Approximate Spectral Clustering via Randomized Sketching

Alex Gittens*
Ebay Research
gittens@caltech.edu

Prabhanjan Kambadur
IBM Research
pkambadu@us.ibm.com

Christos Boutsidis
IBM Research
cboutsi@us.ibm.com

Abstract

Spectral clustering is arguably one of the most important algorithms in data mining and machine intelligence; however, its computational complexity makes it a challenge to use it for large scale data analysis. Recently, several approximation algorithms for spectral clustering have been developed in order to alleviate the relevant costs, but theoretical results are lacking. In this paper, we present a novel approximation algorithm for spectral clustering with strong theoretical evidence of its performance. Our algorithm is based on approximating the eigenvectors of the Laplacian matrix using a randomized subspace iteration process, which might be of independent interest.

1 Introduction

Consider clustering the points in Figure 1. The data in this space are non-separable: there is no readily apparent clustering metric which can be optimized over in order to recover this clustering structure. In particular, the two clusters have the same centers (centroids); hence, distance-based clustering methods such as k-means [33] will fail miserably.

Motivated by shortcomings such as these, researchers have produced a body of more flexible and data-adaptive clustering approaches, now known under the umbrella of *spectral clustering*. The crux of all these approaches is that the points to be clustered are viewed as vertices on a graph, and the weights of the edges between the vertices are assigned according to some similarity measure between the points. A new, hopefully separable, representation for the points is then formed by using the eigenvectors of the Laplacian matrix associated with this similarity graph.

We refer the reader to Section 9 in [46] for a discussion on the history of spectral clustering. The first reports in the literature go back to [14, 17]. According to [46], "since then spectral clustering has been discovered, re-discovered, and extended many times". Shi and Malik brought spectral clustering to the machine learning community in their seminal work on Normalized Cuts and Image Segmentation [42]. Since then, spectral clustering has been used extensively in applications in both data analysis and machine learning [4, 13, 31, 25, 28, 50, 43, 34, 46].

The computational bottleneck in spectral clustering is the computation of the eigenvectors of the Laplacian matrix. Motivated by the need for faster algorithms, several techniques have been proposed by researchers in both the theoretical computer science [44] and the machine learning communities [49, 18, 36, 5, 47]. Algorithms from the theory community admit theoretical guarantees, but are complex to describe and implement. On the other hand, algorithms from the machine learning community are empirically sound, but the corresponding theoretical analysis is lacking.

We bridge this theory-practice gap by describing a simple approximation algorithm for spectral clustering with strong theoretical and empirical performance. Our algorithm approximates the key eigenvectors of the

---

*Part of this work was done while the author was with IBM Research during summer 2012.
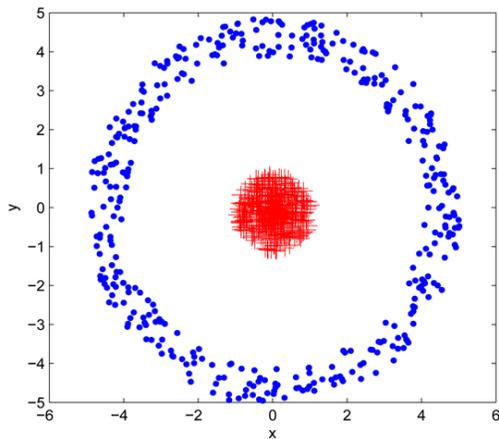
(a)

Figure 1: 2-D data amenable to spectral clustering.

Laplacian matrix of the graph representing the data by using a randomized subspace iteration process. We show, both in theory and in practice, that the running time of the suggested algorithm is sub-cubic and that it finds clusterings that are as good as those obtained using the standard approach.

2   Spectral clustering: the normalized cuts point of view

We review one mathematical formulation of spectral clustering [42]. Denote data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. The goal of clustering is to partition those points into k disjoint sets, for some given k. Towards this end, define a weighted undirected graph $G(V, E)$ with $|V| = n$ nodes and $|E|$ edges as follows: each node in G corresponds to a point $\mathbf{x}_i$; the weight of each edge encodes the similarity between the corresponding two nodes. The similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ whose (i, j)th entry gives the similarity between the two corresponding points is:

$$\mathbf{W}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}}$$

($i \neq j$), where σ is a tuning parameter, and $\mathbf{W}_{ii} = 0$. Given this setup, the spectral clustering problem for $k = 2$ can be viewed as a graph partitioning problem:

Definition 1 (The Spectral Clustering Problem [42]). *Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and $k = 2$ are given. Construct graph $G(V, E)$ as described in the text. Find subgraphs of G, denoted as A and B, to minimize:*

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(B, V)},$$

where

$$\mathrm{cut}(A, B) = \sum_{\mathbf{x}_i \in A, \mathbf{x}_j \in B} \mathbf{W}_{ij};$$

$$\mathrm{assoc}(A, V) = \sum_{\mathbf{x}_i \in A, \mathbf{x}_j \in V} \mathbf{W}_{ij};$$

$$\mathrm{assoc}(B, V) = \sum_{\mathbf{x}_i \in B, \mathbf{x}_j \in V} \mathbf{W}_{ij}.$$

2

This definition generalizes to any $k > 2$. Minimizing the normalized cut objective $\text{Ncut}(A, B)$ in a weighted undirected graph is an NP-Complete problem (see the appendix in [42] for the proof). Motivated by this hardness result, Shi and Malik [42] suggested a relaxation to the problem that is tractable in polynomial time through the Singular Value Decomposition (SVD). First, [42] shows that for *any* G, A, B and partition vector $\mathbf{y} \in \mathbb{R}^n$ with $+1$ to the entries corresponding to A and -1 to the entries corresponding to B it is:

$$4 \cdot \text{Ncut}(A, B) = \mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y} / (\mathbf{y}^T \mathbf{D} \mathbf{y}).$$

Here, $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the diagonal matrix of degree nodes: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Hence, the spectral clustering problem in Definition 1 can be restated as finding such an optimum partition vector $\mathbf{y}$. The real relaxation for spectral clustering asks for a real-valued $\mathbf{y} \in \mathbb{R}^n$ minimizing the same objective:

Definition 2 (The real relaxation for the spectral clustering problem [42])**.** *Given graph G with n nodes, adjacency matrix* $\mathbf{W}$*, and degrees matrix* $\mathbf{D}$ *find* $\mathbf{y}$*:*

$$\mathbf{y} = \underset{\mathbf{y} \in \mathbb{R}^n, \mathbf{y}^T \mathbf{D} \mathbf{1}_n}{\text{argmin}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}}.$$

Once such a $\mathbf{y}$ is found, one can partition the graph into 2 subgraphs by looking at the signs of the elements in $\mathbf{y}$. When $k > 2$, one computes k eigenvectors and then applies k-means clustering on the rows of the matrix containing those eigenvectors (see discussion below).

2.1 An algorithm for spectral clustering

Motivated by the above observations, Ng et. al [31] (see also [48]) suggested the following algorithm for spectral clustering[1]. The input is a set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and the desired number of clusters k. The output is a disjoint partition of the points into k clusters.

Exact Spectral Clustering:

1. Construct the similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ as $\mathbf{W}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}}$ (for $i \neq j$) and $\mathbf{W}_{ii} = 0$.

2. Construct $\mathbf{D} \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.

3. Construct $\tilde{\mathbf{W}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.

4. Find the largest k eigenvectors of $\tilde{\mathbf{W}}$ and assign them as columns to a matrix $\mathbf{Y} \in \mathbb{R}^{n \times k}$.

5. Apply k-means clustering on the rows of $\mathbf{Y}$, and cluster the original points accordingly.

In a nutshell, compute the top k eigenvectors of $\tilde{\mathbf{W}}$ and then apply k-means on the rows of the matrix containing those eigenvectors.

3 Main result

We now describe our algorithm for approximate spectral clustering. Our main idea is to replace step 4 in the above ground-truth algorithm with a procedure that approximates the top k eigenvectors of $\tilde{\mathbf{W}}$. Specifically, we use random projections along with the power iteration. The power iteration for approximating the eigenvectors of matrices is not new: it is a classical technique for eigenvector approximation, known as subspace iteration in the numerical linear algebra literature (see Section 8.2.4 in [19]). The classical subspace iteration implements the initial dimension reduction step with some matrix $\mathbf{S}$ (see the algorithm description

---

[1]Precisely, they suggested one more normalization step on $\mathbf{Y}$ before applying k-means, i.e., normalize $\mathbf{Y}$ to unit row norms, but we ignore this step for simplicity.

below) which has orthonormal columns. This, however, might cause serious problems in the convergence of the algorithm if $\mathbf{S}$ is perpendicular to the top k left principal subspace of the matrix. The analysis of the subspace iteration (see Theorem 8.2.4 in [19]) assumes that this not the case. We overcome this limitation by choosing $\mathbf{S}$ having Gaussian random variables. We call this version of subspace iteration as "randomized subspace iteration". We use it for spectral clustering but the method could be of more general interest [8] [2].

Approximate Spectral Clustering:

1. Construct the similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ as $\mathbf{W}_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}}$ (for $i \neq j$) and $\mathbf{W}_{ii} = 0$.

2. Construct $\mathbf{D} \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.

3. Construct $\tilde{\mathbf{W}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.

4. Let $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times k}$ contain the left singular vectors of

$$\mathbf{B} = (\tilde{\mathbf{W}} \tilde{\mathbf{W}}^T)^p \tilde{\mathbf{W}} \mathbf{S},$$

   with $p \geq 0$, and $\mathbf{S} \in \mathbb{R}^{n \times k}$ being a matrix with $i.i.d$ random Gaussian variables.

5. Apply k-means clustering on the rows of $\tilde{\mathbf{Y}}$, and cluster the original data points accordingly.

3.1 Running time

Step 4 can be implemented in $O(n^2 kp + k^2 n)$ time, as we need $O(n^2 kp)$ time to implement all the matrix-matrix multiplications (right-to-left) and another $O(k^2 n)$ time to find $\tilde{\mathbf{Y}}$. As we discuss in the theorem below, selecting $p \approx O(\ln(kn))$ and assuming that the multiplicative spectral gap of the kth to the $(k+1)$th eigenvalue of $\tilde{\mathbf{W}}$ is large suffices to get very accurate clusterings. This leads to an $O(kn^2 \ln(kn))$ runtime for this step, which is a considerable improvement over the $O(n^3)$ time of the standard approach (see, for example, Section 8.3 in [19]).

3.2 Approximation bound

There are several ways to define the goodness of an approximation algorithm for spectral clustering (see Section 4). Motivated by the work in [23], we adapt the following approach: we *assume* that if, for two matrices $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$, both with orthonormal columns, $\|\mathbf{U} - \mathbf{V}\|_2 \leq \varepsilon$, for some $\varepsilon > 0$, then clustering the rows of $\mathbf{U}$ and the rows of $\mathbf{V}$ with the same method results to the same clustering. Notice that if $\|\mathbf{U} - \mathbf{V}\|_2$ is small, then the distance of each row of $\mathbf{U}$ to the corresponding row of $\mathbf{V}$ is also small. Hence, if for all $i = 1 : n$, $\mathbf{u}_i^T, \mathbf{v}_i^T \in \mathbb{R}^{1 \times k}$ be the ith rows of $\mathbf{U}$ and $\mathbf{V}$, respectively, it is: $\|\mathbf{u}_i^T - \mathbf{v}_i^T\|_2 \leq \|\mathbf{U} - \mathbf{V}\|_2 \leq \varepsilon$. As a result, a distance-based algorithm such as k-means [33] would lead to the same clustering for a sufficiently small $\varepsilon > 0$ (this is equivalent to saying that k-means is robust to small perturbations to the input).

Given this assumption, we argue that the clustering that is obtained by applying k-means on $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times k}$ is similar to the one obtained by applying k-means on $\mathbf{Y} \in \mathbb{R}^{n \times k}$. The following theorem is proved in Section 6.

Theorem 3. *For the orthonormal matrices $\mathbf{Y}$ and $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times k}$, if for some $\varepsilon > 0$ and $\delta > 0$ we choose*

$$p \geq \frac{1}{2} \ln(4\varepsilon^{-1}\delta^{-1} k n^{\frac{1}{2}}) \ln^{-1}\left(\frac{\sigma_k\left(\tilde{\mathbf{W}}\right)}{\sigma_{k+1}\left(\tilde{\mathbf{W}}\right)}\right),$$

---

[2][21] also used the subspace iteration with random gaussian initialization and applied it to the low-rank matrix approximation problem. An important difference there is that they choose a matrix $\mathbf{S}$ with more than k columns; here, we choose $\mathbf{S}$ having exactly k columns. Also, the approximation bounds proved in that paper and in our article are of different form - see the discussion in Section 9.
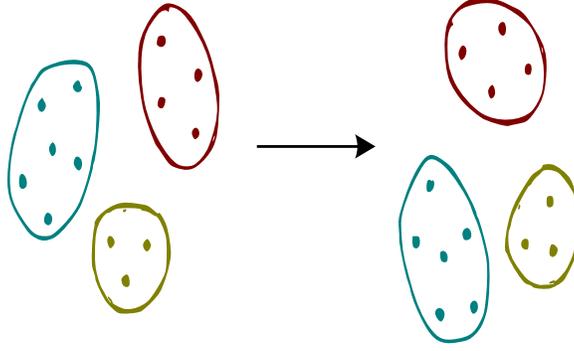
Figure 2: Rows of $\tilde{\mathbf{Y}}$ (left) and rows of $\tilde{\mathbf{Y}}\mathbf{Q}$ (right) in 2-d space.

then, there is orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{k \times k}$ ($\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$) such that with probability at least $1 - e^{-n} - 2.35\delta$,

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2^2 \leq O(\varepsilon^2).$$

This implies the following result for spectral clustering.

Corollary 1. *There is a small $\varepsilon > 0$ for which the rows of $\tilde{\mathbf{Y}}$ and $\mathbf{Y}$ are clustered identically.*

Proof. First, the rows of $\tilde{\mathbf{Y}}\mathbf{Q}$ and $\tilde{\mathbf{Y}}$ are clustered identically since $\mathbf{Q}$ maps $\tilde{\mathbf{Y}}$ to a set of coordinates that is almost only rotated (see Fig. 2). Second, due to the previous theorem and the *assumption* of the robustness of k-means, the rows of $\tilde{\mathbf{Y}}\mathbf{Q}$ and $\mathbf{Y}$ are clustered identically for some $\varepsilon > 0$. Combining those two observations together gives that there is an $\varepsilon > 0$, such that the the rows of $\tilde{\mathbf{Y}}$ and $\mathbf{Y}$ are clustered identically. ∎

Notice that in Corollary 1 we have not identified an exact relationship between ε and how close are the clusterings after applying k-means on $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$. Though this is certainly interesting, we have not been able to come up with such a rigorous relation. Our result relies on an *assumption* that there exists a small $\varepsilon > 0$ such that the two clusterings are identical (see the discussion before Theorem 3). We leave it as an open question if such a rigorous relation exists.

4   Related work

The Hunter and Strohmer result [23]   A result relevant to ours appeared in [23]. Firstly, we follow the same framework for measuring the approximability of an algorithm for spectral clustering (see Theorem 5 and Corollary 6 in [23]). The assumption we use in Corollary 1 in our article was also used in [23]). Hunter and Strohmer did not suggest a specific approximation algorithm; however, they provided a general theory for approximate spectral clustering under perturbations to the eigenvectors used for the clustering (this theory generalizes other similar perturbation bounds, that we disuse below). Specifically, they proved a different version of Theorem 7 of our work (see the proof of Theorem 5 in [23]). Putting their bound in our notation translates to the fact that there exists an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{k \times k}$:

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2 \leq \|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2 + \|\mathbf{Y}^T\tilde{\mathbf{Y}}\|_2.$$

Spectral Clustering via Random Projections   The most relevant algorithmic result to ours is [38]. The authors suggest to reduce the dimension of the data points with random projections before forming the similarity matrix $\mathbf{W}$. Recall that we apply random projections to a scaled version of $\mathbf{W}$. No theoretical results are reported for this approach.

Power Iteration Clustering Another clustering method similar to ours is in [27]. The authors use the power iteration, as we do, on $\tilde{\mathbf{W}}$ but choose $\mathbf{S}$ with just one column. Then, they apply k-means to the resulted vector (even if $k > 2$). No theoretical convergence results are proven for this technique.

Perturbation bounds for Spectral Clustering Define the mis-clustering rate ρ as the number of incorrectly classified data points divided by the total number of points. Suppose $\tilde{\mathbf{L}} = \mathbf{L} + \mathbf{E}$, where $\mathbf{L}$ and $\tilde{\mathbf{L}}$ are the Laplacian matrices of two graphs G and $\tilde{G}$, respectively. G is the graph corresponding to the input points that we want to cluster; $\tilde{G}$ is some other graph, close to G. Assume $k = 2$ and that the clustering is obtained via looking at the signs of the computed eigenvector. Denote by $\tilde{\mathbf{v}}_i$ and $\mathbf{v}_i$ the ith (smallest) eigenvectors of $\tilde{\mathbf{L}}$ and $\mathbf{L}$, respectively. Then, [49, 22, 45] show: $\rho \leq \|\tilde{\mathbf{v}}_2 - \mathbf{v}_2\|_2 \leq (\lambda_2 - \lambda_3)^{-1} \|\mathbf{E}\|_2 + O(\|\mathbf{E}\|_2)$. This bound depends on the "additive gap" of two eigenvalues of the Laplacian matrix. Compare this to our bound in Theorem 3, where we replace the additive gap with the "multiplicative gap".

The Spielman-Teng iterative algorithm Motivated by a result of Mihail [30], Spielman and Teng [44] suggest the following definition for an approximate Fiedler vector: For a Laplacian matrix $\mathbf{L}$ and $0 < \varepsilon < 1$, $\mathbf{v} \in \mathbb{R}^m$ is an ε-approximate Fiedler vector if $\mathbf{v}^T \mathbf{1} = 0$ and

$$\frac{\mathbf{v}\mathbf{L}\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq (1 + \varepsilon)\frac{f^T\mathbf{L}f}{f^Tf} = (1 + \epsilon)\lambda_{n-1}(\mathbf{L}).$$

Here $f \in \mathbb{R}^m$ denotes the Fiedler vector (the eigenvector of $\mathbf{L}$ corresponding to the second smallest eigenvalue). Theorem 6.2 in [44] gives a randomized algorithm to compute such an ε-approximate Fiedler vector w.p. $1 - \delta$, in time $O\left(\text{nnz}(\mathbf{L}) \ln^{27}(\text{nnz}(\mathbf{L})) \ln(\frac{1}{\delta}) \ln(\frac{1}{\varepsilon})\epsilon^{-1}\right)$. This method is based on the fast solvers for linear systems with Laplacian matrices developed in the same paper. Though this is a very strong theoretical result, we are not aware of any efficient implementation.

Other approximation algorithms Yan et al. [49] proposed an algorithm where one first applies k-means to the data to find the so-called centroids and then applies spectral clustering on the centroids and uses this to cluster all the points. Instead of using these centroids, one can select a small set of points from the original points (coreset), apply spectral clustering there and then extend this (e.g. with a nearest neighbor based approach) to all the points [36, 5, 47]. No theoretical results are known for all these approaches. The previous two methods, reduce the dimension of the clustering problem by subsampling data points. Alternatively, one can subsample the similarity matrix $\mathbf{W}$, i.e., use a submatrix of $\mathbf{W}$ and hope that it approximates the entire matrix $\mathbf{W}$ well. This is known as the *Nyström method* [32, 3, 18]. The Nyström method relies on a submatrix to approximate the entire similarity matrix $\mathbf{W}$. There, one usually samples blocks or rows/columns at a time. Alternatively, one can sample the entries themselves. This is the approach of the *spectral clustering on a budget* method [41]. The aim is to randomly select b different entries in the matrix (for some budget constraint b) and store only those. The remaining entries are set to zero, enforcing the approximation to be a sparse matrix whose eigenvectors can be computed efficiently by some iterative eigenvalue solver. For this method, $\rho \lesssim (\lambda_2(\mathbf{L}) - \lambda_3(\mathbf{L}))^{-1} (n^{\frac{5}{3}}b^{-\frac{2}{3}} + n^{\frac{3}{2}}b^{-\frac{1}{2}})$, where $\mathbf{L}$ is the graph Laplacian matrix.

5 Preliminaries

$\mathbf{A}, \mathbf{B}, \ldots$ are matrices; $\mathbf{a}, \mathbf{b}, \ldots$ are column vectors. $\mathbf{I}_n$ is the $n \times n$ identity matrix; $\mathbf{0}_{m \times n}$ is the $m \times n$ matrix of zeros; $\mathbf{1}_n$ is the $n \times 1$ vector of ones. The Frobenius and the spectral matrix-norms are $\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}_{ij}^2$ and $\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2$, respectively. The thin (compact) SVD of $\mathbf{A} \in \mathbb{R}^{m \times n}$ of rank ρ is

$$\mathbf{A} = \underbrace{\left(\begin{array}{cc} \mathbf{U}_k & \mathbf{U}_{\rho-k} \end{array} \right)}_{\mathbf{U_A} \in \mathbb{R}^{m \times \rho}} \underbrace{\left(\begin{array}{cc} \boldsymbol{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\rho-k} \end{array} \right)}_{\boldsymbol{\Sigma_A} \in \mathbb{R}^{\rho \times \rho}} \underbrace{\left(\begin{array}{c} \mathbf{V}_k^T \\ \mathbf{V}_{\rho-k}^T \end{array} \right)}_{\mathbf{V_A^T} \in \mathbb{R}^{\rho \times n}}, \tag{1}$$

with singular values $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_k(\mathbf{A}) \geq \sigma_{k+1}(\mathbf{A}) \geq \ldots \geq \sigma_\rho(\mathbf{A}) > 0$. The matrices $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ and $\mathbf{U}_{\rho-k} \in \mathbb{R}^{m \times (\rho-k)}$ contain the left singular vectors of $\mathbf{A}$; and, similarly, the matrices $\mathbf{V}_k \in \mathbb{R}^{n \times k}$ and $\mathbf{V}_{\rho-k} \in \mathbb{R}^{n \times (\rho-k)}$ contain the right singular vectors. $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k \times k}$ and $\boldsymbol{\Sigma}_{\rho-k} \in \mathbb{R}^{(\rho-k) \times (\rho-k)}$ contain the singular values of $\mathbf{A}$. Also, $\mathbf{A}^+ = \mathbf{V_A} \boldsymbol{\Sigma}_\mathbf{A}^{-1} \mathbf{U}_\mathbf{A}^T$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{A}$. For a symmetric positive definite matrix (SPSD) $\mathbf{A} = \mathbf{B} \mathbf{B}^T$, $\lambda_i(\mathbf{A}) = \sigma_i^2(\mathbf{B})$ denotes the i-th eigenvalue of $\mathbf{A}$. Given a matrix $\mathbf{A}$ with m rows, $\mathbf{P_A} \in \mathbb{R}^{m \times m}$ is the projector matrix that projects in the column span of $\mathbf{A}$, i.e., for any matrix $\mathbf{Z}$ that is a basis for $span(\mathbf{A})$, $\mathbf{P_A} = \mathbf{Z}\mathbf{Z}^+$. For example, $\mathbf{P_A} = \mathbf{U_A}\mathbf{U}_\mathbf{A}^T$. We will also use the following Lemma.

Lemma 4. *Let* $\mathbf{A}$, $\mathbf{B}$ *have n rows and rank $k \leq n$. Then,*

$$\left\|\mathbf{P_A} - \mathbf{P_B}\right\|_2 = \left\|(\mathbf{I} - \mathbf{P_A})\mathbf{P_B}\right\|_2 = \left\|(\mathbf{I} - \mathbf{P_B})\mathbf{P_A}\right\|_2.$$

Proof. This is a simple corollary of [19, Theorem 2.6.1], which states that for any two $n \times k$ orthonormal matrices $\mathbf{W}, \mathbf{Z}$ with $n \geq k$: $\|\mathbf{W}\mathbf{W}^T - \mathbf{Z}\mathbf{Z}^T\|_2 = \|\mathbf{Z}^T\mathbf{W}^\perp\|_2 = \|\mathbf{W}^T\mathbf{Z}^\perp\|_2$. Here, $\mathbf{Z}^\perp \in \mathbb{R}^{n \times (n-k)}$ is such that $[\mathbf{Z}, \mathbf{Z}^\perp] \in \mathbb{R}^{n \times n}$ is a full orthonormal basis; similarly for $\mathbf{W}^\perp$. Applying [19, Theorem 2.6.1] with $\mathbf{W} = \mathbf{U_A}$ and $\mathbf{Z} = \mathbf{U_B}$ and using that $\mathbf{U}_\mathbf{A}^\perp = \mathbf{I}_n - \mathbf{U_A}\mathbf{U}_\mathbf{A}^T$ and $\mathbf{U}_\mathbf{B}^\perp = \mathbf{I}_n - \mathbf{U_B}\mathbf{U}_\mathbf{B}^T$, concludes the proof. ∎

5.1 Random Matrix Theory

The main ingredient in our approximate spectral clustering algorithm is to reduce the dimensions of $\tilde{\mathbf{W}}$ via post-multiplication with a random Gaussian matrix. Here, we summarize two properties of such matrices.

Lemma 5 (The norm of a random Gaussian Matrix [12]). *Let* $\mathbf{A} \in \mathbb{R}^{n \times m}$ *be a matrix with i.i.d. standard Gaussian random variables, where $n \geq m$. Then, for every $t \geq 4$,*

$$\mathbb{P}\{\sigma_1(\mathbf{A}) \geq tn^{\frac{1}{2}}\} \geq e^{-nt^2/8}.$$

Lemma 6 (Invertibility of a random Gaussian Matrix [39]). *Let* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *be a matrix with i.i.d. standard Gaussian random variables. Then, for any $\delta > 0$:*

$$\mathbb{P}\{\sigma_n(\mathbf{A}) \leq \delta n^{-\frac{1}{2}}\} \leq 2.35\delta.$$

6 Proof of Theorem 3

First, we need the following intermediate result.

Lemma 7. *Let* $\mathbf{Y} \in \mathbb{R}^{n \times k}$, $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times k}$ *be the matrices described in the exact and the approximate spectral clustering algorithms, respectively. Then, there is an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n \times k}$ ($\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$) such that:*

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2^2 \leq 2k\|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2^2.$$

Proof. Let $\mathbf{Z} = \tilde{\mathbf{Y}}^T\mathbf{Y} \in \mathbb{R}^{k \times k}$ has rank $\rho_\zeta \leq k$. We use the full SVD of $\mathbf{Z}$, which is: $\mathbf{Z} = \mathbf{U_Z}\boldsymbol{\Sigma_Z}\mathbf{V}_\mathbf{Z}^T$, where all three matrices in the SVD have dimensions $k \times k$. $\mathbf{U_Z}$ and $\mathbf{V_Z}$ are square orthonormal matrices; $\boldsymbol{\Sigma_Z}$ is a diagonal matrix with ρ_ζ non-zero elements along its main diagonal. Let $\mathbf{Q} = \mathbf{U_Z}\mathbf{V}_\mathbf{Z}^T$. Clearly, $\mathbf{Q}$ is a $k \times k$ matrix with full rank and $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$.

We manipulate the term $\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2^2$ as follows: $\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2^2 \leq \|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_\mathrm{F}^2 \leq \|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_\mathrm{F}^2 \leq 2k\|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2^2$. The first inequality is from a simple property which connects the spectral and the Frobenius norm. The last inequality is because $\|\mathbf{X}\|_\mathrm{F}^2 \leq \mathrm{rank}(\mathbf{X})\|\mathbf{X}\|_2^2$, for any $\mathbf{X}$, and the fact that $\mathrm{rank}(\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T) \leq 2k$. To justify this notice that $\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T$ can be written as the product of two matrices each with rank at most $2k$:

$$\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T = \begin{pmatrix} \mathbf{Y} & \tilde{\mathbf{Y}} \end{pmatrix} \begin{pmatrix} \mathbf{Y}^T \\ -\tilde{\mathbf{Y}}^T \end{pmatrix}.$$

We prove the second inequality below. We manipulate the terms $\|\mathbf{Y}-\tilde{\mathbf{Y}}\mathbf{Q}\|_{\mathrm{F}}^2$ and $\|\mathbf{Y}\mathbf{Y}^T-\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_{\mathrm{F}}^2$ separately:

$$
\begin{aligned}
\alpha \;&:=\; \|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{U_Z}\mathbf{V_Z}^T\|_{\mathrm{F}}^2 \\
&=\; \mathrm{Tr}\left(\left(\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{U_Z}\mathbf{V_Z}^T\right)^T \left(\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{U_Z}\mathbf{V_Z}^T\right)\right) \\
&=\; \mathrm{Tr}\left(2\mathbf{I}_k - 2\mathbf{Y}^T\tilde{\mathbf{Y}}\mathbf{U_Z}\mathbf{V_Z}^T\right) \\
&=\; \mathrm{Tr}\left(2\mathbf{I}_k - 2\mathbf{V_Z}\mathbf{\Sigma_Z}\mathbf{V_Z}^T\right) \\
&=\; 2\mathrm{Tr}\left(\mathbf{I}_k\right) - 2\mathrm{Tr}\left(\mathbf{V_Z}\mathbf{\Sigma_Z}\mathbf{V_Z}^T\right) \\
&=\; 2k - 2\mathrm{Tr}\left(\mathbf{\Sigma_Z}\right) \\
&=\; 2k - 2\mathrm{Tr}\left(\tilde{\mathbf{Y}}^T\mathbf{Y}\right)
\end{aligned}
$$

$$
\begin{aligned}
\beta \;&:=\; \|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_{\mathrm{F}}^2 \\
&=\; \mathrm{Tr}\left(\left(\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right)^T \left(\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right)\right) \\
&=\; \mathrm{Tr}\left(\mathbf{Y}\mathbf{Y}^T - 2\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\mathbf{Y}^T + \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right) \\
&=\; \mathrm{Tr}\left(\mathbf{Y}\mathbf{Y}^T\right) - \mathrm{Tr}\left(2\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\mathbf{Y}^T\right) + \mathrm{Tr}\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right) \\
&=\; k - 2\mathrm{Tr}\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\mathbf{Y}^T\right) + k \\
&=\; 2k - 2\mathrm{Tr}\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\mathbf{Y}^T\right)
\end{aligned}
$$

Next, observe that:

$$
\mathrm{Tr}\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\mathbf{Y}^T\right) \le \mathrm{Tr}\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{Y}\right) \le \mathrm{Tr}\left(\tilde{\mathbf{Y}}^T\mathbf{Y}\right),
$$

since $\tilde{\mathbf{Y}}$ has orthonormal columns and $\mathbf{Y}^T$ has orthonormal rows; so, $\alpha \le \beta$, which concludes the proof. ∎

We also need the following result, which we prove in Section 7.1

Corollary 8. *Fix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with rank at least k. Let $p \ge 0$ be an integer and draw $\mathbf{S} \in \mathbb{R}^{n\times k}$, a matrix of i.i.d. standard Gaussian random variables. Fix $\delta \in (0,1)$ and $\varepsilon \in (0,1)$. Let*

$$
\mathbf{\Omega}_1 = (\mathbf{A}\mathbf{A}^T)^p\mathbf{A}\mathbf{S},
$$

and

$$
\mathbf{\Omega}_2 = \mathbf{A}_k.
$$

If

$$
p \ge \frac{1}{2}\ln(4\varepsilon^{-1}\delta^{-1}\sqrt{kn})\ln^{-1}\left(\frac{\sigma_k\left(\mathbf{A}\right)}{\sigma_{k+1}\left(\mathbf{A}\right)}\right),
$$

then with probability at least $1 - e^{-n} - 2.35\delta$,

$$
\|\mathbf{P}_{\mathbf{\Omega}_1} - \mathbf{P}_{\mathbf{\Omega}_2}\|_2^2 \le \frac{\varepsilon^2}{1+\varepsilon^2} = O(\varepsilon^2).
$$

Combining Lemma 7 and Corollary 8 proves Theorem 3. Applying Corollary 8 with $\mathbf{A} = \tilde{\mathbf{W}}$ and

$$p \geq \frac{1}{2} \ln(4\varepsilon^{-1}\delta^{-1}k^{\frac{1}{2}}n^{\frac{1}{2}}) \ln^{-1}\left(\frac{\sigma_k\left(\tilde{\mathbf{W}}\right)}{\sigma_{k+1}\left(\tilde{\mathbf{W}}\right)}\right),$$

gives

$$\|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2^2 \leq O(\varepsilon^2).$$

Rescaling $\varepsilon' = \varepsilon/\sqrt{k}$ and choosing p as

$$p \geq \frac{1}{2} \ln(4\varepsilon^{-1}\delta^{-1}kn^{\frac{1}{2}}) \ln^{-1}\left(\frac{\sigma_k\left(\tilde{\mathbf{W}}\right)}{\sigma_{k+1}\left(\tilde{\mathbf{W}}\right)}\right),$$

gives

$$\|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2^2 \leq O(\varepsilon^2/k).$$

Combining this bound with Theorem 7 we obtain:

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}\|_2^2 \leq 2k\|\mathbf{Y}\mathbf{Y}^T - \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\|_2^2 \leq 2k \cdot O(\varepsilon^2/k) = O(\varepsilon^2),$$

as desired.

7 Analysis of Subspace Iteration

We start with a matrix perturbation bound.

Lemma 9. *Consider three matrices* $\mathbf{D}, \mathbf{C}, \mathbf{E} \in \mathbb{R}^{m \times n}$; *Let* $\mathbf{D} = \mathbf{C} + \mathbf{E}$ *and* $\mathbf{P_C}\mathbf{E} = \mathbf{0}_{m \times n}$. *Then,*

$$\left\|(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2^2 = 1 - \lambda_{\text{rank}(\mathbf{C})}(\mathbf{C}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{C}^T).$$

Proof. Let the columns of $\mathbf{U}$ constitute an orthonormal basis for the range of $\mathbf{C}$ and those of $\mathbf{V}$ constitute an orthonormal basis for the range of $\mathbf{D}$. Denoting the rank of $\mathbf{C}$ by $k = \text{rank}(\mathbf{C})$, we observe that

$$\begin{aligned}
\left\|(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2^2 &= \left\|\mathbf{P_C}(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2 \\
&= \left\|\mathbf{U}\mathbf{U}^T - \mathbf{U}\mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U}\mathbf{U}^T\right\|_2 \\
&= \left\|\mathbf{U}(\mathbf{I} - \mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U})\mathbf{U}^T\right\|_2 \\
&= \left\|\mathbf{I} - \mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U}\right\|_2.
\end{aligned}$$

The first equality follows from the fact that $\left\|\mathbf{X}\right\|_2^2 = \left\|\mathbf{X}\mathbf{X}^T\right\|_2$, and the idempotence of projections. The last inequality uses the unitary invariance of the spectral norm and the fact that $\mathbf{U}^T$ maps the unit ball of its domain onto the unit ball of $\mathbb{R}^k$.

It follows that $\left\|(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2^2 = 1 - \lambda_k(\mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U}) = 1 - \lambda_k(\mathbf{V}^T\mathbf{U}\mathbf{U}^T\mathbf{V}) = 1 - \lambda_k(\mathbf{U}\mathbf{U}^T\mathbf{V}\mathbf{V}^T)$, where our manipulations are justified by the fact that, when the two products are well-defined, the eigenvalues of $\mathbf{AB}$ are identical to those of $\mathbf{BA}$ up to multiplicity of the zero eigenvalues. We further observe that $\lambda_k(\mathbf{U}\mathbf{U}^T\mathbf{V}\mathbf{V}^T) = \lambda_k(\mathbf{P_C}\mathbf{P_D}) = \lambda_k(\mathbf{P_C}\mathbf{D}\mathbf{D}^\dagger) = \lambda_k(\mathbf{P_C}(\mathbf{C} + \mathbf{E})\mathbf{D}^\dagger) = \lambda_k(\mathbf{C}\mathbf{D}^\dagger)$ because $\mathbf{P_C}\mathbf{E} = \mathbf{0}_{m \times n}$. So,

$$\left\|(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2^2 = 1 - \lambda_k(\mathbf{C}\mathbf{D}^\dagger). \tag{2}$$

Recall one expression for $\mathbf{D}^\dagger : \mathbf{D}^\dagger = (\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T$. Using this identity,

$$\lambda_k(\mathbf{C}\mathbf{D}^\dagger) = \lambda_k(\mathbf{P_C}\mathbf{C}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T) = \lambda_k(\mathbf{C}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T\mathbf{P_C})$$

9

$$= \lambda_k(\mathbf{C}(\mathbf{D}^T\mathbf{D})^\dagger(\mathbf{P_C}\mathbf{D})^T).$$

Since $\mathbf{P_C}\mathbf{D} = \mathbf{P_C}(\mathbf{C} + \mathbf{E}) = \mathbf{C}$, we conclude that $\lambda_k(\mathbf{C}\mathbf{D}^\dagger) = \lambda_k(\mathbf{C}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{C}^T)$. Substituting this equality into (2) gives the desired identity. ∎

We continue with a bound on the difference of the projection matrices corresponding to two different subspaces. Given a matrix $\mathbf{A}$, the first subspace is the so-called dominant subspace of $\mathbf{A}$, i.e., the subspace spanned by the top k left singular vectors of $\mathbf{A}$; and the second subspace is the subspace obtained after applying the truncated power iteration to $\mathbf{A}$.

Lemma 10. *Let* $\mathbf{S} \in \mathbb{R}^{n \times k}$ *with* $\mathrm{rank}(\mathbf{A}_k\mathbf{S}) = k$, *then for* $\mathbf{\Omega}_1 = (\mathbf{A}\mathbf{A}^T)^p\mathbf{A}\mathbf{S}$ *and* $\mathbf{\Omega}_2 = \mathbf{A}_k$, *we obtain*

$$\left\|\mathbf{P}_{\mathbf{\Omega}_1} - \mathbf{P}_{\mathbf{\Omega}_2}\right\|_2^2 = 1 - \lambda_k(\mathbf{A}_k\mathbf{S}(\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S})^\dagger\mathbf{S}^T\mathbf{A}_k^T).$$

Proof. In Lemma 9, take $\mathbf{C} = \mathbf{A}_k\mathbf{S}$ and $\mathbf{E} = \mathbf{A}_{\rho-k}\mathbf{S}$ so that $\mathbf{D} = \mathbf{A}\mathbf{S} = \mathbf{C} + \mathbf{E}$. Indeed, since $\mathrm{rank}(\mathbf{A}_k\mathbf{S}) = k = \mathrm{rank}(\mathbf{A}_k)$, the range spaces of $\mathbf{A}_k\mathbf{S}$ and $\mathbf{A}_k$ are identical, so $\mathbf{P_C}\mathbf{E} = \mathbf{P}_{\mathbf{A}_k}\mathbf{A}_{\rho-k}\mathbf{S} = \mathbf{0}_{m \times n}$. Lemma 9 gives

$$\left\|(\mathbf{I} - \mathbf{P_D})\mathbf{P_C}\right\|_2^2 = 1 - \lambda_k(\mathbf{A}_k\mathbf{S}(\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S})^\dagger\mathbf{S}^T\mathbf{A}_k^T).$$

Using Lemma 4 concludes the proof. ∎

Next, we prove a bound as in the previous lemma, but with the right hand side involving different terms.

Theorem 11. *Given* $\mathbf{A} \in \mathbb{R}^{m \times n}$, *let* $\mathbf{S} \in \mathbb{R}^{n \times k}$ *be such that* $\mathrm{rank}(\mathbf{A}_k\mathbf{S}) = k$ *and* $\mathrm{rank}(\mathbf{V}_k^T\mathbf{S}) = k$. *Let* $p \geq 0$ *be an integer and let*

$$\gamma_p = \left\|\mathbf{\Sigma}_{\rho-k}^{2p+1}\mathbf{V}_{\rho-k}^T\mathbf{S}(\mathbf{V}_k^T\mathbf{S})^{-1}\mathbf{\Sigma}_k^{-(2p+1)}\right\|_2.$$

Then, for $\mathbf{\Omega}_1 = (\mathbf{A}\mathbf{A}^T)^p\mathbf{A}\mathbf{S}$, *and* $\mathbf{\Omega}_2 = \mathbf{A}_k$, *we obtain*

$$\left\|\mathbf{P}_{\mathbf{\Omega}_1} - \mathbf{P}_{\mathbf{\Omega}_2}\right\|_2^2 = \frac{\gamma_p^2}{1 + \gamma_p^2}.$$

Proof. Note that $\mathbf{A}_k\mathbf{S}$ has full column rank, so $\mathbf{S}^T\mathbf{A}_k^T\mathbf{A}_k\mathbf{S} \succ \mathbf{0}$ is invertible. It follows that $\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S} = \mathbf{S}^T\mathbf{A}_k^T\mathbf{A}_k\mathbf{S} + \mathbf{S}^T\mathbf{A}_{\rho-k}^T\mathbf{A}_{\rho-k}\mathbf{S} \succ \mathbf{0}$ is also invertible. Also $\mathbf{V}_k^T\mathbf{S}$ is invertible. Hence, $(\mathbf{X} = \mathbf{V}_{\rho-k}\mathbf{\Sigma}_{\rho-k}^2\mathbf{V}_{\rho-k}^T)$,

$$\lambda_k(\mathbf{A}_k\mathbf{S}(\mathbf{S}\mathbf{A}^T\mathbf{A}\mathbf{S})^\dagger\mathbf{S}^T\mathbf{A}_k^T) =$$
$$= \lambda_k\left(\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T\mathbf{S}(\mathbf{S}\mathbf{A}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{V}_k\mathbf{\Sigma}_k\mathbf{U}_k^T\right)$$
$$= \lambda_k\left(\mathbf{\Sigma}_k\mathbf{V}_k^T\mathbf{S}(\mathbf{S}\mathbf{A}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{V}_k\mathbf{\Sigma}_k\right)$$
$$= \lambda_{\max}\left(\left(\mathbf{\Sigma}_k\mathbf{V}_k^T\mathbf{S}(\mathbf{S}\mathbf{A}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{V}_k\mathbf{\Sigma}_k\right)^{-1}\right)^{-1}$$
$$= \lambda_{\max}\left(\left(\mathbf{S}^T\mathbf{V}_k\mathbf{\Sigma}_k\right)^{-1}\left(\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S}\right)\left(\mathbf{\Sigma}_k\mathbf{V}_k^t\mathbf{S}\right)^{-1}\right)^{-1}$$
$$= \lambda_{\max}\left(\mathbf{I}_k + \left(\mathbf{S}^T\mathbf{V}_k\mathbf{\Sigma}_k\right)^{-1}\mathbf{S}^T\mathbf{X}\mathbf{S}(\mathbf{\Sigma}_k\mathbf{V}_k^t\mathbf{S})^{-1}\right)^{-1}$$
$$= \left(1 + \left\|\mathbf{\Sigma}_{\rho-k}\mathbf{V}_{\rho-k}^T\mathbf{S}(\mathbf{\Sigma}_k\mathbf{V}_k^T\mathbf{S})^{-1}\right\|_2^2\right)^{-1}$$
$$= \left(1 + \left\|\mathbf{\Sigma}_{\rho-k}\mathbf{V}_{\rho-k}^T\mathbf{S}(\mathbf{V}_k^T\mathbf{S})^{-1}\mathbf{\Sigma}_k^{-1}\right\|_2^2\right)^{-1}$$
$$= (1 + \gamma_p^2)^{-1}.$$

The result follows by substituting this expression into the expression given in Lemma 10. ∎

The term γ_p has a compelling interpretation: note that

$$\gamma_p \leq \left(\frac{\sigma_{k+1}(\mathbf{A})}{\sigma_k(\mathbf{A})}\right)^{2p+1} \left\|\mathbf{V}_{\rho-k}^T \mathbf{S}\right\|_2 \left\|(\mathbf{V}_k^T \mathbf{S})^{-1}\right\|_2;$$

the first term, a power of the ratio of the $(k+1)$th and kth singular values of $\mathbf{A}$, is the inverse of the spectral gap; the larger the spectral gap, the easier it should be to capture the dominant k dimensional range-space of $\mathbf{A}$ in the range space of $\mathbf{AS}$. Increasing the power increases the beneficial effect of the spectral gap. The remaining terms reflect the importance of choosing an appropriate sampling matrix $\mathbf{S}$: one desires $\mathbf{AS}$ to have a small component in the direction of the undesired range space spanned by $\mathbf{U}_{\rho-k}$; this is ensured if $\mathbf{S}$ has a small component in the direction of the corresponding right singular space. Likewise, one desires $\mathbf{V}_k$ and $\mathbf{S}$ to be strongly correlated so that $\mathbf{AS}$ has large components in the direction of the desired space, spanned by $\mathbf{U}_k$.

To apply Theorem 11 practically to estimate the dominant k-dimensional left singular space of $\mathbf{A}$, we must choose an $\mathbf{S} \in \mathbb{R}^{n \times k}$ that satisfies $\mathrm{rank}(\mathbf{A}_k \mathbf{S}) = k$ and bound the resulting γ_p. The next corollary gives a bound on γ_p when $\mathbf{S}$ is chosen to be a matrix of i.i.d. standard Gaussian r.v.s.

7.1 Proof of Corollary 8

Clearly $\mathbf{A}_k \mathbf{S}$ has rank k (the rank of the product of a Gaussian matrix and an arbitrary matrix is the minimum of the ranks of the two matrices). Also, $\mathbf{V}_k^T \mathbf{S}$ is a $k \times k$ random Gaussian matrix (the product of a random Gaussian matrix with an orthonormal matrix is also a random Gaussian matrix) and hence is invertible almost surely according to Lemma 6. So, Theorem 11 is applicable. Estimate γ_p as:

$$
\begin{aligned}
\gamma_p &\leq \left\|\mathbf{\Sigma}_{\rho-k}^{2p+1}\right\|_2 \left\|\mathbf{V}_{\rho-k}^T \mathbf{S}\right\|_2 \left\|(\mathbf{V}_k^T \mathbf{S})^{-1}\right\|_2 \left\|\mathbf{\Sigma}_k^{-(2p+1)}\right\|_2 \\
&= \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2p+1} \frac{\sigma_{\max}\left(\mathbf{V}_{\rho-k}^T \mathbf{S}\right)}{\sigma_{\min}\left(\mathbf{V}_k^T \mathbf{S}\right)} \\
&\leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2p+1} \frac{\sigma_{\max}\left(\mathbf{V}^T \mathbf{S}\right)}{\sigma_{\min}\left(\mathbf{V}_k^T \mathbf{S}\right)} \\
&\leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2p+1} \frac{4\sqrt{n-k}}{\delta/\sqrt{k}} \\
&= \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2p+1} \cdot 4\delta^{-1}\sqrt{k(n-k)}.
\end{aligned}
$$

Here, $\mathbf{V} = [\mathbf{V}_k, \mathbf{V}_k^\perp]$ is a square matrix with orthonormal columns. The last inequality follows from the fact that $\mathbf{V}^T \mathbf{S}$ and $\mathbf{V}_k^T \mathbf{S}$ are matrices of standard Gaussians, and then using Lemma 5 with $t = 4$ and Lemma 6. The guarantees of both Lemmas apply with probability at least $1 - e^{-n} - 2.35\delta$, so

$$\gamma_p \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2p+1} \cdot 4\delta^{-1} \cdot \sqrt{kn},$$

holds with at least this probability. Hence, to ensure $\gamma_p \leq \varepsilon$, it suffices to choose p satisfying

$$p \geq \tfrac{1}{2}\left(\ln(4\varepsilon^{-1}\delta^{-1}\sqrt{kn})\ln^{-1}\left(\frac{\sigma_k(\mathbf{A})}{\sigma_{k+1}(\mathbf{A})}\right) - 1\right).$$

In this case, $\|\mathbf{P_Y} - \mathbf{P_{\tilde{Y}}}\|_2^2 \leq \frac{\varepsilon^2}{1+\varepsilon^2} = O(\varepsilon^2)$.

Name	n	d	#nnz	k
SatImage	4435	36	158048	6
Segment	2310	19	41469	7
Vehicle	846	18	14927	4
Vowel	528	10	5280	11

Table 1: Datasets from the libSVM multi-class classification page [10] that were used in our experiments.

8 Experiments

The selling point of approximate spectral clustering is the reduction in time to solution without sacrificing the quality of the clustering. Computationally, approximation saves time as we compute the eigenvectors of a smaller matrix — $(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)^p\tilde{\mathbf{W}}\mathbf{S} \in \mathbb{R}^{n \times k}$ — as opposed to $\tilde{\mathbf{W}} \in \mathbb{R}^{n \times n}$; however, we do incur the cost of computing this smaller matrix itself. The exponent p is a parameter to the spectral clustering algorithm that determines the quality of the approximate solution (see Theorem 3). Accordingly, the goal of our experiments is to: (1) establish that the quality of clustering does not suffer due to the approximation, and (2) show the effect of p on the solution quality and the running time. Note that our experiments are not meant to produce the best quality clusters for the test data; this requires fine-tuning several parameters such as the kernel bandwidth for $\mathbf{W}$, which is out of the scope of this paper.

To conduct our experiments, we developed MATLAB versions of the exact and approximate algorithms (Section 3 and 2.1). To compute $\mathbf{W}$, we use the heat kernel:

$$\ln(\mathbf{W}_{ij}) = \frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{\sigma_{ij}},$$

where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{x}_j \in \mathbb{R}^d$ are the data points and σ_{ij} is a tuning parameter; σ_{ij} is determined using the self-tuning method described in [50]. For the partial singular value decomposition of $\tilde{\mathbf{W}}$, we used IRLBA [2] due to it's excellent support for sparse decompositions. To compute $\tilde{\mathbf{W}}\mathbf{S}$, we used MATLAB's built-in `normrnd` function to generate a Gaussian $\mathbf{S}$. For the complete singular value decomposition of $(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)^p\tilde{\mathbf{W}}\mathbf{S}$, we use MATLAB's built-in `svd` function. We also use MATLAB's built-in `kmeans` function for the final clustering; the options 'EmptyAction', 'singleton' were turned on in addition to specifying a maximum of 100 iterations and 10 'Replicates'. The output labeling order of both the exact and the approximate algorithms do not necessarily match the true labels. To avoid comprehensively testing all k permutations of the output labels for optimality with the true labels, we use the Hungarian algorithm [24]. As a measure of quality, we use the clustering-rate (CR), which can be defined as the (normalized) number of correctly clustered objects; that is $CR = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_i$, where $\mathbb{I}_i$ is an indicator variable that is 1 when an object is correctly clustered and 0 when it is not. We ran our experiments on four multi-class datasets from the libSVMTools webpage (Table 1). All experiments were carried out using MATLAB 7.8.0.347 (R2009a) on a 64-bit linux server running 2.6.31-23. Finally, to highlight the timing differences between using the exact and the approximate algorithm, we only report the time taken to compute the singular value decomposition; all the other steps are the same for both algorithms.

To determine the quality of the clustering obtained by our approximate - randomized - algorithm, we ran 10 iterations on each of the 4 datasets listed in Table 1, and report results for the one which minimizes the clustering rate CR%. This was done to minimize the effect of the randomness of $\mathbf{S}$. In each iteration, we varied p from $[0, 10]$.

First, we explore how the error in Theorem 3 behaves while p grows. Recall that we compute $\hat{\mathbf{Q}}$ as

$$\hat{\mathbf{Q}} = \underset{\mathbf{Q}^T\mathbf{Q}=\mathbf{I}_k}{\operatorname{argmin}} ||\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{Q}||_{\mathrm{F}},$$

i.e., $\hat{\mathbf{Q}} = \mathbf{U_B}\mathbf{V_B}^T$ ($\mathbf{B} = \mathbf{Y}^T\tilde{\mathbf{Y}}$). This is the famous Procrustes problem. Figure 3 confirms that the approxi-

Name	Exact		Approximate				
			p=2		Best under exact time		
	CR%	time (secs)	CR%	time (secs)	CR%	time (secs)	p
SatImage	69.33	1.24	53.59	0.29	*76.46*	.88	6
Segment	56.83	1.185	29.30	0.126	*58.91*	0.53	10
Vehicle	41.25	0.024	*43.97*	0.009	—	—	—
Vowel	36.55	0.016	29.35	0.003	36.28	0.010	8

Table 2: Clustering rate (CR %, best of 10 runs) for exact and approximate algorithms on the datasets from Table 1. For approximate algorithms, we report two sets of numbers: (1) the CR% achieved at $p = 2$ along with the time and (2) the best CR% achieved while taking less time than the exact algorithm, where time reported (in seconds) is to complete the power iterations and the eigen-decomposition of $(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)^p\tilde{\mathbf{W}}S$ with p. A "—" in the right three columns indicates that the best CR% under exact time for the approximate algorithms was also achieved at $p = 2$ (Vehicle). We see that the approximate algorithms perform *at least* as good as their exact counterparts. Note that CR% is heavily dependent on the quality of $\mathbf{W}$, which we did not optimize.
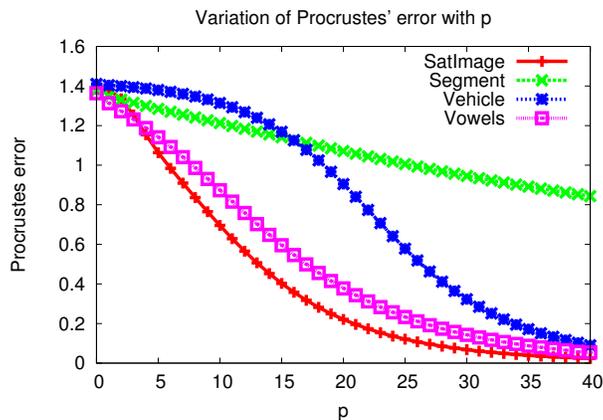

Figure 3: Reduction in the error $||\mathbf{Y} - \tilde{\mathbf{Y}}\hat{\mathbf{Q}}||_2$ (see Theorem 3 and text below) with increase in p.
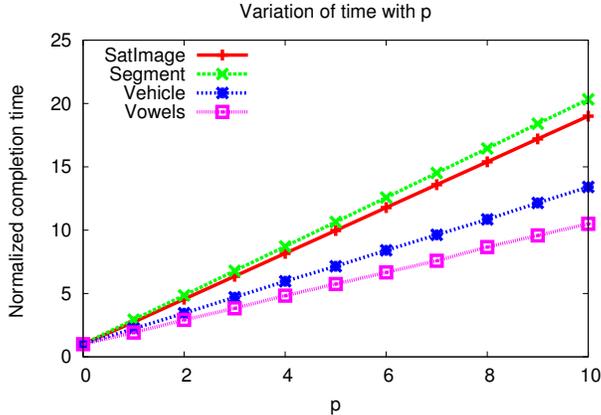
Figure 4: Variation of time (normalized to p=0) to complete the power iterations and the eigen-decomposition of $(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T)^p\tilde{\mathbf{W}}S$. The baseline times (at p=0) are SatImage (0.0648 secs), Segment (0.0261 secs), Vehicle (0.0027 secs), and Vowel (0.0012 secs). Times reported are in seconds.

mation error decreases with an increase in p.

Next, we report results for the clustering rate CR% in Table 2. In columns 2 and 3, we see the results for the exact algorithm, which serves as our baseline for quality and performance. In columns 4 and 5, we see the CR% with 2 power iterations ($p = 2$) along with the time required for the power iterations and the eigendecomposition. Immediately, we see that at $p = 2$, the Vehicle dataset achieves good performance by beating or matching the performance of the exact algorithm, resulting in a 2.5x speedup. The only outlier is the Segment dataset, which achieves poor CR% at $p = 2$. In columns 6—8, we report the best CR% that was achieved by the approximate algorithm while staying under the time taken by the exact algorithm; we also report the value of p and the time taken (in seconds) for this result. A "—" in the right three columns indicates that the best CR% under exact time for the approximate algorithms was also achieved at $p = 2$ (Vehicle). We see that even when constrained to run in less time than the exact algorithm, the approximate algorithm bests the CR% achieved by the exact algorithm. For example, at $p = 10$, the approximate algorithm reports 76.46% for Segment, while still providing a 1.4x speedup. We can see that in many cases, 2 or fewer power iterations ($p \leq 2$) suffice to get good quality clustering (Segment dataset being the exception). Note that the clustering quality also depends on the quality of $\mathbf{W}$ and optimizations in k-means, both of which we did not attempt to optimize.

Next, we turn our attention to the relation of p to the time to compute the power iterations followed by the eigendecomposition; this is depicted in Figure 4. All the times are normalized by the time taken at $p = 0$ to enable reporting numbers for all datasets on the same scale; notice that when $p = 0$, the approximation to $\tilde{\mathbf{W}}$ is merely $\tilde{\mathbf{W}}\mathbf{S}$. As expected, as p increases, we see a linear increase in the time.

Finally, we report the variation of the clustering rate with p in Figure 5; all the rates are normalized by the CR% at $p = 0$. We observe that the clustering quality is comparable to the ground truth. Although one would expect the clustering CR% to increase monotonically while we increase p, this is not entirely true. Recall that we assumed that $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are clustered identically as $\varepsilon \to 0$. However, in our setting ε is not approaching 0 (see Figure 3), hence k-means finds (slightly) different clusters for small ε, e.g., $\varepsilon = 0.1$.

9   Concluding Remarks

The proposed approximation algorithm for spectral clustering builds on a family of randomized, sketching-based algorithms which, in recent years, led to similar running time improvements to other classical machine learning and linear algebraic problems: (i) least-squares regression [37, 6, 16, 11]; (ii) linear equations with
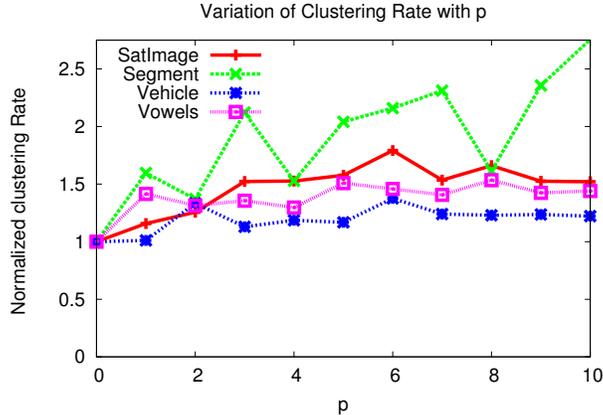
14

Figure 5: Variation of clustering rate (normalized to $p = 0$) with p. The baseline clustering rates (at p=0) are SatImage (42.68%), Segment (21.38%), Vehicle (32.97%), and Vowel (22.35%). Clustering rates reported are the best out of 10 runs.

Symmetric Diagonally Dominant matrices [26] (iii) low-rank approximation of matrices [29, 20, 7, 11]; (v) matrix multiplication [15, 40]; (vi) k-means clustering [9]; (vii) support vector machines [35]; (viii) Canonical Correlation Analysis (CCA) [1].

Our algorithm is similar to many of those methods in regard to reducing the dimensions of the input matrix via random projections. However, to prove the theoretical behavior of the algorithm we derived some new results that could be of independent interest. The main novel technical ingredient is that we show that the difference between $\mathbf{Y}$ and an orthonormal transformation of $\tilde{\mathbf{Y}}$ is bounded (see Theorem 3). Previous work analyzes the approximation error after projecting $\tilde{\mathbf{W}}$ on $\mathbf{P}_{\tilde{\mathbf{Y}}}$. Specifically, [21] argues that $\|\tilde{\mathbf{W}} - \mathbf{P}_{\tilde{\mathbf{Y}}}\tilde{\mathbf{W}}\|_2$ is small.

From a different perspective, our algorithm can be viewed as a randomized version of the classical subspace iteration method from the numerical linear algebra literature. In classical subspace iteration, $\mathbf{S}$ is chosen as any orthonormal matrix; however, this might be problematic if $\mathbf{S}$ is perpendicular to the principal left subspace of the input matrix. We overcome this limitation by choosing $\mathbf{S}$ a matrix with Gaussian random variables. Our analysis indicates that the subspaces generated by our method are as good as those generated by the standard approach. [21] also uses the same randomized version of subspace iteration. An important difference between [21] and our algorithm is the number of dimensions of the gaussian matrix that we used for the dimensionality reduction. Our analysis is restricted to choose $\mathbf{S}$ having k columns. For example, we don't know how to prove Theorem 11 by choosing $\mathbf{S}$ with more than k columns. We leave it as a topic of future investigation whether oversampling (i.e., choosing $\mathbf{S}$ with more than k columns) will improve the bounds in Theorem 11 and Corollary 8.

Acknowledgements

References

[1] H. Avron, C. Boutsidis, S. Toledo, and A. Zouzias. Efficient dimensionality reduction for canonical correlation analysis. In *In ICML*, 2013.

[2] J. Baglama and L. Reichel. Restarted block lanczos bidiagonalization methods. *Numerical Algorithms*, 43(3):251–272, 2006.

[3] C. Baker. *The numerical treatment of integral equations*, volume 13. Clarendon Press, 1977.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[5] J. Bezdek, R. Hathaway, J. Huband, C. Leckie, and R. Kotagiri. Approximate clustering in very large relational data. *International journal of intelligent systems*, 21(8):817–841, 2006.

[6] C. Boutsidis and P. Drineas. Random projections for the nonnegative least-squares problem. *Linear Algebra and its Applications*, 431(5-7):760–771, 2009.

[7] C. Boutsidis and A. Gittens. Improved matrix algorithms via the subsampled randomized hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, 2013.

[8] C. Boutsidis and M. Magdon-Ismail. Faster svd-truncated regularized least-squares, preprint, `http://arxiv.org/abs/1401.0417`, 2014.

[9] C. Boutsidis, A. Zouzias, and P. Drineas. Random projections for k-means clustering. In *In NIPS*, 2010.

[10] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[11] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *STOC*, 2013.

[12] K. R. Davidson and S. J. Szarek. Local Operator Theory, Random Matrices and Banach Spaces. In W. B. Johnson and J. Lindenstrauss, editors, *Handbook of the Geometry of Banach Spaces*, volume 1. Elsevier Science, 2001.

[13] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274. ACM, 2001.

[14] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[15] P. Drineas and R. Kannan. Fast Monte-Carlo algorithms for approximate matrix multiplication. In *FOCS*, 2001.

[16] P. Drineas, M. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):217–249, 2011.

[17] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[18] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):214–225, 2004.

[19] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[20] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[21] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[22] L. Huang, D. Yan, M. Jordan, and N. Taft. Spectral clustering with perturbed data. *NIPS*, 2008.

[23] B. Hunter and T. Strohmer. Performance analysis of spectral clustering on compressed, incomplete and inaccurate measurements. Preprint: `http://arxiv.org/abs/1011.0997`, 2011.

[24] R. Jonker and T. Volgenant. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, 1986.

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

[26] I. Koutis, G. Miller, and R. Peng. Approaching optimality for solving SDD systems. *In FOCS*, 2010.

[27] F. Lin and W. W. Cohen. Power iteration clustering. In *ICML*, 2010.

[28] R. Liu and H. Zhang. Segmentation of 3d meshes through spectral clustering. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 298–305. IEEE, 2004.

[29] P. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 2010.

[30] M. Mihail. Conductance and convergence of Markov chains-A combinatorial treatment of expanders. In *FOCS*, 1989.

[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.

[32] E. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.

[33] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *FOCS*, 2006.

[34] A. Paccanaro, J. A. Casbon, and M. A. Saqi. Spectral clustering of protein sequences. *Nucleic acids research*, 34(5):1571–1580, 2006.

[35] S. Paul, C. Boutsidis, M. Magdon-Ismail, and P. Drineas. Random projections for support vector machines. In *In AISTATS*, 2013.

[36] M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. *NIPS*, 2005.

[37] V. Rokhlin and M. Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36):13212, 2008.

[38] T. Sakai and A. Imiya. Fast spectral clustering with random projection and sampling. In *Machine Learning and Data Mining in Pattern Recognition*, pages 372–384. Springer, 2009.

[39] A. Sankar, D. A. Spielman, and S.-H. Teng. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM Journal on Matrix Analysis and Applications*, 28(2):446–476, 2006.

[40] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *In FOCS*, 2006.

[41] O. Shamir and N. Tishby. Spectral clustering on a budget. *AISTATS*, 2011.

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

[43] S. Smyth and S. White. A spectral clustering approach to finding communities in graphs. In *SDM*, 2005.

[44] D. Spielman and S.-H. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *ArxiV preprint, http://arxiv.org/pdf/cs/0607105v4.pdf*, 2009.

[45] G. Stewart and G. Stewart. *Introduction to matrix computations*, volume 441. Academic press New York, 1973.

[46] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[47] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek. Approximate spectral clustering. *Advances in Knowledge Discovery and Data Mining*, pages 134–146, 2009.

[48] 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.

[49] D. Yan, L. Huang, and M. Jordan. Fast approximate spectral clustering. In *KDD*, 2009.

[50] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.