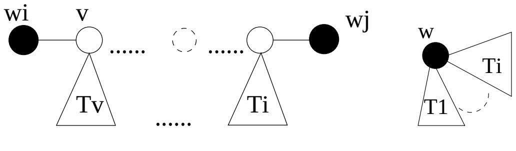

Figure 3: The organization of B-vertices that are greedily removed.

In the following we describe an algorithm to construct P, a $[\phi, \rho]$-decomposition of B, with $\phi > 1/4$ and ρ constant. We first construct an edge cut $\mathcal{C}$. Consider a path p between $w_i, w_j \in W$ including w_i and w_j. Let e be an arbitrary edge of smallest weight among the edges of p. We include e in $\mathcal{C}$. By doing this for every path p of this form, we decompose V into vertex disjoint trees each containing a unique vertex $w \in W$.

We will decompose each tree T_w independently. We describe the process for a given T_w. The removal of w disconnects T_w into a set of single vertices $R = \{r1, \dots, r_i\}$ and a number of non-trivial trees T_i with roots t_i. We form the cluster $w \cup R$ and we add it to P. The closure graph of $w \cup R$ is a star, so its conductance is 1. Now let $T_i' = T_i + (t_i, w)$ and compute P_i, the $[1/2, 6/5]$-decomposition of each T_i'. Each P_i includes exactly one cluster containing w. We remove w from its cluster in P_i and we add the cluster to P, along with the rest of the clusters in P_i. By construction, all clusters that are added to P are vertex disjoint. Note that w is a leaf in each T_i', so removing it from it cluster in P_i' does not disconnect the cluster. Hence all clusters added to P are connected. If the cluster of w in some P_i contains only two vertices, then T_i' must have at least 4 vertices, and P_i has at least 2 non-singleton clusters. This shows that the clustering gives a constant reduction factor in the number of vertices of T_i. In the worst case the vertices of W remain as singletons in P, but since $|W|$ is a constant fraction of n, the vertex reduction factor of P is constant.

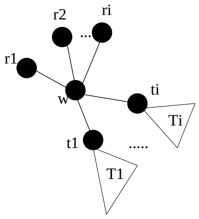

Figure 4: Computing a tree decomposition.

It remains to show that the closure of the clusters in P have conductance at least 1/4. The clusters that are not incident to an edge in $\mathcal{C}$ satisfy the constraint by construction. However we have boundary clusters each of which contains exactly one vertex which is incident to some edge in $\mathcal{C}$.

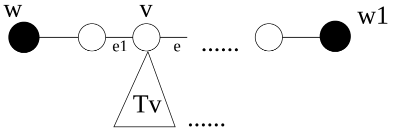

Figure 5: The boundary cluster

Assume that a cluster $U \in P$ contains a vertex v which is a adjacent to $e \in \mathcal{C}$, and let T_U be the tree induced by U. Recall that e is the lightest edge on a path between w and some $w_1 \in W$. This scenario is depicted in Figure 5. Let T_U' denote the closure of T_U restricted in the tree T_w on which we applied the $[1/2, 6/5]$-decomposition. By construction the conductance of T_U' is at least 1/2. We also have $T_U^o = T_U' + e$. Note that T_U' contains e_1. Hence the volume of v in T_U^o is at most two times its volume in T_U'. Hence adding e in T_U' can decrease its conductance by at most a factor of 2.

We finally claim that P is a $[1/(4k), \rho]$-decomposition for A. Let $A[V_i], B[V_i]$ be the graphs induced by the cluster $V_i \in P$ in A and B respectively. Let e_v be the vector which has a single non-zero entry corresponding to the vertex v. We have $e_v^T A e_v = vol_A(v)$, and similarly for B. We also know that

$e_v^T A e_v \leq k e_v^T A e_v$. It follows that the volume of v in A is at most k times its volume in B. Comparing to $B[V_i]^o$, the closure $A[V_i]^o$ contains additional edges and some additional vertices that are leaves. Since the total capacity of an edge cut in $A[V_i]^o$ can only increase with respect to the same cut in $B[V_i]^o$, the conductance can only decrease by a factor of k. The conductance of $B[V_i]^o$ is at least $1/4$, hence the conductance of $A[V_i]^o$ is at least $1/(4k)$.

The graph B can be constructed with linear work in $O(\log n)$ time as shown in [18]. The edges cut between vertices in W can also be found in $O(\log n)$ time and total linear work. The same holds for the tree decompositions. □

Theorem 2.3. *Graphs with no K_s minor or s^2 genus have a $[\Theta(1/(\log^3 ns^2 \log s)), O(1)]$ decomposition which can be computed in $O(n \log^2 n)$ time.*

Proof. In the proof of Theorem 2.2 we used a subgraph preconditioner with a condition number k and a constant fraction of non-tree edges. The condition number provided us with the bound on the conductance $\phi > 1/(4k)$, while the number of non-tree edges led to the bound on the reduction factor ρ. Theorem 3.1 of [27], in combination with the low stretch trees of [9], shows that graphs with no K_s minor or s^2 genus have a subgraph preconditioner with a constant fraction of non-tree edges and condition number $O(\log^3 ns^2 \log s)$. This preconditioner can be constructed in $O(n \log^2 n)$ time. The rest of the proof remains identical to the proof of Theorem 2.2. □

3 Steiner preconditioners

This Section requires some well known definitions and facts from the support theory for preconditioning. We refer the reader to the Appendix, and for a more complete picture to [4]. Given a graph A with n vertices, a *Steiner support graph* S for A is a graph with n vertices corresponding to the vertices of A and m extra or *Steiner* vertices. Gremban and Miller showed that Steiner graphs can be used as preconditioners [12]. The analysis of their quality can be reduced to the analysis of the support of the pair (A, B) where B is the Schur complement with respect to the Steiner vertices of S.

Gremban used the fact that $\sigma(A, S) = \sigma(A, B)$ (proposition 6.1 in [4]) to give easy bounds on $\sigma(A, B)$. In the other direction, bounding the support $\sigma(B, A)$ is a difficult task because not only B is dense, but in general it doesn't have a closed analytic expression. Generally, until the paper of Maggs et. al [20] it was not known whether there is a good Steiner tree preconditioner. However, their analysis concerns only Steiner trees. In this Section we present a way for analyzing the support for more general Steiner graphs. To keep our discussion contained and more oriented towards its applications, we focus our attention to the following definition of Steiner graphs, with respect to multi-way clusterings.

Definition 3.1. **[Quotient and Steiner graph]** *Let P be an edge cut, i.e. a partitioning of the vertices of the graph A into disjoint sets V_i, $i = 1, \ldots, m$. Let T_i be a tree with: (i) leaves corresponding to the vertices in V_i, (ii) root r_i, and (iii) for each $u \in V_i$, $w(r_i, u)$ is the total incident weight of u in A. We define the* **quotient graph** *Q on the set of the roots of the trees T_i, by letting $w(r_i, r_j) = cap(V_i, V_j)$. We define the* **Steiner graph** *with respect to P, as $S_P = Q + \sum_{i=1}^m T_i$.*

The main result of this Section is a Theorem that characterizes the support $\sigma(S_P, A)$ with respect to the parameters ϕ, γ of the decomposition P. Before we get there we need to show some Lemmas. Our results are based on the following characterization of $\sigma(B, A)$, shown in [20].

Lemma 3.2. *If S is a Steiner graph for A and B_S is Schur complement with respect to the elimination of the Steiner vertices of S, we have*

$$\sigma(B_S, A) = \max_x \min_y \left(\left(\begin{array}{c} x \\ y \end{array} \right)^T S \left(\begin{array}{c} x \\ y \end{array} \right) \right) / x^T A x$$

where $y \in \mathbb{R}^m$, and x is orthogonal to the constant vector.

Lemma 3.3. **[Steiner support transitivity]** *Let S', S be Steiner graphs for A, with the same number of vertices. Also, let $B_{S'}, B_S$ be the Schur complements with respect to the elimination of the Steiner vertices of S', S. We have $\sigma(B_S, A) \leq \sigma(S, S')\sigma(B_{S'}, A)$.*

Proof. Lemma 3.2 implies that for all vectors $x \in \mathbb{R}^n$ there is a vector $y_x \in \mathbb{R}^m$ such that

$$(x|y_x)^T S'(x|y_x) \leq \sigma(B_{S'}, A)(x^T A x).$$

By the definition of $\sigma(S, S')$ this implies that for all vectors x, we have

$$(x|y_x)^T S(x|y_x) \leq \sigma(S, S')\sigma(B_{S'}, A)(x^T A x).$$

Then, Lemma 3.2 implies directly the bound on $\sigma(B_S, A)$. □

In the following, to simplify our notation, whenever it is understood that S is a Steiner graph of A, we will denote $\sigma(B_S, A)$ by $\sigma(S, A)$.

Lemma 3.4. **[Star complement support]** *Let A be a graph with n vertices of volumes $a_1 \leq \ldots \leq a_n$ and S be the star graph with n edges corresponding to the vertices of A. Assume that for all $i \leq n-1$, the weight c_i of the i^{th} edge of S satisfies $c_i \leq \gamma^{-1} a_i$. Then if (i) $c_n \leq \gamma^{-1} a_n$ or (ii) $a_n \geq \sum_{k \leq n-1} a_k$ we have $\sigma(S, A) \leq 2/(\gamma \phi_A^2)$, where ϕ_A is the conductance of A.*

Proof. By definition we have $\sigma(S, A) = \sigma(B, A)$ where B is the Schur complement with respect to the elimination of the root of S. The edge weights for B are given by $b_{i,j} = c_i c_j / \sum_k c_k$. For the volume b_i of the vertex i in B, we have $b_i = c_i(\sum_{j \neq i} c_k)/\sum c_k \leq c_i$. If we fix c_i for $i \leq n-1$, b_n is clearly increasing in c_n and (letting c_n go to infinity) we see that

$$b_n \leq \sum_{k \leq n-1} c_k \leq \gamma^{-1} \sum_{k \leq n-1} a_k \leq \gamma^{-1} a_n.$$

Therefore in both cases (i) and (ii) we have $b_i \leq (\gamma)^{-1} a_i$ for all a_i. So, if D_G denotes the diagonal of the Laplacian G, we have $x^T D_B x \leq \gamma^{-1} x^T D_A x$ for all vectors x. Now, we have

$$\begin{aligned} \sigma(B, A) &= \max_x \frac{x^T B x}{x^T A x} = \max_x \frac{x^T D_A x}{x^T A x} \frac{x^T B x}{x^T D_B x} \frac{x^T D_B x}{x^T D_A x} \\ &\leq \gamma^{-1} \lambda_{max}(D_B^{-1} B) \lambda_{\min}^{-1}(D_A^{-1} A) \end{aligned}$$

The last inequality uses standard facts about (generalized) eigenvalues. We have $\lambda_{\min}(D_A^{-1} A) \geq \phi_A^2/2$ by the Cheeger inequality [6]. We also have $\lambda_{\max}(D_B^{-1} B) \leq 2$ by Gershgorin's theorem ([29]) and the observation that the row sums of $D_A^{-1} A$ are less than 2. This completes the proof. □

Finally, we are ready for our main result.

Theorem 3.5. *If P is a (ϕ, γ) decomposition of A then $\sigma(S_P, A) \leq 3(1 + 2/(\gamma\phi^2))$. If P is a $[\phi, \rho]$ decomposition of A then $\sigma(S_P, A) \leq 3(1 + 2/\phi^3)$.*

Proof. We first observe that

$$\sigma(S_P, A) = \sigma(S_P + A, A) - 1.$$

By construction, there is a one-to-one correspondence of the vertices in A and the leaves of S_P. Let e be an edge connecting vertices of the two clusters V_i and V_j in A. There is a unique path $p(e)$ of length 3, connecting r_i, r_j and using e in $S_P + A$. The capacities along $p(e)$ are at least $w(e)$. Thus we can route $w(e)$ units of the edge (r_i, r_j) through this path, with congestion 1. By doing this for all edges connecting V_i and V_j we get an embedding of $S_P + A$ into $S_P + A - Q$; the embedding has dilation 3, and congestion 1. Using a standard support theory argument (whose proof is based on the splitting Lemma 5.4), this proves that

$$\sigma(S_P + A, S_P + A - Q) \leq 3.$$

Observe that $S_P + A - Q$ is a forest of trees. Using the definition of support and the splitting Lemma we get,

$$\sigma(S_P + A - Q, A) = 1 + \sigma(S_P - Q, A) \leq 1 + \max_i \sigma(T_i, A_i).$$

Now if P is a (ϕ, γ) decomposition $\sigma(T_i, A_i) \leq 2/(\gamma\phi)^2$, by Lemma 3.4. We have seen in Section 2 that a $[\phi, \rho]$ decomposition is a (ϕ, ϕ) decomposition, with at most one vertex $u \in V_i$ for which $cap(u, V_i - u) \leq \phi vol(u)$. Note that since we have $cap(u, V_i - v_i) \geq \phi \min\{vol(u), vol(V - u)\}$, we must have

$$vol(u) \geq vol(V - u) = \sum_{w \in \{V - u\}} vol(w).$$

Hence $\sigma(T_i, A_i) \leq 2/\phi^3$, by the second case in Lemma 3.4. Finally, using Lemma 3.3, we have

$$\begin{aligned} \sigma(S_P, A) &\leq \sigma(S_P + A, A) \\ &\leq \sigma(S_P + A, S_P + A - Q)\sigma(S_P + A - Q, A). \end{aligned}$$

Combining the above inequalities finishes the proof. □

3.1 Preconditioning for fixed degree graphs

The ideas from Sections 2 and 3 have a very simple and fully parallel implementation in the case of fixed degree graphs. The decomposition is computed by performing the following simple steps: **[1]** From the given graph A, form the graph $\hat{A}$ by independently perturbing each edge by a random constant in $(1, 2)$. **[2]** For each vertex v keep in A the heaviest incident edge of v in $\hat{A}$, to form a subgraph B of A, which is a forest of trees. **[3]** Independently split each tree in B into clusters of size at most k for some constant k.

To see why step **[1]** generates a forest B, consider the graph $\hat{B}$ consisting of the edges of B with their weighting in $\hat{A}$. The graph $\hat{B}$ is unimodal, i.e. for each path $(v_1, v_2, \ldots, v_k)$ there is no edge (v_i, v_{i+1}) which is lighter than its two adjacent edges. This happens because (v_i, v_{i+1}) is the heavier incident edge of either v_i or v_{i+1}. From this it also follows that $\hat{B}$ and thus B are forests of trees. We claim that if the maximum degree in the graph is d, this simple process generates a $[2d^2k, 2]$ decomposition. To see this, observe that the conductance of the closure of each cluster in $\hat{B}$ is at least $1/(dk)$ by the unimodality property. This implies that the conductance of the closure of every class in A is at most $1/(2d^2k)$. The reduction factor is at least 2 because every vertex is assigned to a cluster. By Theorem 3.5 the decomposition can be used to construct a preconditioner with a constant condition number and at most $n/2$ Steiner vertices.

3.2 Effectiveness and practicality

A complete study of the new Steiner preconditioners is out of the scope of this paper. However, we would like to make some remarks on their effectiveness and practicality, based on our experiments with regular meshes and graphs generated from 3D optical coherence tomography (OCT) scans that exhibiting large edge weight variations both at a global and a local scale (due to noise).

Remark 1. All prior constructions of combinatorial preconditioners for constant degree graphs are based on the computation of a spanning tree of the graph (a maximum weight spanning tree [15], or a low stretch spanning tree [9]), and its subsequent enrichment with other edges from the graph. There is no known way to parallelize the construction.

In comparison, as described in Section 3.1, our clustering algorithm is essentially independent from the structure of the graph and can be implemented with three passes of the matrix representing the graph. Each column of the matrix can be processed independently from the rest of the columns, hence the clustering can be found completely in parallel. If R is the $n \times m$ 0-1 matrix describing the vertex-cluster memberships, the quotient graph (which is the 'main' part of the Steiner preconditioner as described in Definition 3.1) can be expressed algebraically as $Q = R^T AR$. Thus, it can be easily computed via parallel sparse matrix multiplication.

We compared a prototype sequential implementation of our construction in MATLAB, against the Boost Graph Library code for computing *only* the maximum weight spanning tree, without the subsequent addition of edges. On a 3D weighted regular grid with 10^6 vertices our code is at least 4 times faster. Naturally, a greater speed-up is expected when the edge enrichment phase and the use of parallelism come into the picture.

Remark 2. The Steiner preconditioner S_P consists of n leaves and the quotient Q which has m Steiner vertices. Preconditioning with S_P involves the solution of a linear system in S_P. Gaussian elimination of the leaf variables in S_P can be done completely independently, lending itself to a very straightforward implementation, which algebraically amounts to computing weighted cluster-wise sums. In contrast, the greedy Gaussian elimination of degree one and two nodes in subgraph preconditioners is a sequence of dependent eliminations. Although a certain amount of parallelization in their computation is possible, a parallel implementation is quite more complicated.

Remark 3. In instances of planar and 3D graphs that appear in applications, the recursive application of our simple contraction process tends to yield super-clusters that induce "round" subgraphs in the original graph. Such subgraphs are known to have Steiner trees with asymptotically better (with respect to their size) condition numbers relative to subgraph preconditioners [20, 21].

To verify the theoretical predictions, we solved a weighted 3D grid using a Steiner preconditioner and a subgraph preconditioner. The special structure of the 3D grid allowed us to bypass the monolithic spanning tree construction of [28, 9] and build the subgraph preconditioner using the more effective miniaturization ideas from [18]. In order to make the preconditioners directly comparable in terms of their condition number, we designed them so that they achieve roughly the same reduction factor (around 4) in the size of the graph/system. In Figure 6 we plot the evolution of the norm of the residual error $||r_i||_2 = ||Ax_i - b||_2$, which reflects the effectiveness of the preconditioners. Clearly, the convergence is several times faster with the Steiner preconditioner.

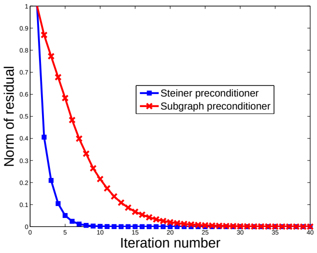

Figure 6: Steiner vs subgraph preconditioners

4 A spectral portrait of (ϕ, γ) decompositions

Let A be a Laplacian and D be the diagonal of A. The matrix $P = I - AD^{-1}$ is the probability transition matrix; the probability that a random walk starting at vertex i is at vertex j after t steps of the walk is the j^{th} entry of the vector $P^t e_i$, where $e_i(i) = 1$, and $e_i(j \neq i) = 0$. The computation of the probability distribution $P^t e_i$ can take up to linear time even for very small t. On the other hand any mixture of the distributions of different random walks $\sum_i w_i(P^t e_i) = P^t \sum_i a_i w_i$, can be computed by t matrix vector multiplications with P and the vector $w = \sum_i a_i w_i$. This can be done in time linear in t and the number of edges in the graph.

If the eigenvalues of P are $-1 < \mu_1 \leq \mu_2 \leq \ldots \leq \mu_n \leq 1$, the eigenvalues of P^t are $\mu_1^t, \mu_2^t, \ldots, \mu_n^t$. Thus, a small number $t = O(\log n)$, essentially wipes out the constant eigenvalues of P, so that $P^t w$ is "mostly" a linear combination of the eigenvectors corresponding to eigenvalues of P that are asymptotically close to 1. A slightly unpleasant fact about the eigenvectors of P is that they are not orthogonal, because P is not symmetric. Because of this we will work with the normalized Laplacian $\hat{A} = D^{-1/2} A D^{-1/2}$. It is not hard to see that if x_i is an eigenvector of $\hat{A}$ with eigenvalue λ_i then $D^{-1/2}x$ is an eigenvector P with eigenvalue $\mu_i = 1 - \lambda_i$.

Let P be a (ϕ, γ) decomposition of A, and S_P be the corresponding Steiner graph constructed in Section 3. Let B be the Schur complement of S_P and $\hat{B} = D^{-1/2} B D^{-1/2}$. We base our approach on the intuition that when the condition number $\kappa(A, B) = \kappa(\hat{A}, \hat{B})$ is small , the eigenvectors of A are expected to be good approximations of the eigenvectors of $\hat{B}$. In other words, a low frequency eigenvector of $\hat{A}$ is expected to be near orthogonal to a high frequency eigenvector of $\hat{B}$. This intuition is quantified in Theorem 5.6. The second key observation is that the 'low' frequency subspace of $\hat{B}$ has a very simple form: it consists of cluster-wise constant vectors scaled by $D^{1/2}$. Using the notation introduced so far, the exact details are given in the following Theorem.

Theorem 4.1. *Let R be an $n \times m$ matrix where $R(i, j) = 1$ if vertex i belongs to cluster j and $R(i, j) = 0$ otherwise. Let y be any vector in $Null(R^T D^{1/2})$, and x be any unit vector which is a linear combination of vectors of $\hat{A}$ corresponding to eigenvalues smaller than λ_i. We have*

$$(x^T y)^2 \leq 3\lambda_i(1 + 2(\gamma\phi^2)^{-1})).$$

Hence there is a unit vector $z \in Range(D^{1/2}R)$ such that

$$(x^T z)^2 \geq 1 - 3\lambda_i(1 + 2(\gamma\phi^2)^{-1}).$$

Proof. It can be verified that the quotient graph Q in S_P can be written algebraically as $Q = R^T AR$. Let $V = DR$ and $D_Q = R^T DR$. The support graph S_P then has the following form

$$S_P = \begin{pmatrix} D & -V \\ -V^T & Q + D_Q \end{pmatrix}.$$

Besides its algorithmic definition 5.5, it is well known that the Schur complement B of S_P with respect to the Steiner vertices, can be expressed algebraically as

$$B = D - V(Q + D_Q)^{-1}V^T.$$

Let

$$\hat{B} = D^{-1/2}BD^{-1/2} = I - D^{1/2}R(Q + D_Q)^{-1}R^T D^{1/2}.$$

It is easy to see that the subspace $Null(R^T D^{1/2})$ is an eigenspace of $\hat{B}$ with eigenvalue 1. If the decomposition P is a (ϕ, γ) decomposition, we know by Theorem 3.5 that $\lambda(B, A) \leq 3(1+2(\gamma\phi^2)^{-1})$. It is easy to see that $\lambda(\hat{B}, \hat{A}) = \lambda(B, A)$. Let x be a linear combination of eigenvectors of $\hat{A}$ with eigenvalues smaller than λ_i, and $y \in Null(R^T D^{1/2})$. Then, by applying Theorem 5.6 to $(\hat{B}, \hat{A})$ gives

$$(x^T y)^2 \leq \lambda_{\max}(\hat{B}, \hat{A})\lambda_i \leq 3\lambda_i(1 + 2(\gamma\phi^2)^{-1}).$$

Note now that if y is the projection of x into $Null(R^T D^{1/2})$ and z is its projection into $\mathcal{R}(D^{1/2}R)$, we have $x = y + z$, with $y^T z = 0$. From this, we get $\|z\|^2 = (x^T z)^2$ and $\|y\|^2 = (x^T y)^2$. Since $\|z\|^2 + \|y\|^2 = 1$ the second claim follows. □

5 Appendix

Definition 5.1. [Support and condition numbers]
The support $\sigma(A,B)$ of two Laplacians (A,B) is defined as

$$\sigma(A,B) = \min\{t \in \mathbb{R} : x^T(\tau B - A)x \geq 0, \text{for all } x \text{ and all } \tau \geq t\}.$$

The condition number is defined as

$$\kappa(A,B) = \sigma_{\max}(A,B)\sigma_{\max}(B,A).$$

Definition 5.2. [Generalized eigenvalues]
The set of generalized eigenvalues $\Lambda(A,B)$ of a pair of Laplacians is defined by

$$\Lambda(A,B) = \{\lambda : \text{there is real vector } x \text{ such that } Ax = \lambda Bx\}.$$

Lemma 5.3. [Rayleigh quotient characterization of support] *If A, B have the same size, we have*

$$\lambda_{\max}(A,B) = \sigma(A,B) = \max_{x^T j \neq 0} (x^T Ax)/(x^T Bx),$$

where j denotes the constant vector.

The following well known Lemma is central to the development of support graph theory [4].

Lemma 5.4. [The splitting Lemma]
If $A = \sum_i A_i$ and $B = \sum_i B_i$ we have

$$\sigma(A,B) \leq \max_i \sigma(A_i, B_i).$$

Definition 5.5. [Schur complement]
Let T be a weighted star with $n+1$ vertices and edge weights $d_1, \ldots, d_n$. The Schur complement *$S(T,v)$ of T with respect to its root v, is the graph defined by the weights $S_{ij}(T,v) = d_i d_j / D$ where $D = \sum_i d_i$. Let A be any graph, $A[V-v]$ be the graph induced in A by the vertices in $V - v$, and T_v be the star graph consisting of the edges incident to v in A. The Schur complement $S(A,v)$ of A with respect to vertex v is the graph $A[V-v] + S(T_v, v)$. Let $W \subset V$ and v be any vertex in W. The Schur complement with $S(A,W)$ is recursively defined as*

$$S(A,W) = S(S(A,v), W - v) = S(S(A, W - v), v).$$

Let A, B be positive definite matrices. We let $\lambda_1 \leq \ldots \leq \lambda_n$ denote the eigenvalues of A and $\mu_1 \leq \ldots \leq \mu_n$ denote the eigenvalues of B. Let $\kappa_{\max}$ and $\kappa_{\min}$ denote $\lambda_{\max}(A,B)$ and $\lambda_{\min}(A,B)$. We therefore have $\lambda_{\max}(B,A) = 1/\kappa_{\min}$ and $\lambda_{\min}(B,A) = 1/\kappa_{\max}$.

Theorem 5.6. *Let $\mathcal{X}, \mathcal{Y}$ be invariant subspaces of A and B respectively. Let the columns of X and Y be the normalized eigenvectors that span $\mathcal{X}$ and $\mathcal{Y}$ respectively. We have $AX = X\Lambda_X$, $BY = YM_Y$, where Λ_X, M_Y are diagonal matrices containing the corresponding eigenvalues. Let $y \in \mathcal{Y}$ and $x \in \mathcal{X}$ be unit vectors. Suppose $\min_t (\Lambda_X)_{t,t} = \lambda_i$, $\max_t (M_Y)_{t,t} = \mu_j$, and $\min_t (M_Y)_{t,t} = \mu_i$, $\max_t (\Lambda_X)_{t,t} = \lambda_j$. Then, we have*

$$(x^T y)^2 \leq \min\{\kappa_{\max}\frac{\mu_j}{\lambda_i}, \frac{1}{\kappa_{\min}}\frac{\lambda_j}{\mu_i}\}.$$

Proof. Let y be an arbitrary unit vector in $\mathcal{Y}$, with $y = u + v$, where $u \in \mathcal{X}$ and $v \in \mathcal{X}_\perp$, with $\|u\|_2^2 + \|v\|_2^2 = 1$. By using the A-orthogonality of u, v, and positive definiteness, we have

$$y^T Ay = u^T Au + v^T Av \geq u^T Au \geq \|u\|^2 \lambda_i.$$

By definition, we have $y^T By \leq \mu_j$, and by the min-max characterization of the generalized eigenvalues, we have

$$\kappa_{\max} \geq \frac{y^T Ay}{y^T By} \geq \frac{\|u\|^2 \lambda_i}{\mu_j}. \tag{1}$$

Now let x' denote $u/\|u\|_2$. It is easy to see that

$$x' = \arg\max_{x \in \mathcal{X}} x^T y$$

and that $\|u\|_2^2 = (x'^T y)^2$. Combining this with equation 1 proves the first inequality. The second inequality follows from the first by interchanging the roles of A and B and noting that $\lambda_{\max}(B, A) = 1/\lambda_{\min}(A, B)$. □

6 Acknowledgments

This work was partially supported by the National Science Foundation under grant number CCF-0635257.

References

[1] Arik Azran and Zoubin Ghahramani. A new approach to data driven clustering. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 57–64, New York, NY, USA, 2006. ACM Press.

[2] Piotr Berman and Georg Schnitger. On the performance of the minimum degree ordering for gaussian elimination. *SIAM J. Matrix Anal. Appl.*, 11(1):83–88, 1990.

[3] Marcin Bienkowski, Miroslaw Korzeniowski, and Harald Räcke. A practical algorithm for constructing oblivious routing schemes. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms*, pages 24–33, 2003.

[4] Erik G. Boman and Bruce Hendrickson. Support theory for preconditioning. *SIAM J. Matrix Anal. Appl.*, 25(3):694–717, 2003.

[5] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *FOCS*, pages 378–388, 1999.

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

[7] Timothy A. Davis, John R. Gilbert, Stefan I. Larimore, and Esmond G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004.

[8] Petros Drineas, Alan M. Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.

[9] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 494–503, 2005.

[10] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10:345–363, 1973.

[11] John R. Gilbert and Robert E. Tarjan. The analysis of a nested dissection algorithm. *Numerische Mathematik*, 50(4):377–404, 1987.

[12] Keith Gremban. *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems.* PhD thesis, Carnegie Mellon University, Pittsburgh, October 1996. CMU CS Tech Report CMU-CS-96-123.

[13] Chris Harrelson, Kirsten Hildrum, and Satish Rao. A polynomial-time tree decomposition to minimize congestion. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms*, pages 34–43, 2003.

[14] D.S. Hochbaum and D.B. Shmoys. A best possible approximation algorithm for the k-center problem. *Math. Oper Re.*, 10:180–184, 1985.

[15] Anil Joshi. *Topics in Optimization and Sparse Linear Systems.* PhD thesis, University of Illinois at Urbana Champaing, 1997.

[16] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[17] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[18] Ioannis Koutis and Gary L. Miller. A linear work, $O(n^{1/6})$ time, parallel algorithm for solving planar Laplacians. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, 2007.

[19] R.J. Lipton, D. Rose, and R.E. Tarjan. Generalized nested dissection. *SIAM Journal of Numerical Analysis*, 16:346–358, 1979.

[20] Bruce M. Maggs, Gary L. Miller, Ojas Parekh, R. Ravi, and Shan Leung Maverick Woo. Finding effective support-tree preconditioners. In *Proceedings of the 17th Annual ACM Symposium on Parallel Algorithms*, pages 176–185, 2005.

[21] Gary L. Miller and Peter C. Richter. Lower bounds for graph embeddings and combinatorial preconditioners. In *Proceedings of the sixteenth Annual ACM Symposium on Parallel Algorithms*, pages 112–119, 2004.

[22] Rafail Ostrovsky and Yuval Rabani. Polynomial time approximation schemes for geometric k-clustering. In *FOCS*, pages 349–358, 2000.

[23] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *J. Comput. Syst. Sci.*, 61(2):217–235, 2000.

[24] François Pellegrini. A parallelisable multi-level banded diffusion scheme for computing balanced partitions with smooth boundaries. In *Euro-Par 2007, Parallel Processing, 13th International Euro-Par Conference, Rennes, France, August 28-31, 2007, Proceedings*, pages 195–204, 2007.

[25] Harald Räcke. Minimizing congestion in general networks. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 43–52. IEEE, 2002.

[26] Margaret Reid-Miller, Gary L. Miller, and Francesmary Modugno. List ranking and parallel tree contraction. In John Reif, editor, *Synthesis of Parallel Algorithms*, chapter 3, pages 115–194. Morgan Kaufmann, 1993.

[27] Daniel A. Spielman and Shang-Hua Teng. Solving Sparse, Symmetric, Diagonally-Dominant Linear Systems in Time $0(m^{1.31})$. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 416. IEEE Computer Society, 2003.

[28] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 81–90, June 2004.

[29] G.W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory.* Academic Press, Boston, 1990.

[30] Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In *NIPS*, pages 640–646, 2000.