Spectral Embedding of k-Cliques, Graph Partitioning and k-Means

Pranjal Awasthi* Moses Charikar† Ravishankar Krishnaswamy‡ Ali Kemal Sinop§

February 3, 2015

Abstract

We introduce and study a new notion of graph partitioning, intimately connected to k-means clustering. Informally, our graph partitioning objective asks for the optimal spectral simplification of a graph as a disjoint union of k normalized cliques. It is a variant of graph decomposition into expanders (where expansion is not measured w.r.t. the induced graph). Optimizing this new objective is closely related to clustering the effective resistance embedding of the original graph. Our semidefinite programming based approximation algorithm for the new objective is closely related to spectral clustering: it optimizes the k-means objective on a certain smoothed version of the resistive embedding. We also show that spectral clustering applied directly to the original graph also gives guarantees for our new objective function.

In order to illustrate the power of our new notion, we show that approximation algorithms for our new objective can be used in a black box fashion to approximately recover a partition of a graph into k pieces given a guarantee that such a partition exists with sufficiently large gap in internal and external conductance.

*Princeton University. Email: pawasthi@cs.cmu.edu
†Princeton University. Email: moses@cs.princeton.edu
‡Columbia University. Email: ravishan@cs.cmu.edu
§Simons Institute for the Theory of Computing. Email: asinop@cs.cmu.edu

Contents

1 Introduction

Graph partitioning is a fundamental problem in computer science with various applications in computer vision, machine learning and bioinformatics, among others. Such problems have been studied extensively in theoretical computer science (e.g. sparsest cut, multicut, multiway cut, etc) [27, 15, 17, 12]. Designing a good objective function for graph partitioning is a balancing act between several, often conflicting, desiderata. For instance, the objective function formulation should be succinct and mathematically tractable. In traditional graph partitioning objectives, this is captured by formulating the function as a combinatorial measure of the number of edges cut. From a practitioner's point of view, the hope is that the study of the objective function will naturally lead to good algorithms which succeed on practical instances of the problem, if not on all. In this paper, we introduce and study a new spectral notion of graph partitioning, intimately connected to k-means clustering. We make the claim, and formally justify, that our objective function has many of the desirable properties mentioned above.

Formally, we study the following problem (see Section 2 and 3 for definitions of these concepts): Given a graph G with normalized Laplacian L_G, the goal is to find a partition of the vertices $\Gamma = \{C_1, \ldots, C_k\}$, with associated clustering matrix K_Γ whose (i,j)th entry is $1 - \frac{1}{|C_\ell|}$ if $i = j \in C_\ell$, $\frac{-1}{|C_\ell|}$ if both $i, j \in C_\ell$, and 0 otherwise. The objective is to find a partition Γ such that $L_G \succeq \lambda K_\Gamma$ and λ is maximized.

Informally, our graph partitioning objective asks for the optimal spectral simplification of the graph as a disjoint union of normalized cliques. It is a variant of graph decomposition into expanders (where expansion is not measured w.r.t. the induced graph in each piece of the decomposition). Any good decomposition for this objective must respect the structure of small cuts in the graph. Our notion of embedding normalized cliques is a one-sided version of the notion of spectral similarity of graph Laplacians used in the recent influential sequence of work on spectral sparsification [38, 36, 8]. It is akin to the notion of spectrally thin trees that have been studied before [18], except that we look for a decomposition into normalized cliques that is spectrally thin, instead of trees.

To solve our spectral embedding problem we use a semidefinite programming formulation to design an $O(k)$ approximation algorithm for our new objective (Section 4); we also show a matching $\Omega(k)$ integrality gap (Section 4.1). Enroute to this result, we establish several interesting structural properties of our objective formulation and the corresponding SDP relaxation. Our upper bound exploits a connection to k-means: the constant factor hidden by the $O()$ notation is exactly the integrality gap of a natural SDP relaxation for k-means (first studied by [32, 33], and recently investigated by [7] from the perspective of exact recovery). As an aside, we mention that studying this SDP relaxation for k-means seems to be a promising avenue to obtain better worst case approximation algorithms for the k-means objective (we don't know of any integrality gap worse than factor 1.25) but we do not explore this direction in the present work. For our purposes, we directly appeal to a result by Jain and Vazirani [19] on k-median clustering to show that the integrality gap for the k-means formulation is $O(1)$. Further, in Section 5 using the matrix multiplicative weights algorithm of Arora and Kale [3], we show how we can obtain an oblivious rounding of our graph partitioning SDP solution: this gives us a polynomial list of partitions that are good on average. (While we do not need this directly to establish our results, we believe that this is an interesting tool for further study of our graph partitioning objective).

Optimizing our new partitioning objective is equivalent to clustering the effective resistance embedding of the original graph (Section 3) – intimately connected to spectral clustering approaches popular in machine learning. We exploit this characterization in our algorithm (Section 3.2) – we believe this is the first use of the resistive distance embedding in a provable algorithm, although effective resistances (i.e. the values) themselves have been used [36] in spectral sparsification, and the resistive distance embedding has been

used in graph partitioning heuristics [22, 23, 34]. Our SDP based approximation algorithm is very similar to spectral clustering [31, 40, 30] (the choice of k-means as a clustering formulation arises naturally in optimizing our objective) – a new twist in our algorithm is a certain *smoothing* we apply to the resistive distance embedding. We are able to analyze one step of this smoothing operation to get our $O(k)$ guarantee; it is quite possible that an analysis of multiple steps of smoothing will lead to bicriterion guarantees (i.e. improved approximation if we allow more than k pieces in the partition). We further strengthen the connection to spectral clustering by showing in Section 7 that popular variants of this approach applied directly to the original graph also gives guarantees for our new objective function. There is great interest in understanding why the spectral clustering heuristic works. In Section 1.2, we discuss various previous approaches to explain the working of this method. We believe our new graph partitioning objective function and associated algorithm makes a meaningful contribution to this body of work.
Finally, in Section 6, in order to illustrate the power of our new notion of graph decomposition, we show that approximation algorithms for our new objective can be used as a black box to find a partition of a graph into k pieces with sufficiently large gaps between internal and external conductance. Partitions with such properties were studied by [20] (see also [16]); they also arise in the work of Ng, Jordan and Weiss [31]. Given the promise that such a partition exists, any approximation algorithm for our objective directly produces a partition that is guaranteed to have small symmetric difference with this unknown partition.

1.1 Related Work

Graph partitioning objectives have received a lot of attention in theoretical computer science. Most notable among them is the sparsest cut objective [27, 6, 5] which attempts to find a 2-partition with minimum conductance. Conductance is also closely related to the second smallest eigenvalue of the graph Laplacian [11, 1, 2]. Generalizations of this to k-partitions aim at finding k clusters with small conductance [26, 28] as well as high conductance within each cluster [20, 16]. These works exploit gaps in the higher eigenvalues of the graph Laplacian to design algorithms which achieve a good clustering. The most relevant to our work is the result of [16] which uses the spectrum of the graph Laplacian to design a good clustering if one exists (See Section 7 for a result of similar flavor using our new objective). However as opposed to our algorithm, the algorithm proposed in [16] does not fit naturally into the framework of spectral clustering as used in practice. See the next section for details. A sequence of work has also shown that spectral partitioning works well in geometric settings [37, 21, 9].

1.2 Spectral clustering and k-means

Spectral clustering approaches in practice consist of two main steps: a) transforming the original set of points to be clustered using the spectrum of the data matrix (or an associated matrix), and b) optimizing the k-means objective in the new space to obtain a k-partition. The choice of using the k means objective in the new space is not an arbitrary one and several prior works have hinted at a deeper connection. The simplest scenario consists of the original set of points in $\Re^d$ such that there is a true unknown k-partition with the mean vectors of each partition being far away from each other. In this case, Kumar and Kannan [25] show that performing a linear embedding of the points onto the span of the top k singular vectors of the data matrix and optimizing for k means in the new space will lead to good approximations to the true mean vectors. This approach also has applications in the planted partition model of Mcsherry [29].
In many real world applications however, one does not have a true clustering with well separated mean vectors. In such scenarios spectral clustering approaches prove to be immensely successful again. The idea is to now look at non linear embeddings. This approach pioneered by the work of Weiss, Meila and Shi

and Ng, Jordan and Weiss [40, 30, 31] looks at the spectrum of the Laplacian L of the affinity graph. The affinity graph A is an $n \times n$ matrix where the entry $A(i,j) \propto \exp(-\|x_i - x_j\|^2)$. The k dimensional embedding of a point is obtained by using the corresponding coordinates in the top k eigenvectors of L. Ng, Jordan and Weiss [31] justify this approach by showing that if the true clustering is well separated in the sense that it has densely connected clusters with sparse connections across them then in the new space the optimal k means clustering will have a small cost. In many instances, it has been observed that replacing the Laplacian based embedding with the resistive embedding (using the eigen vectors of $L^{\frac{\dagger}{2}}$) works better as it induces a stronger block structure in the induced distance metric [34]. As an application of our result we strengthen this connection by showing in Section 6 how to recover a good point wise approximation to a well separated clustering using our algorithm which is based on resistive embeddings followed by solving k-means. Spectral approaches have also been used with great success in partitioning problems where one is naturally interested in optimizing a cut based objective function [30, 35, 41]. It has also been observed that such approaches have connections, and in some cases are equivalent to optimizing a kernel k-means objective in the original space [13, 14].

2 Preliminaries

2.1 Linear Algebra

Let $\mathbb{R}^{m\times n}$ be the set of all $m \times n$ real matrices; $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices; $\mathbb{S}^m_+ \subset \mathbb{S}^m$ be the set of $m \times m$ positive semidefinite (PSD) matrices. Alternatively, we will use $A \succeq 0$ to denote $A \in \mathbb{S}^m_+$. The $m \times m$ identity matrix is denoted by I_m. For any subset $C \subseteq [n]$, $\mathbf{e}_C \in \mathbb{R}^n$ is the indicator vector for C so that $(\mathbf{e}_C)_i = 1$ iff $i \in C$; 0 else. Define $J_C \stackrel{\text{def}}{=} \mathbf{e}_C \mathbf{e}_C^T$ as the matrix whose entries are 0 except the principal minor on C, which is all-1's.

For any $A \in \mathbb{R}^{m\times n}$, we use A^T and A^{-1} to denote A's transpose and inverse, respectively. We will use $A^\Pi, A^\perp \in \mathbb{S}^m_+$ to refer to the projection matrix onto the column span of A and its orthogonal complement, respectively. Note that $A^\Pi + A^\perp = I_m$. Finally, we will use $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ to denote the maximum and minimum singular values of A.

Given $A \in \mathbb{S}^m$, we define $A^\dagger$ to be the pseudo-inverse of A; and $\lambda_{\min}(A)$, $\lambda_{\max}(A)$ as the the minimum and maximum eigenvalues of A, respectively: $\lambda_{\min}(A) \stackrel{\text{def}}{=} \min_{q\neq 0} \frac{q^T A q}{\|q\|^2}$; $\lambda_{\max}(A) \stackrel{\text{def}}{=} \max_{q\neq 0} \frac{q^T A q}{\|q\|^2}$.

Given $A, B \in \mathbb{S}^m_+$, let $\lambda_{\min}(A,B)$ be the minimum generalized eigenvalue of A and B, $\lambda_{\min}(A,B) \stackrel{\text{def}}{=} \min_{q:Bq\neq 0} \frac{q^T A q}{q^T B q}$. An equivalent definition is $\lambda_{\min}(A,B) \stackrel{\text{def}}{=} \max\{\lambda \big| A \succeq \lambda B\}$.

2.2 Graphs

Given a graph G with adjacency matrix A, let D be the diagonal matrix with $D_{ii} = d_i$ where d_i is the weighted degree of node i. We define the Laplacian matrix of G as $L_G = D - A$. Similarly, let $K \in \mathbb{S}^n_+$ be the Laplacian matrix of a normalized clique, $K \stackrel{\text{def}}{=} I - \frac{1}{n} J$.

2.3 Partitions

We define $\Gamma_k(n)$ to be the set of all proper k-partitions of $[n]$:

$$\Gamma_k(n) \stackrel{\text{def}}{=} \Big\{ \Gamma = \{S_1, \ldots, S_k\} \;\Big|\; S_i \neq \emptyset,\ S_1 \uplus \ldots \uplus S_k = [n] \Big\}.$$

We can construct an associated graph for Γ by placing a normalized clique on each $S \in \Gamma$. We will use K_Γ to denote the Laplacian matrix of corresponding graph.

Definition 2.1. *Given two k-partitions $\Gamma_1, \Gamma_2 \in \Gamma_k(n)$; we say Γ_1 and Γ_2 are ε-close if there is a perfect matching $M \subseteq \Gamma_1 \times \Gamma_2$ such that any matched pair $(S,T) \in M$ has $|S \Delta T| \leq \varepsilon \min(|S|, |T|)$.*

Theorem 2.2. *Given a pair of k-partitions, $\Gamma_1, \Gamma_2 \in \Gamma_k(n)$; consider the matrices $\mathbf{\Gamma_1}, \mathbf{\Gamma_2} \in \mathbb{R}^{n \times k}$ where $\mathbf{\Gamma_1}$ is the orthonormal basis corresponding to Γ_1 ($\mathbf{\Gamma_2}$ is defined similarly):*

$$\mathbf{\Gamma_1} = \left[\frac{1}{\sqrt{|S|}} \mathbf{e}_S \middle| S \in \Gamma_1 \right],$$

Suppose there exists $\varepsilon < 1/2$ with $\sigma_{\min}(\mathbf{\Gamma_1}^T \mathbf{\Gamma_2}) \geq \sqrt{1-\varepsilon}$. Then Γ_1 and Γ_2 are 2ε-close.

Proof. We define $\pi_1 : \Gamma_1 \to \Gamma_2$ and $\pi_2 : \Gamma_2 \to \Gamma_1$ as the following:

$$\forall S \in \Gamma_1 : \pi_1(S) \stackrel{\text{def}}{=} \operatorname*{argmax}_{T \in \Gamma_2} \frac{|S \cap T|}{|T|} \quad \text{and} \quad \forall T \in \Gamma_2 : \pi_2(T) \stackrel{\text{def}}{=} \operatorname*{argmax}_{S \in \Gamma_1} \frac{|S \cap T|}{|S|}.$$

Consider $M = \{(S, \pi_1(S)) \mid S \in \Gamma_1\}$: By Claims 2.4 and 2.5, M is indeed a perfect matching between Γ_1 and Γ_2. Now consider any matched pair $(S,T) \in M$. Without loss of generality, say $|S| \geq |T|$. By Claim 2.3, $|S \cap T| \geq (1-\varepsilon)|S|$. Since $|S \Delta T| = |S| + |T| - 2|S \cap T|$:

$$|S \Delta T| \leq |S| + |T| - 2(1-\varepsilon)|S| = 2\varepsilon|S| + (|T| - |S|) \leq 2\varepsilon|S|.$$

We finish our proof with Claims 2.3 to 2.5.

Claim 2.3. *If $\pi_1(S) = T$, then $|S \cap T| \geq (1-\varepsilon)|T|$. Similarly, if $\pi_2(T) = S$, then $|S \cap T| \geq (1-\varepsilon)|S|$.*

Proof. Consider the matrix $P = \mathbf{\Gamma_1}^T \mathbf{\Gamma_2} \mathbf{\Gamma_2}^T \mathbf{\Gamma_1} \in \mathbb{S}^k_+$ so that $\lambda_{\min}(P) = \sigma^2_{\min}(\mathbf{\Gamma_1}^T \mathbf{\Gamma_2})$. Thus $\lambda_{\min}(P) = \sigma_{\min}(\mathbf{\Gamma_1}^T \mathbf{\Gamma_2})^2 \geq 1 - \varepsilon$. In particular, all diagonals of P are at least $1-\varepsilon$. Consider any diagonal corresponding to $S \in \Gamma_1$:

$$1 - \varepsilon \leq \frac{\mathbf{e}_S^T \mathbf{\Gamma_2} \mathbf{\Gamma_2}^T \mathbf{e}_S}{|S|} = \sum_{T \in \Gamma_2} \frac{|S \cap T|^2}{|S||T|} \leq \left(\max_{T' \in \Gamma_2} \frac{|S \cap T'|}{|T'|} \right) \sum_{T \in \Gamma_2} \frac{|S \cap T|}{|S|} = \max_{T' \in \Gamma_2} \frac{|S \cap T'|}{|T'|},$$

which, by construction, is equal to $\frac{|S \cap \pi_1(S)|}{|\pi_1(S)|}$. This proves the first part of the claim. The second part follows immediately by applying the same argument on Γ_2 and Γ_1. □

Claim 2.4. *Both π_1 and π_2 are bijections.*

Proof. Suppose $\pi_1(S) = \pi_1(S') = T$ for some $S \neq S'$. Since S, S' are disjoint and $\varepsilon < \frac{1}{2}$:

$$|T| \geq |S \cap T| + |S' \cap T| \geq 2(1-\varepsilon)|T| > |T|,$$

a contradiction. A similar argument shows that π_2 is a bijection as well. □

□

Claim 2.5. $\pi_1 = \pi_2^{-1}$.

Proof. Suppose not. Since both Γ_1 and Γ_2 are bijections by Claim 2.4, there exists a cycle of the form $(S_0, T_0, \ldots, S_{m-1}, T_{m-1}, S_m = S_0)$ where $\pi_1(S_i) = T_i$ and $\pi_2(T_i) = S_{i+1}$ for some $m \geq 2$.
By construction, $|S_i \cap T_i| \geq (1-\varepsilon)|T_i|$ which means $\varepsilon|T_i| \geq |T_i \setminus S_i|$. Since S_i and S_{i+1} are disjoint, $|T_i \setminus S_i| \geq |T_i \cap S_{i+1}|$. Again, by construction, $|T_i \cap S_{i+1}| \geq (1-\varepsilon)|S_{i+1}|$. Therefore:

$$\varepsilon|T_i| \geq (1-\varepsilon)|S_{i+1}| \implies |T_i| \geq \frac{1-\varepsilon}{\varepsilon}|S_{i+1}| > |S_{i+1}| \quad (\text{by } \varepsilon < 1/2).$$

By a similar argument, we can also show that $|S_i| > |T_i|$. Consequently, $|S_0| > |S_1| > \ldots > |S_m| = |S_0|$ which is a contradiction. So all cycles have length 2, which implies $\pi_1 = \pi_2^{-1}$. □

The k-means Clustering Problem. Given a partition $\Gamma = \{C_1, C_2, \ldots, C_k\}$, and a set of n points/vectors $\{v_1, v_2, \ldots, v_n\}$ in Euclidean space, with, say their gram matrix being P. Then, the cost of the k-means clustering of the n points according to the clustering Γ is (using the above expansion for each cluster C) precisely $K_\Gamma \bullet P := \mathrm{Tr}(K_\Gamma P) = \sum_{C\in\Gamma} \frac{1}{|C|} \sum_{i,j\in C} \|v_i - v_j\|^2$. This is easy to see because given a cluster C with mean $\mu(C) = \frac{1}{|C|}\sum_{i\in C} v_i$, the sum of squared distances of every point to its mean is equal to the ratio of the total sum of all pairwise distances between points within the cluster and the size of the cluster.
Finally, given a non-negative symmetric matrix X (such as the edge weights of a graph), let L_X be the corresponding Laplacian matrix. For any doubly stochastic matrix X, $L_X = I - X$. And, given a subset S, we use $\mathbf{e}_S$ to denote the indicator vector for S so that $[\mathbf{e}_S]_i = 1$ iff $i \in S$; we also use $\mathbf{e}$ for the all-1's vector.

3 Spectral Embedding of k-Cliques

We now formally define the spectral embedding we study in this work. Given as input a connected graph G with the corresponding Laplacian matrix L_G, the goal is to find a partition Γ into k cliques while optimizing the following objective:

$$\max \lambda \text{ st } L_G \succeq \lambda K_\Gamma.$$

In order to understand this objective better consider the following simple scenario first. Given a Laplacian matrix L and k-partition, for some λ, how can we even verify $L \succeq \lambda K_\Gamma$? From an algorithmic perspective, this is easy: Compute the minimum generalized eigenvalue of L and K_Γ and compare it against λ. From an analysis perspective on the other hand, this yields little to no insight for us on how to certify that the graph admits a good spectral k-clique-embedding. There is no analogue of Cheeger's inequality for the generalized case, and indeed there is strong evidence to believe none exists [39]. As a result, we first present a characterization of k-clique embeddability and, using this, derive a sufficient condition for the existence of one. This condition then leads to a natural algorithm for finding such clustering.

3.1 Necessary and Sufficient Conditions

In general, proving a lower bound is harder than proving an upper bound. So we start by relating the minimum eigenvalue to a standard maximum eigenvalue problem. While there are other bounds for this [10], we prove one most amenable to us.

Lemma 3.1. *Given graph G and a k-partition Γ of V, such that any $C \in \Gamma$ induces a connected component in G: $\max\{\lambda \mid L_G \succeq \lambda K_\Gamma\} = \frac{1}{\lambda_{\max}(K_\Gamma L_G^\dagger K_\Gamma)}$. Furthermore the maximum eigenvector p of $K_\Gamma L_G^\dagger K_\Gamma$ satisfies $K_\Gamma p = p$.*

For simplicity in notation, we define $L \stackrel{\text{def}}{=} L_G$, $M \stackrel{\text{def}}{=} K_\Gamma$ and $K \stackrel{\text{def}}{=} L \cdot L^\dagger$. Additionally, let $\lambda_{\min}(L, M) = \lambda_{\min}$ and $\lambda_{\max}(ML^\dagger M) = \lambda_{\max}$. Since any cluster $C \in \Gamma$ is connected in G, $\lambda_{\min} > 0$.
The first identity (i) follows immediately from the definition of $\lambda_{\min}$: $\lambda_{\min}(L, M) = \min_{q:Mq\neq 0} \frac{q^T Lq}{q^T Mq}$, So for all q, $q^T Lq \geq \lambda_{\min} q^T Mq$ implying $L \succeq \lambda_{\min} M$. We prove (ii) in two parts.

Proof of $\lambda_{\max} \geq 1/\lambda_{\min}$. Given any g with $Lg = \lambda_{\min} Mg$, we can assume $Kg = g$, since Kg also satisfies this identity. Therefore:

$$g = Kg = L^\dagger Lg = \lambda_{\min} L^\dagger Mg.$$

Multiplying with $\frac{1}{\lambda_{\min}} M$ on both sides,

$$\frac{1}{\lambda_{\min}} Mg = ML^\dagger Mg.$$

For $p \stackrel{\text{def}}{=} Mg$, together with the fact that $M^2 = M$, this becomes

$$\frac{1}{\lambda_{\min}} p = ML^\dagger Mp.$$

Therefore $\lambda_{\max}(ML^\dagger M) \geq \frac{1}{\lambda_{\min}}$. □

Proof of $\lambda_{\max} \leq 1/\lambda_{\min}$. We always have $\lambda_{\min} \geq 0$ so we only need to consider the case of $\lambda_{\max} > 0$. Given corresponding eigenvector p with $M\left(L^\dagger Mp\right) = \lambda_{\max} p$, it is easy to see that $Mp = p$. Therefore

$$ML^\dagger p = \lambda_{\max} p.$$

Eigenvalues of a matrix and its transpose are the same, thus

$$(ML^\dagger)^T = L^\dagger M$$

has maximum eigenvalue $\lambda_{\max}$ with eigenvector g, so that $L^\dagger Mg = \lambda_{\max} g$. Multiplying both sides by L and observing that $LL^\dagger = K$ with $KM = M$,

$$\lambda_{\max} \cdot Lg = LL^\dagger Mg = KMg = Mg.$$

Hence

$$\lambda_{\min}(L, M) \leq \frac{g^T Lg}{g^T Mg} = \frac{1}{\lambda_{\max}}.$$ □

Equipped with Lemma 3.1, our goal is now much simpler: Instead of lower bounding the minimum generalized eigenvalue of a pair of matrices, we want to upper bound the maximum eigenvalue of a single matrix.

Theorem 3.2. *Given a graph G with normalized adjacency (normalized Laplacian) matrix A (L resp.), suppose Γ is a k-partition such that any $C \in \Gamma$ induces a connected component in G. Then:*

$$\lambda_{\min}(L, K_\Gamma) \geq \max_{\tau \in \mathbb{Z}_+} \left\{ \max(\lambda^{-1}, 2\tau) + \operatorname{Tr}\left[K_\Gamma A^\tau (L^\dagger)_k A^\tau \right] \right\}^{-1}.$$

Here λ is the k^{th} smallest eigenvalue of L, and $(L^\dagger)_k$ is the matrix $L^\dagger$ projected onto its top k-eigenvectors. Equivalently, for y_i's being the columns of $[L^\dagger]_k^{1/2} A^\tau$, the trace is equal to the cost of clustering these vectors using Γ.
In particular, for the choice of $\tau \leftarrow 0$:

$$\lambda_{\min}(L, K_\Gamma) \geq \left\{ \frac{1}{\lambda_k} + \mathrm{Tr}\left[K_\Gamma (L^\dagger)_k\right] \right\}^{-1}.$$

Proof. Let λ_i and q_i be the i^{th} smallest eigenvalue and corresponding eigenvector of L. Suppose G has c-connected components. Then $0 = \lambda_1 = \ldots = \lambda_c < \lambda_{c+1} \leq \ldots \leq \lambda_n \leq 2$. Moreover if we define $K \stackrel{\text{def}}{=} L \cdot L^\dagger$ (the projection matrix onto the complement of connected components of G), then $K = \sum_{j>c} q_j q_j^T$. By our assumption, $K_\Gamma K = K K_\Gamma = K_\Gamma$.
For any $x \in [0, 2]$, $\frac{1}{x} = \frac{1}{1-(1-x)} \leq 2\tau + \frac{(1-x)^{2\tau}}{x}$. Using this, we can now upper bound $L^\dagger$:

$$L^\dagger = \sum_{j>c} \frac{1}{\lambda_j} q_j q_j^T \preceq \sum_{c<j<k} \left(2\tau + \frac{(1-\lambda_i)^{2\tau}}{\lambda_i}\right) q_j q_j^T + \frac{1}{\lambda_r} \sum_{j \geq r} q_j q_j^T \preceq \max(2\tau, \lambda_r^{-1}) K + A^\tau (L)_k A^\tau.$$
$$K_\Gamma L^\dagger K_\Gamma \preceq \max(2\tau, \lambda_r^{-1}) K_\Gamma + K_\Gamma A^\tau (L)_k A^\tau K_\Gamma.$$

Since $\lambda_{\max}(X + Y) \leq \lambda_{\max}(X) + \lambda_{\max}(Y)$,

$$\begin{aligned}\lambda_{\max}(K_\Gamma L^\dagger K_\Gamma) &\leq \max(2\tau, \lambda_r^{-1}) \underbrace{\lambda_{\max}(K_\Gamma)}_{=1} + \lambda_{\max}(K_\Gamma \underbrace{A^\tau (L)_k A^\tau}_{\succeq 0} K_\Gamma) \\ &\leq \max(2\tau, \lambda_r^{-1}) + \mathrm{Tr}(K_\Gamma A^\tau (L)_k A^\tau K_\Gamma).\end{aligned}$$

The proof is complete by using the identity from Lemma 3.1. □

3.2 Main Algorithm

The lower bound presented in Theorem 3.2 corresponds to a natural algorithm for finding Γ. We state the algorithm only for connected graphs. The disconnected case can easily be handled by recursing on each component separately. Even though this algorithm has polynomial running time, it can be implemented much more efficiently. However for our purposes, this is sufficient.
Input. Number of clusters k, normalized adjacency and Laplacian matrices $A, L = I - A$ respectively.
Output. λ and k-partition Γ such that $L \succeq \lambda K_\Gamma$.

1. *(Resistive Embedding)* For every $i \in V$, let $Y_i^{(0)} \leftarrow (L^{\dagger/2})_k \mathbf{e}_i$ be the resistive embedding of node i projected onto top k-eigenvalues.
2. *(Best k-Partition So Far)* Let Γ_{best} be an arbitrary k-partition. $\lambda_{\text{best}} \leftarrow 0$.
3. For $\tau \leftarrow 0$ to $\frac{1}{\lambda_2}$ do:
   (a) Find a k-means solution Γ for $Y^{(t)} = \left[Y_i^{(t)} \;\middle|\; i \in V \right]$.

$$\lambda_{cur} \leftarrow \left[\max(2\tau, \lambda_k^{-1}) + \|Y^{(t)} K_\Gamma\|_F^2 \right]^{-1}.$$

(b) If $\lambda_{cur} > \lambda_{\text{best}}$, then $\Gamma_{\text{best}} \leftarrow \Gamma$ and $\lambda_{\text{best}} \leftarrow \lambda_{cur}$.

(c) *(Smoothing)* For every $i \in V$, $Y_i^{(t+1)} \leftarrow \sum_j A_{ij} Y_j^{(t)}$.

Theorem 3.3. *(Correctness) The output of this algorithm, Γ_{best} and λ_{best}, satisfies $L \succeq \lambda_{\text{best}} K_{\Gamma_{\text{best}}}$. (Approximation Factor) Furthermore $\lambda_{\text{best}} \geq \frac{\lambda_{\mathsf{OPT}}}{O(k)}$, where λ_{OPT} is the optimum.*

Proof. Correctness follows immediately from Theorem 3.2.
Let Γ_{opt} be an optimal solution with $L \succeq \lambda_{\mathsf{OPT}} K_{\Gamma_{\text{opt}}}$. Consider the projection matrix onto top k-eigenvectors of $L^\dagger$, Q_k: $Q_k = \sum_{j<k} q_j q_j^T$. Note that Q_k commutes with L. Thus, if we multiply the first expression on both sides with $Q_k L^{\dagger/2} Q_k = (Y^{(0)})^T (Y^{(0)})$:

$$Q_k \succeq \lambda_{\mathsf{OPT}} \cdot (Q_k L^{\dagger/2} Q_k) K_{\Gamma_{\text{opt}}} (Q_k L^{\dagger/2} Q_k).$$

Taking the trace, $k \geq \lambda_{\mathsf{OPT}} \|Y^{(0)} K_{\Gamma_{\text{opt}}}\|_F^2$. Therefore at time $\tau = 0$, there exists a k-means solution of cost $\leq \frac{k}{\lambda_{\mathsf{OPT}}}$. Hence the algorithm will find some Γ_{best} with

$$\|Y^{(0)} K_{\Gamma_{\text{best}}}\|_F^2 \leq \frac{O(k)}{\lambda_{\mathsf{OPT}}}.$$

Consequently $\lambda_{best} \leq \frac{1}{\lambda_k} + \frac{O(k)}{\lambda_{\mathsf{OPT}}} \leq \frac{O(k)}{\lambda_{\mathsf{OPT}}}$ where we used the fact that $\lambda_{\mathsf{OPT}} \leq \lambda_k$. □

Remark: Connection with Spectral Clustering. In machine learning, data mining and similar fields, a very common approach for clustering is to apply k-means on:

(I) Either the resistive embedding,

(II) Or the embedding obtained by the smallest k-eigenvectors.

In these cases, our algorithm above, or more precisely its analysis from Theorem 3.2, could be seen to offer an explanation for *what kind of k-partitions* such methods implicitly seek.
For example, if an algorithm of type (I) finds a k-partition with small cost in the resistive embedding; it means the underlying clusters behave like an "expander". The same conclusion holds if an algorithm of type (II) finds a k-partition and provided that the underlying graph has $\lambda_2 \approx \lambda_k$. Hence an intriguing practical problem is whether the applying the smoothing step helps clustering in such domains. We give a more rigorous connection in Section 7.

4 $O(k)$-Approximation Algorithm and Matching Integrality Gap

In this section, we present our main algorithmic result regarding spectrally embedding k-partitions with an approximation factor of $O(k)$. Our algorithm is based on rounding an SDP relaxation. Next we will prove that our analysis is tight by presenting an $\Omega(k)$ integrality gap.

Theorem 4.1. *There is an efficient $O(k)$ approximation algorithm for the spectral embedding problem.*

Our algorithm is based on rounding the following semidefinite programming relaxation using the procedure described in Section 3.2. Indeed, we relax the requirement of finding a disjoint union of cliques on each cluster to the following:

$$\max\ \lambda \text{ st } L_G \succeq \lambda(I - X);\ \text{Tr}(X) = k;\ X\mathbf{e} = \mathbf{e};\ X \text{ is diagonally dominant.}$$

One immediate observation which will be subsequently useful is that X can be viewed as a feasible fractional solution to the k-means problem on n (unknown) points! Our rounding algorithm proceeds by in fact finding a k-partition Γ such that $L(X) = I - X \succeq \Omega(\frac{1}{k})K_\Gamma$, where K_Γ is the normalized Laplacian of the disjoint union of cliques for each cluster in Γ. But this implies that $L_G \succeq \Omega(\frac{1}{k\lambda})K_\Gamma$, giving us our desired $O(k)$ approximation. Our main result of this section is the following lemma.
For the rest of the section, we assume that the graph corresponding to the adjacency matrix X is connected. If not, we can recurse in each of the connected components and get our final result.

Lemma 4.2. *Given any symmetric matrix X which is doubly stochastic (i.e., every row sum is 1 and every column sum is 1), diagonally dominant (i.e., $X_{ij} \leq \min(X_{ii}, X_{jj})$), and has trace k, i.e., $\sum_i X_{ii} = k$; the algorithm in Section 3.2 on input X will always find a k-partition Γ with $L_X \succeq \frac{1}{O(k)}K_\Gamma$.*

Proof. Consider $\tau = 1$. As we noted in Section 3.2, we can view $\mathrm{Tr}\left(K_\Gamma X L^\dagger X\right)$ as the k-means cost of clustering the data points represented by the Gram matrix $XL^\dagger X$ according to the clustering Γ. Now, in order to find a good clustering Γ that minimizes this trace, let us plug in X, after noticing that X itself is *a feasible fractional solution for the k-means problem*. The fractional cost is then $\mathrm{Tr}(L \cdot XL^\dagger X) = \mathrm{Tr}(X^2) \leq \mathrm{Tr}(X) = k$. Therefore, by appealing to the Jain-Vazirani[19] approximation algorithm for k-means clustering (which is also an LP-based rounding algorithm and hence establishes $O(1)$-upper bound on the integrality gap of the natural LP), we can get a partition Γ with matching cost. □

4.1 $\Omega(k)$ Integrality Gap

Now we will prove that our analysis for this rounding is essentially tight by constructing a matching $\Omega(k)$ integrality gap instance.

Theorem 4.3. *Given k, consider $X \stackrel{\text{def}}{=} \frac{1}{2(k+1)}C_{k+1} + \left(1 - \frac{1}{k+1}\right)I_{k+1}$, where C_{k+1} is the adjacency matrix of an unweighted cycle graph on $k+1$ nodes. X satisfies the following. (a) X is a feasible fractional k-means solution. (b) For any k-partition Γ, $\lambda_{\min}(L_X, K_\Gamma) < O(1/k)$. In particular, X is an $\Omega(k)$-integrality gap for our spectral embedding relaxation.*

Proof. By construction, X is doubly stochastic and diagonally dominant. X can also be viewed as the uniform distribution over $k+1$ different k-partitions of $\{0, \ldots, k\}$ where i^{th} partition, $i = 0 \ldots k$, places i and $i \pmod{k+1}$ in the same cluster and everything else in its own cluster. This establishes that $X \succeq 0$. Hence X is indeed feasible.
Observe that $L_X = \frac{1}{k+1}I - \frac{1}{2(k+1)}C_{k+1}$, which is the Laplacian matrix of a normalized $k+1$ cycle scaled by $\frac{1}{k+1}$. In particular, $\lambda_{\max}(L_X) \leq \frac{2}{k+1}$. On the other hand, any k-partition Γ has $K_\Gamma \neq 0$, which means there exists some q with $q^T K_\Gamma q = 1$. For such q, $q^T L_X q < \frac{2}{k+1}$. Therefore $\lambda_{\min}(L_X, K_\Gamma) \leq \frac{2}{k+1}$. □

4.2 Smoothing for Multiple Steps

We end this section with a small remark regarding the use of single versus multiple steps of smoothing and possibility of obtaining a bi-criteria approximation.
While we consider only a single step of smoothing for our rounding algorithm to get a factor $O(k)$ bound, in general, there are instances where multiple steps of smoothing decreases the approximation factor exponentially.

For example, consider the case of N-dimensional hypercube with $n = 2^N$. Given k of the form $k = \binom{N}{\leq w}$ for some w, the eigenvectors of $A^\tau (L^\dagger)_k A^\tau$ correspond to the size-$\leq w$ subsets of $[N]$. In particular, for each $S \subseteq [N] : |S| \leq w$, there is a distinct eigenvector q with eigenvalue

$$\lambda_S = \frac{N}{2|S|}\Big(1 - \frac{2|S|}{N}\Big)^{2\tau} \leq \frac{N}{2|S|} \exp\Big(-\frac{4\tau|S|}{N}\Big).$$

correspond to size $\leq w$ subsets of $[N]$,
the non-zero eigenvectors of
the (lazy) d- hypercube. That is, $X = \frac{1}{2}I + \frac{1}{2d}H$, where H is the adjacency matrix of the d-dimensional hypercube over $2k = 2^d$ nodes. Essentially, G is a convex combination over all the d dictator- clusterings, where each partition has k clusters of size 2. After τ steps of smoothing, there exists a k-partition Γ with $\big[\lambda_{\min}(L_X, K_\Gamma)\big]^{-1} \leq 2\tau + O(1)\operatorname{Tr}(L_X \cdot X^\tau L_X^\dagger X^\tau) = 2\tau + O(1)\operatorname{Tr}(X^{2\tau})$. For every binary d-vector $y \in \{0,1\}^d$ with weight $w = \sum_i y_i$, X has an eigenvalue $\lambda_y = 1 - \frac{w}{d}$. Hence $\operatorname{Tr}(X^{2\tau}) = 2k\mathbb{E}_{y\in\{0,1\}^d}\Big[\Big(1 - \frac{w}{d}\Big)^{2\tau}\Big] \leq \Big(1 + \exp(-2\tau/d)\Big)^d \leq \exp\Big[d\exp(-2\tau/d)\Big]$. For $\tau \leftarrow \frac{1}{2}d\ln d = O(\ln k \ln\ln k)$, $\operatorname{Tr}(X^{2\tau}) \leq O(1)$. Hence our upper bound goes down from $O(k)$ to $O(\ln k \ln\ln k)$. So this is an example where the algorithm analysis is provably better after multiple rounds of smoothing, as opposed to just one.

Remark 4.4 (Bi-criteria)**.** *Even though the integrality gap given in Theorem 4.3 precludes the possibility of achieving $o(k)$-factor approximation, it remains an intriguing possibility whether we can get $\operatorname{poly}\log(k)$-factor with a $O(k)$-partition using the full power of smoothing for multiple steps.*

5 Spectrally Embedding a Distribution of Partitionings

In this section, we prove the following theorem, which shows that we can get much better approximation guarantees if we allow ourselves to find a distribution over k-partitions.

Theorem 5.1. *Given a feasible fractional solution X to the k-partitioning problem, we can efficiently find a distribution $\mathcal{X}$ over k-partitions such that $L_X \succeq \alpha E_{\Gamma\sim\mathcal{X}}[K_\Gamma]$ where $\alpha > \frac{1}{216}$ is a constant. For the curious reader, $1/\alpha$ is the approximation ratio (more specifically, integrality gap) of the best known rounding algorithm for the Euclidean k-means problem using the natural LP. If running time is not a concern, then $1/\alpha$ is the true integrality gap of the natural k-means LP.*

A nice additional property this says is that if X is a fractional solution for the *Euclidean k-means problem*, then we can round X **obliviously** into an integral clustering *even without looking at the point-set, or their distances*, and still achieve the best possible approximation factor.
We prove this by writing down the SDP for the best convex combination of k-partitions into X, and analyzing its dual. Consider the following SDP, which, given a feasible fractional k-means solution, tries to find the best convex combination of K_Γ's which embed into X, and its dual. Here, let $\mathbf{\Gamma}$ denote the set of all possible k-partitions of a given graph G.

$$\max \sum_\Gamma w_\Gamma \text{ st } \sum_\Gamma w_\Gamma K_\Gamma \preceq L_X.$$
$$\min L_X \cdot Y \text{ st } K_\Gamma \cdot Y \geq 1, \quad \forall \Gamma \in \mathbf{\Gamma},\ Y \succeq 0,$$

Clearly, notice that if we show that the primal solution has objective value at least α, then we're done (by scaling by $1/\sum_\Gamma w_\Gamma$, we'll get a convex combination which embeds into $(1/\alpha)L_X$). In what follows, we'll first show that, indeed, this is true, and subsequently show how to approximately solve this SDP using the Multiplicative Weights framework of Arora and Kale [4].

To show that the objective value is at least α, consider the dual. We show that the dual optimal has value at least α. Indeed, what is the dual trying to solve? Upon careful inspection, it is trying to find, given X, the *worst-case* set of points in Euclidean space, for which the k-means LP has the largest integrality gap. Indeed, suppose the set of points $\{y_1, y_2, \ldots, y_n\}$ are such that their gram matrix is Y, then $K_\Gamma \cdot Y$ is precisely the k-means cost of the data set according to clustering Γ. So the dual asks for all the true clusterings to have cost at least 1 while minimizing the fractional cost of the k-means LP. But this is precisely the integrality gap instance! From existing rounding algorithms (most relevant to our work is that of Jain and Vazirani [19]), we know that the integrality gap is bounded by a small constant $c_{JV} \leq 216$, and hence the dual objective is at least a constant $1/c_{JV} \geq \alpha$. We hence know that the dual SDP has optimal value at least $\alpha = \frac{1}{216}$. Hence, the primal objective has value at least α, which completes the existential result.

It remains to show that we can efficiently construct the distribution $\mathcal{X}$. Indeed, we show that we can achieve this using the Matrix Multiplicative Weights framework of Arora and Kale [4]. Perhaps not surprisingly, the "oracle" needed in their algorithm amounts to running the Jain-Vazirani approximation algorithm for k-means. We prove this in Appendix B.

6 An Application: Identifying Clusters in Well Separated Graphs

In this section, we demonstrate the power of our objective by presenting an application in a traditional k-way clustering problem. In order to simplify the exposition, we assume G is regular as usual, with normalized adjacency and Laplacian matrices given by A and $L = I - A$, respectively. However our results carry over to the non-regular case (using the appropriate notion of expansion and sparsity) as well. Indeed, suppose a graph G has a good *k-partition into expanders*, such that each cluster has low external sparsity and every cluster is internally an expander. Then we show that, as long as the internal expansion is sufficiently more than the external sparsity[1], then we can recover a clustering which is ϵ-close, in terms of symmetric difference, by simply computing our k-partition. Before we delve into the details, let us introduce some notation.

Definition 6.1. *Given a graph G and a k-partition Γ, we say that Γ is a (k, ϕ, λ) partition for G provided the following: (i) Every $S \in \Gamma$ has small sparsity in G, i.e., $\phi_G(S) \leq \phi$. (ii) Every $S \in \Gamma$ induces an algebraic expander, i.e., $\lambda_2(L[S]) \geq \lambda$.*

A similar notion appeared in the work of Gharan and Trevisan [16]. There they proved the existence of such clustering when there is a gap between λ_k and λ_{k+1}. They also provided an algorithmic version assuming when there is a larger gap. Unfortunately a direct comparison of both algorithms is not possible, as the goals are different: Their algorithm is designed to work assuming only a gap in the spectrum, thus requires a much larger gap (between ϕ and λ) than ours; whereas our algorithm works assuming the existence of such clustering.

Our main result of this section is Theorem 6.2, which says that if a graph admits some (k, ϕ, λ) partition, then our algorithm will output a k-partition which is ε-close to it.

[1] To the best of our knowledge, all known constructive algorithms need some such assumption which may appear in different manifestations, like an eigenvalue gap, for instance.

Theorem 6.2. *There exists a constant $1 > \alpha > 0$ such that the following holds. Given graph G with (k, ϕ, λ)-partition $\Gamma_{\mathsf{OPT}} = \{T_1, \ldots, T_k\}$ and $\varepsilon \stackrel{\text{def}}{=} \frac{k\phi}{\lambda} \leq \alpha$, our algorithm will output a k-partition Γ of the form $\Gamma = \{S_1, \ldots, S_k\}$ such that: $\forall i \in [k] : |S_i \Delta T_i| \leq O(\varepsilon) \min(|S_i|, |T_i|)$.*

We devote the remainder of this section to the proof of Theorem 6.2. We use $\Gamma_{\mathsf{OPT}} = \{T_1, \ldots, T_k\}$ to denote a (k, ϕ, λ)-partition of G with $\frac{k\phi}{\lambda} = \varepsilon$ for some $\varepsilon \leq O(1)$. We will use $\Gamma = \{S_1, \ldots, S_k\}$ to refer to the partition found by our algorithm. Unless noted otherwise, we use S (solution we found) to refer to the clusters in Γ and T (ground truth) to refer to the clusters in Γ_{OPT}.
For further convenience, we define $A_{\Gamma_{\mathsf{OPT}}}$ and A_Γ as the normalized adjacency matrices for the union of cliques on Γ_{OPT} and Γ, respectively:

$$A_{\Gamma_{\mathsf{OPT}}} \stackrel{\text{def}}{=} \sum_T \frac{1}{|T|} J_T \text{ and } A_\Gamma \stackrel{\text{def}}{=} \sum_S \frac{1}{|S|} J_S.$$

Note that $K_{\Gamma_{\mathsf{OPT}}} = I - A_{\Gamma_{\mathsf{OPT}}}$ and $K_\Gamma = I - A_\Gamma$.
In Proposition 6.3, we will show that L spectrally dominates the union of cliques on Γ_{OPT} and Γ.

Proposition 6.3. *If each $T \in \Gamma_{\mathsf{OPT}}$ induces a λ-algebraic expander, then $L \succeq \lambda K_{\Gamma_{\mathsf{OPT}}}$ and $L \succeq \frac{\beta\lambda}{k} K_\Gamma$ for some universal constant $\beta > 0$.*

Proof. For any pair of subsets A, B, define $L[A, B]$ as the Laplacian matrix induced between A and B including only the edges between A and B. Observe $L[A, B] \succeq 0$. By abusing notation, we can express L as $L = \sum_{i \leq j} L[T_i, T_j] \succeq \sum_i L[T_i]$. Since $L[T]$ is an algebraic expander for every $T \in \Gamma_{\mathsf{OPT}}$, $L[T] \succeq \lambda K_T$. Therefore $L \succeq \lambda \sum_T K_T = \lambda K_{\Gamma_{\mathsf{OPT}}}$. Second part follows from Theorem 3.3. □

We will prove that $A_\Gamma \approx A_{\Gamma_{\mathsf{OPT}}}$ so as to relate Γ and Γ_{OPT}. First, we need to upper bound the spectral radius of the Laplacian obtained by contracting each cluster in Γ_{OPT}.

Lemma 6.4. *Let $\phi \stackrel{\text{def}}{=} \max_{T \in \Gamma_{\mathsf{OPT}}} \phi_G(T)$. Then $\phi \leq \lambda_{\max}(A_{\Gamma_{\mathsf{OPT}}} L A_{\Gamma_{\mathsf{OPT}}}) \leq 2\phi$.*

Proof. Define $D \in \mathbb{S}^k_+$ as the diagonal matrix with entries $(|T| \mid T \in \Gamma_{\mathsf{OPT}})$ and $P \in \{0, 1\}^{n \times k}$ as the matrix whose columns are indicator vectors for each $T \in \Gamma_{\mathsf{OPT}}$. Note that $U \stackrel{\text{def}}{=} PD^{-1/2} \in \mathbb{R}^{n \times k}$ is an orthonormal basis, $U^T U = I_k$. Moreover $UU^T = \Pi$. The entry of matrix $\widehat{L} \stackrel{\text{def}}{=} P^T L P$ at (S, T) for $S, T \in \Gamma_{\mathsf{OPT}}$ is

$$\widehat{L}_{S,T} = \begin{cases} \text{total weight of edges crossing } S, C(S, \overline{S}) & \text{if } S = T, \\ -\text{total weight of edges between } S \text{ and } T & \text{else.} \end{cases}$$

Hence $\widehat{L}$ is a Laplacian matrix and $\widehat{L} \preceq 2\,\mathrm{diag}(C(T, \overline{T}))_{T \in \Gamma_{\mathsf{OPT}}}$. If we multiply with $D^{-1/2}$ on both sides, on RHS we obtain a diagonal matrix with entries $\frac{2C(T,\overline{T})}{|T|} \leq 2\phi_G(T) \leq 2\phi$:

$$U^T L U = D^{-1/2} \widehat{L} D^{-1/2} \preceq 2\phi I_k.$$

Again, multiplying with U, U^T on the left and right, respectively:

$$2\phi\Pi = 2\phi UU^T \succeq UU^T LUU^T = A_{\Gamma_{\mathsf{OPT}}} L A_{\Gamma_{\mathsf{OPT}}}.$$

For the lower bound, consider $q \leftarrow K\mathbf{e}_T$ where $T = \mathrm{argmax}_{T \in \Gamma_{\mathsf{OPT}}} \phi_G(T)$:

$$\Pi q = q \implies q^T A_{\Gamma_{\mathsf{OPT}}} L A_{\Gamma_{\mathsf{OPT}}} q = q^T L q = \phi_G(T) \frac{|T| \cdot |\overline{T}|}{n} = \phi_G(T) \|q\|^2. \qquad \square$$

We are ready to relate $A_{\Gamma_{\mathsf{OPT}}}$ and A_Γ to each other via Lemma 6.4. Recall $L \succeq \frac{\lambda}{O(k)} K_\Gamma$. Multiply with $A_{\Gamma_{\mathsf{OPT}}}$:

$$\frac{\lambda}{O(k)} A_{\Gamma_{\mathsf{OPT}}} K_\Gamma A_{\Gamma_{\mathsf{OPT}}} \preceq A_{\Gamma_{\mathsf{OPT}}} L A_{\Gamma_{\mathsf{OPT}}} \preceq 2\phi A_{\Gamma_{\mathsf{OPT}}}$$

where we used Lemma 6.4 in the last step. In particular, for $\mathbf{\Gamma}$ and $\mathbf{\Gamma}_{\mathsf{OPT}}$ being the orthonormal matrices corresponding to Γ and Γ_{OPT} as described in Theorem 2.2, then we see that $I_k - \mathbf{\Gamma}^T \mathbf{\Gamma} = K_\Gamma$, $\Gamma_{\mathsf{OPT}}^T \Gamma_{\mathsf{OPT}} = A_{\Gamma_{\mathsf{OPT}}}$:

$$A_{\Gamma_{\mathsf{OPT}}} K_\Gamma A_{\Gamma_{\mathsf{OPT}}} \preceq O(\varepsilon) A_{\Gamma_{\mathsf{OPT}}} \implies (1 - O(\varepsilon)) I_k \preceq \mathbf{\Gamma}_{\mathsf{OPT}}{}^T \mathbf{\Gamma} \mathbf{\Gamma}^T \mathbf{\Gamma}_{\mathsf{OPT}}.$$

Theorem 2.2 immediately implies that Γ and Γ_{OPT} are $O(\varepsilon)$ close.

7 Connection with Spectral Clustering

Finally, we present yet another connection with a traditional approach often deployed in practice — spectral clustering. In this approach, given a graph, the algorithm for graph partitioning is to essentially compute the top k eigenvectors of the graph Laplacian, project every vertex to the top k eigenvectors, and simply run a k-means clustering algorithm on these vectors. We now show that, implicitly, there is a connection to our objective function. Indeed, we show that if the spectral clustering has low cost for some k, then the same clustering is a good solution for our PSD-embedding of k-cliques problem as well!

Given graph G, let (λ_i, q_i) be the pair of i^{th} smallest eigenvalue and corresponding eigenvector. Consider $Q_i \stackrel{\text{def}}{=} \sum_{2 \le j \le i} q_i q_i^T$. Observe that $Q_n = K$. Recall the basic spectral clustering heuristic for k-clusters: Output the partition found by running k-means on Q_k. In the next theorem, we will show that if this heuristic finds a small cost solution, then the solution is spectrally embeddable into G.

Theorem 7.1. *Suppose for some k, there exists a k-partition Γ such that: $\mathrm{Tr}\left(K_\Gamma \cdot Q_k\right) \le \kappa \frac{\lambda_2}{\lambda_k - \lambda_2}$. Then $L \succeq \frac{\lambda_k}{\kappa+1} K_\Gamma$. Moreover this is within factor $\kappa + 1$ of the best possible.*

The proof is in Appendix C.

8 Conclusions

In this paper we propose a new notion of graph partitioning which involves spectrally embedding a disjoint union of k (normalized) cliques into the original graph. We motivate and justify the study of our notion of spectral embedding by exhibiting several interesting connections to k-means, spectral clustering and the use of resistive embeddings which is a common heuristic in practical applications. We give the first theoretical justification for the utility of resistive embeddings in clustering and partitioning problems. As an application of our framework we also show how to recover good partitions on graphs where one exists.

When studying graph partitioning problems, the modeling aspect of choosing a good objective function is often overlooked in favor of the study of more traditional objectives. Our work illustrates that a well chosen objective function can lead to insights into the operation of heuristic algorithms as well as bring up intriguing algorithmic questions. One such question concerns the use of a novel smoothing step that we use in our approximation algorithm. Although we only use one step of smoothing to get our guarantees, we believe that additional steps of smoothing should help in designing constant factor bi-criteria approximations for the problem. It would also be interesting to see if smoothing helps in practice in conjunction with current spectral clustering heuristics.

References

[1] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. 2

[2] N. Alon and V. D. Milman. Lambda1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985. 2

[3] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 227–236. ACM, 2007. 1

[4] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 227–236. ACM, 2007. 11, 18, 19

[5] S. Arora, J. Lee, and A. Naor. Euclidean distortion and the sparsest cut. *Journal of the American Mathematical Society*, 21(1):1–21, 2008. 2

[6] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009. 2

[7] P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward. Relax, no need to round: integrality of clustering formulations. *arXiv preprint arXiv:1408.4045*, 2014. 1

[8] J. Batson, D. A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. 1

[9] P. Biswal, J. R. Lee, and S. Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM (JACM)*, 57(3):13, 2010. 2

[10] E. G. Boman and B. Hendrickson. Support theory for preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 25(3):694–717, 2003. 5

[11] J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*, 625:195–199, 1970. 2

[12] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. 1

[13] I. Dhillon, Y. Guan, and B. Kulis. *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer, 2004. 3

[14] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004. 3

[15] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996. 1

[16] S. O. Gharan and L. Trevisan. Partitioning into expanders. In *SODA*, pages 1256–1266. SIAM, 2014. 2, 11

[17] O. Goldschmidt and D. S. Hochbaum. Polynomial algorithm for the k-cut problem. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 444–451. IEEE, 1988. 1

[18] N. J. Harvey and N. Olver. Pipage rounding, pessimistic estimators and matrix concentration. In *SODA*, pages 926–945. SIAM, 2014. 1

[19] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001. 1, 9, 11, 20

[20] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004. 2

[21] J. A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. *SIAM Journal on Computing*, 35(4):882–902, 2006. 2

[22] N. L. D. Khoa and S. Chawla. Large scale spectral clustering using resistance distance and spielman-teng solvers. In *Discovery Science*, pages 7–21. Springer, 2012. 2

[23] N. L. D. Khoa and S. Chawla. A scalable approach to spectral clustering with sdd solvers. *Journal of Intelligent Information Systems*, pages 1–20, 2013. 2

[24] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002. 17

[25] A. Kumar and R. Kannan. Clustering with spectral norm and the k-means algorithm. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 299–308. IEEE, 2010. 2

[26] J. R. Lee, S. Oveis Gharan, and L. Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1117–1130. ACM, 2012. 2

[27] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999. 1, 2

[28] A. Louis, P. Raghavendra, P. Tetali, and S. Vempala. Many sparse cuts via higher eigenvalues. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1131–1140. ACM, 2012. 2

[29] F. McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001. 2

[30] M. Meila and J. Shi. A random walks view of spectral segmentation. 2001. 2, 3

[31] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002. 2, 3

[32] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, 2007. 1

[33] J. Peng and Y. Xia. A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, pages 79–96. Springer, 2005. 1

[34] H. Qiu and E. R. Hancock. Clustering and embedding using commute times. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1873–1890, 2007. 2, 3

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000. 3

[36] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. 1

[37] D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2):284–305, 2007. 2

[38] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. 1

[39] L. Trevisan. Is cheeger-type approximation possible for nonuniform sparsest cut? *arXiv preprint arXiv:1303.2730*, 2013. 5

[40] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 975–982. IEEE, 1999. 2, 3

[41] E. Xing, E. P. Xing, M. Jordan, and M. I. Jordan. On semidefinite relaxations for normalized k-cut and connections to spectral clustering. 2003. 3

A Metric Labeling for Getting A Low Conductance Clustering

In this section, we complete show the following theorem, which says that if we start with an ϵ-close clustering to the target clustering, then we can use metric labeling to get another, which has low external conductance. The only difference is that we require stronger separation guarantees between the induced and external expansion parameters. Indeed, throughout Section 6, we only needed $\lambda \geq \Omega(k\Phi)$, here we need $\lambda \geq k^2 \log k \log\log k\Phi$. Formally,

Theorem A.1. *Let $\Gamma^* = \{S_1, S_2, \ldots, S_k\}$ be a clustering satisfying*

(i) $\phi_G(S_i) \leq \phi$,

(ii) $\lambda_2(G[S_i]) \geq \lambda$, *with* $\lambda > 10k^2 \log k \log\log k\phi$

And suppose we have a clustering $\Gamma = \{T_1, T_2, \ldots, T_k\}$ which is very close to Γ^* in the sense that $|S_i \Delta T_i| \leq \epsilon \min(|S_i|, |T_i|)$. Here, $\epsilon = k \cdot \phi/\lambda$. Then, we can recover efficiently a clustering $\Gamma' = \{T'_1, T'_2, \ldots T'_k\}$ where each T'_i has sparsity at most $O(k \log k)\phi$. Note that we are unable to guarantee high induced expansion in the pieces T'_i.

Proof. Let $\Gamma^* = \{S_1, S_2, \ldots S_k\}$ be the optimal clustering, and let $\epsilon = k \cdot \phi/\lambda < 1/(10k \log k \log\log k)$. Moreover, we are given $\Gamma = \{T_1, T_2, \ldots T_k\}$ such that $\forall i$

$$|S_i \Delta T_i| \leq \epsilon \min(|S_i|, |T_i|)\,.$$

We want to convert Γ to $\Gamma' = \{T'_1, T_2, \ldots, T'_k\}$ such that $\forall i$,

$$\phi_G(T'_i) \leq O(k \log k)\phi$$

We will solve this problem using metric labeling, first introduced and studied by Kleinberg and Tardos [24]. In this problem, we are given a graph $G = (V, E)$, a set of labels $L = \{1, 2, \ldots, k\}$, and a distance (semi-)metric between the labels $d(\cdot, \cdot) : L \times L \to \mathbb{R}_{\geq 0}$. Moreover, there is a non-negative cost $c(v, i)$ of assigning a vertex v to label i. The goal is to assign each vertex to a label using a mapping $f : V \to L$ such that the sum of total assignment cost and the total edge-crossing cost, i.e., $\sum_v c(v, f(v)) + \sum_{(u,v)\in E} d(f(u), f(v))$, is minimized. We will appeal to the following theorem due to Kleinberg and Tardos [24].

Theorem A.2. *There is an efficient $O(\log k \log\log k)$ approximation algorithm for the metric labeling problem.*

In our reduction, the set of labels is $L = \{1, 2, \dots k\}$. The assignment cost is defined as follows: $\forall j \in T_i$

$$c(j, i) = 0$$

and $c(j, i' \neq i) = \frac{1}{\epsilon|T_i|}$. The distance metric between labels is as follows: $\forall i, j$

$$d(i, j) = \frac{1}{\phi|T_i|} + \frac{1}{\phi|T_j|}$$

It is easy to check that this is a metric.
Now consider a feasible solution which transports each vertex in T_i to its set S_j in Γ^*, and evaluate the cost incurred by this solution. Due to the bounded symmetric difference property between Γ^* and Γ, in order to get S_i from T_i, at most $\epsilon|T_i|$ points need to be assigned a different label which will together cost at most 1. Hence, summing over all clusters, the total assignment cost is at most k. The total edge-crossing distance is at most $\sum_{i,j\in\Gamma} |E(S_i, S_j)|(\frac{1}{\phi|T_i|} + \frac{1}{\phi|T_j|}) \leq \sum_{i\in S} |E(S, \bar{S})|\frac{1}{\phi|T_i|} \leq \sum_i \frac{|S_i|}{|T_i|} = O(k(1+\epsilon))$. Here again, the size of S_i and T_i are within $(1 \pm \epsilon)$ of each other due to the small symmetric difference.
Hence the optimal cost is at most $3k$. Using Theorem A.2, we will obtain a solution (which can be interpreted as a new clustering $\Gamma' = \{T'_1, T'_2, \dots, T'_k\}$) of metric labeling cost at most $3k\log(k)\log\log k$. We claim that this clustering Γ' satisfies low expansion for each cluster. To see this, we will first argue that for all i, $|T'_i \setminus T_i| \leq \frac{1}{2}|T_i|$. If more than half the points move from T_i then the labeling cost would be at least $\frac{1}{2\epsilon} > 3k\log(k)\log\log k$, since $\epsilon < \frac{1}{10k\log k\log\log k}$, which contradicts our approximation guarantee.
Next we will argue about expansion of each T'_i. We know that $\sum_{i,j} |E(T'_i, T'_j)|\frac{1}{\phi|T_i|} = \sum_i \frac{E(T'_i, \bar{T'_i})}{\phi|T_i|} \leq 3k\log(k)$. This means that individually, $\frac{E(T'_i, \bar{T'_i})}{\phi|T'_i|} \leq 2\frac{E(T'_i, \bar{T'_i})}{\phi|T_i|} \leq 6k\log(k)$ and hence the expansion of T'_i is at most $6k\log(k)\phi$. □

B Constructing a Good Distribution over Partitions: Proof of Theorem 5.1

In order to constructively solve the SDP in Section 5 (note that the SDP has exponentially many variables), we appeal to the Matrix Multiplicative Weights framework due to Arora and Kale [4]. Let c_{JV} denote the integrality gap of the Jain-Vazirani algorithm for k-means clustering. We will solve the primal using Arora-Kale approach to get a feasible solution of value $(1-\delta)(1/c_{JV})$ for any constant δ. Formally, we appeal black-box to the following result of Arora and Kale [4]:

Theorem B.1. *Consider the following SDP optimization problem with target solution value* val *(that is feasible):*

$$\max \mathbf{b} \bullet \mathbf{y}$$
$$\sum_j \mathbf{A}_j y_j \preceq \mathbf{C}$$
$$\mathbf{y} \geq \mathbf{0}$$

Also suppose there is an efficient "oracle" algorithm for solving the following linear system given any positive semi-definite $\mathbf{Z}$*:* $\{\mathbf{y} : \mathbf{y} \geq \mathbf{0}; \mathbf{b} \bullet \mathbf{y} \geq \mathsf{val}; \sum_{j=1}^m (\mathbf{A_j} \bullet \mathbf{Z}) y_j - \mathbf{C} \bullet \mathbf{Z} \leq 0\}$*. Then, for any δ, there is an efficient algorithm which runs in time polynomial in $n, 1/\mathsf{val}, 1/\delta$ and ρ which finds a feasible* $\mathbf{y}$ *with objective value at least* $(1-\delta)\mathsf{val}$*. Here ρ is the width, i.e.,* $\max_j \|A_j y_j - C\|$ *of the system.*

In our case, the $\mathbf{A}_j$'s correspond to the K_Γ's, the vector $\mathbf{y}$ corresponds to the vector $\mathbf{w}$, the vector $\mathbf{b}$ is the all ones vector, the matrix $\mathbf{C}$ corresponds to $L(X)$, and finally we set val to be $1/c_{JV}$. Indeed, due to the nice structure of both the matrix $L(X)$ and the set of matrices K_Γ, it is easy to see that the width ρ is at most 2, which bounds the overall runtime.

Moreover, the oracle algorithm is also simple: plugging in our values of $\mathbf{A}, \mathbf{b}, \mathbf{C}$, we get that it amounts to solving the following system: $\{\mathbf{y} : \mathbf{y} \geq \mathbf{0}; \sum_\Gamma y_\Gamma \geq 1/\mathsf{c_{JV}}; \sum_\Gamma (K_\Gamma \bullet \mathbf{Z}) y_\Gamma - L(X) \bullet \mathbf{Z} \leq 0\}$. Indeed, as mentioned earlier, if we view the psd matrix $\mathbf{Z}$ as the gram matrix of a set of n points P in Euclidean space, then $L(X) \bullet \mathbf{Z}$ is precisely the fractional k-means cost of solution X on dataset P. And using the Jain-Vazirani algorithm, we can find a integer clustering Γ' such that its cost $K_{\Gamma'} \bullet \mathbf{Z} \leq c_{JV} L(X) \bullet \mathbf{Z}$, and so we can set $y_{\Gamma'} = 1/c_{JV}$ and satisfy the system of equations we are checking! This completes the proof of our multiplicative-weights based algorithm. For an informal explanation of how the Arora-Kale algorithm works, please read on.

B.1 Informal Explanation of Matrix Multiplicative Weights

We will now sketch the informal description of how the Matrix Multiplicative Weights algorithm of Arora and Kale [4] works. Readers familiar with the framework can entirely skip this section, as it only provides a rough overview of the steps of the algorithm. The basis of the algorithm is the following identity: $A \succeq B$ if and only if $A \bullet C \geq B \bullet C$ for all PSD-matrices C. So, the Arora-Kale algorithm intuitively views each psd matrix C as an expert, and maintains a distribution $\mathcal{D}_t$ over experts at each time step t (all of this is succinctly implemented in the final algorithm). Initially, $\mathcal{D}_t$ has all its mass on I, and it updates this over time with the goal of in fact trying to show *infeasibility* of the primal SDP! Indeed, suppose it finds a distribution $\mathcal{D}_t$[2] (with expectation $M_t = E_{M \sim \mathcal{D}_t}[M]$) such that the system $\{\mathbf{w} : \sum_\Gamma w_\Gamma \geq 1/c_{JV}, \sum_\Gamma w_\Gamma K_\Gamma \bullet M_t \leq L(X) \bullet M_t\}$ is infeasible, then we would have found our proof of infeasibility. On the other hand, suppose the above system is indeed feasible, and suppose we find a vector $\mathbf{w}_t$ which satisfies these constraints, then the Arora-Kale algorithm updates $\mathcal{D}_t$ to $\mathcal{D}_{t+1}$ by looking at the *reward matrix* $\sum_\Gamma w_{t,\Gamma} K_\Gamma - L(X)$. Indeed, if an expert C is such that $(\sum_\Gamma w_{t,\Gamma} K_\Gamma - L(X)) \bullet C$ is very positive, then we increase its weight a lot, and if its very negative, we decrease its weight.[3]

Then, after T rounds, if we haven't found a proof of infeasibility, then the experts algorithm guarantees that our overall expected reward is almost at least the reward of the best expert in hindsight. Our overall reward is simply,

$$0 \geq \frac{1}{T} \sum_{t=1}^{T} \big(\sum_\Gamma w_{t,\Gamma} K_\Gamma - L(X) \big) \bullet M_t$$

Here, this sum is at most 0 because we have assumed that always found a feasible $\mathbf{w}_t$ for all steps of our algorithm. And the reward of any expert C is

$$\frac{1}{T} \sum_{t=1}^{T} \big(\sum_\Gamma w_{t,\Gamma} K_\Gamma - L(X) \big) \bullet C$$

The experts algorithm guarantees that, for all experts C, our reward is at least its reward, and in particular,

$$0 \geq \frac{1}{T} \sum_{t=1}^{T} \big(\sum_\Gamma w_{t,\Gamma} K_\Gamma - L(X) \big) \bullet C - \delta$$

[2] Of course, the algorithm does all of this implicitly.

[3] Recall that we are still trying to establish a proof of infeasibility, and when we will have failed, we'd have found a good feasible solution!

for some small constant δ. But this says that the solution $\hat{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$ is almost a feasible dual solution. It's dual objective value is at least $1/c_{JV}$ (as the individual $\mathbf{w}_t$'s satisfied this), and moreover, for all psd C, it satisfies

$$\sum_{\Gamma} \hat{w}_\Gamma K_\Gamma \bullet C \leq L(X) \bullet C + \delta$$

Intuitively, this almost means that $\sum_\Gamma \hat{w}_\Gamma K_\Gamma \preceq L(X) + \delta I$. They also show that T only depends polynomially on n and $1/\delta$. And we can make it strictly feasible by scaling the $\hat{\mathbf{w}}$ by a bit while only losing out on a little in the objective function.

Throughout this above analysis, we have assumed that we will always find a feasible $\mathbf{w}_t$ for all of the T steps. Why is that? Indeed, here is where we use the rounding algorithm due to Jain and Vazirani [19]. Let us revisit what the linear system corresponds to. Given a distribution $\mathcal{D}_t$ with expectation $M_t = E_{M\sim\mathcal{D}_t}[M]$ (which is positive semi-definite), it is $\{\mathbf{w} : \sum_\Gamma w_\Gamma \geq 1/c_{JV}, \sum_\Gamma w_\Gamma K_\Gamma \bullet M_t \leq L(X) \bullet M_t\}$. But if we view M_t as the gram matrix of a set of n points P_t in Euclidean space, then $L(X) \bullet M_t$ is precisely the fractional k-means cost of solution X on dataset P_t. And using the Jain-Vazirani algorithm, we can find a integer clustering Γ_t such that its cost $K_{\Gamma_t} \bullet M_t \leq c_{JV} L(X) \bullet M_t$, and so we can set $w_{\Gamma_t} = 1/c_{JV}$ and satisfy the system of equations we are checking! Essentially, this completes the high level overview of the Arora-Kale algorithm.

C Proof of Theorem 7.1

For some r (we will fix r to k later) Γ satisfies:

$$\operatorname{Tr}\left(K_\Gamma \cdot Q_r\right) \leq \kappa \frac{\lambda_2}{\lambda_r - \lambda_2}.$$

It is easy to see that $L^\dagger \preceq \frac{1}{\lambda_r}\left(\frac{\lambda_r-\lambda_2}{\lambda_2} Q_r + K\right)$.We want to upper bound $\lambda_{\max}(K_\Gamma L^\dagger K_\Gamma)$ which is at most $\frac{1}{\lambda_r}\left[\frac{\lambda_r-\lambda_2}{\lambda_2}\operatorname{Tr}(K_\Gamma Q_r) + 1\right] \leq \frac{1}{\lambda_r}\left(\kappa + 1\right)$. By Lemma 3.1, $L \succeq \frac{\lambda_r}{\kappa+1} K_\Gamma$. Substituting $r = k$ yields the first claim. Second one follows from $\lambda_{\min}(L, K_{\Gamma'}) \leq \lambda_k$ for any k-partitioning Γ'.