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 random projections, a.k.a randomized sketching. Our experimental results demonstrate that the proposed approximation algorithm compares remarkably well to the exact algorithm.

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 these particular clustering structures. In particular, in both cases, the two clusters have the same centers (centroids); hence, distance-based clustering methods such as \(k\)-means [32] will fail miserably.

Motivated by shortcomings such as these, researchers have produced a body of more flexible and data-adaptive clustering approaches, 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 two 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 a Laplacian matrix associated with the similarity graph as the coordinates for the points. We refer the reader to Section 9 in [45] for a discussion on the history of spectral clustering. The first reports in the literature go back to [13, 16]. According to [45], “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 [41] (see Section 2). Since then, spectral clustering has been used extensively in applications in both data analysis and machine learning [4, 12, 30, 24, 27, 49, 42, 33, 45].

The computational bottleneck in spectral clustering is the computation of the eigenvectors of the Laplacian matrix. Motivated by the need for faster algorithms, several algorithms have been proposed by researchers in both the theoretical computer science [43] and the machine learning communities [48, 17, 35, 5, 46]. 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 gap by describing a simple approximation algorithm for spectral clustering with strong theoretical and empirical performance. Our algorithm approximates the key eigenvectors of the Laplacian matrix of the graph representing the data by using a truncated subspace iteration process with a Gaussian random initialization. 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 standard approaches.

* Part of this work was done while the author was with IBM Research during summer 2012.
2 The spectral clustering problem: the normalized cuts point of view

First we review one mathematical formulation of spectral clustering. Denote data points \( x_1, x_2, \ldots, x_n \) in \( d \)-dimensional space. The goal of clustering is to cluster those points into \( k \) disjoint sets. 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 \( x_i \) \((i = 1:n)\) in the data; the weight of each edge encodes the similarity between the corresponding two nodes. The similarity matrix \( W \in \mathbb{R}^{n \times n} \) whose \((i,j)\)th entry gives the similarity between the two corresponding points is,

\[
W_{ij} = e^{-\frac{|x_i - x_j|^2}{\sigma}} \quad (i \neq j),
\]

where \( \sigma \) is a tuning parameter, and \( W_{ii} = 0 \). Given this setup, the spectral clustering problem for \( k = 2 \) can be viewed as a graph partitioning problem (the definition below admits an easy generalization when \( k > 2 \)).

**Definition 1 (The Spectral Clustering Problem [41]).** Let \( x_1, x_2, \ldots, x_n \in \mathbb{R}^d \) are given. Construct graph \( G(V, E) \) as described in the text above. Find subgraphs of \( G \), denoted as \( A \) and \( B \), to minimize the objective

\[
\text{Ncut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)},
\]

where \( \text{cut}(A,B) = \sum_{x_i \in A, x_j \in B} W_{ij} \); \( \text{assoc}(A,V) = \sum_{x_i \in A, x_j \in V} W_{ij} \); \( \text{assoc}(B,V) = \sum_{x_i \in B, x_j \in V} W_{ij} \).

Minimizing the normalized cut objective \( \text{Ncut}(\cdot) \) in a weighted undirected graph is an NP-Complete problem (see the appendix in [41] for the proof). Motivated by this hardness result, Shi and Malik [41] suggested a relaxation to the problem that is tractable in polynomial time through the SVD. First, [41] shows that for any \( G, A, B \) and partition vector \( y \in \mathbb{R}^n \) with \(+1\) to the entries corresponding to \( A \) and \(-1\) to the entries corresponding to \( B \) it is: \( 4 \cdot \text{Ncut}(A,B) = y^T(D - W)y / (y^TDy) \). Here, \( D \in \mathbb{R}^{n \times n} \) is the diagonal matrix of degree nodes: \( D_{ii} = \sum_j W_{ij} \). Hence, the spectral clustering problem in Definition 1 can be restated as finding such an optimum partition vector \( y \). The real relaxation for spectral clustering asks for a real-valued \( y \in \mathbb{R}^n \) minimizing the same objective:

**Definition 2 (The Real Relaxation for the Spectral Clustering Problem [41]).** Given graph \( G \) with \( n \) nodes, adjacency matrix \( W \), and degrees matrix \( D \) find \( y: y = \arg\min_{y \in \mathbb{R}^n} \frac{y^T(D-W)y}{y^TDy} \).

Once such a \( y \) is found, we partition the graph into 2 subgraphs by looking at the signs of the elements in \( y \). When \( k > 2 \), we first compute \( k \) eigenvectors and then apply \( k \)-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, which are due to [41], Ng, Jordan, and Weis [30] (see also [47]) suggested the following algorithm for spectral clustering, which we consider as the ground-truth algorithm. The input to this spectral clustering algorithm is a set of points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ 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 $W \in \mathbb{R}^{n \times n}$ as $W_{ij} = e^{-\|x_i - x_j\|^2/\sigma}$ (for $i \neq j$) and $W_{ii} = 0$.
2. Construct $D \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $D_{ii} = \sum_j W_{ij}$.
3. Construct $\tilde{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.
4. Find the largest $k$ eigenvectors of $\tilde{W}$ and assign them as columns to a matrix $Y \in \mathbb{R}^{n \times k}$.
5. Apply $k$-means clustering on the rows of $Y$, and cluster the original points accordingly.

In a nutshell, compute the top $k$ eigenvectors of $\tilde{W}$ 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 $\tilde{W}$. Specifically, we use random projections along with the power iteration. This method 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 [18]). The linear algebra community is characteristically concerned with obtaining results with as high numerical precision as feasible, so when subspace iteration is applied, the stopping condition usually depends on a priori tests of accuracy rather than a fixed number of iterations. Recently, researchers have realized the benefits of truncating these classical iterative processes after a small number of iterations in large-dimensional settings where each iteration may be quite expensive. The survey [20] gives an overview of these truncated processes in the context of low-rank matrix approximation. Our contribution is the application of this idea of truncating the iterative process of subspace iteration to spectral clustering and a novel analysis that shows that clusterings are preserved remarkably well.

**Approximate Spectral Clustering**

1. Construct the similarity matrix $W \in \mathbb{R}^{n \times n}$ as $W_{ij} = e^{-\|x_i - x_j\|^2/\sigma}$ (for $i \neq j$) and $W_{ii} = 0$.
2. Construct $D \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $D_{ii} = \sum_j W_{ij}$.
3. Construct $\tilde{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.
4. Let $\tilde{Y} \in \mathbb{R}^{n \times k}$ contain the left singular vectors of $B = (\tilde{WW}^T)^p\tilde{W}$, with $p \geq 0$, and $S \in \mathbb{R}^{n \times k}$ being a matrix with i.i.d random Gaussian variables.
5. Apply $k$-means clustering on the rows of $\tilde{Y}$, and cluster the original data points accordingly.

---

1 Precisely, they suggested one more normalization step on $Y$ before applying $k$-means, i.e., normalize $Y$ to unit row norms, but we ignore this step for simplicity.
Step 4 can be implemented in $O(n^2 kp + k^2 n)$ time, as we need $O(n^2 p)$ time to implement all the matrix-matrix multiplications (right-to-left) and another $O(k^2 n)$ time to find $\tilde{Y}$. As we discuss in the theorem below, selecting $p \approx \ln(kn)$ and assuming that the multiplicative spectral gap of the $k$th to the $(k+1)$th eigenvalue of $\tilde{W}$ 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 [18]).

There are several ways to define the goodness of an approximation algorithm for spectral clustering (see Section 4). Motivated by the work in [22] we adapt the following approach: we assume that if, for two matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{n \times k}$, both with orthonormal columns, $\|U - V\|_2 \leq \varepsilon$, for some arbitrarily small $\varepsilon > 0$, then clustering the rows of $U$ and the rows of $V$ with the same method should result to the same clustering. Notice that if $\|U - V\|_2$ is small, then the distance of each row of $U$ to the corresponding row of $V$ is also small, i.e., for all $i = 1: n$, let $u_i^T, v_i^T \in \mathbb{R}^{1 \times k}$ be the $i$th rows of $U$ and $V$, then,

$$\|u_i - v_i\|_2 \leq \|U - V\|_2 \leq \varepsilon.$$ 

Hence, a distance-based algorithm such as $k$-means [32] would lead to the same clustering as $\varepsilon \to 0$ (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 $\tilde{Y} \in \mathbb{R}^{n \times k}$ is similar to the one obtained by applying $k$-means on $Y \in \mathbb{R}^{n \times k}$. The following theorem is proved in Section 6.

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

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

then there is an orthonormal matrix $Q \in \mathbb{R}^{k \times k}$ such that with probability at least $1 - e^{-n} - 2.35\delta$,

$$\|Y - \tilde{Y}Q\|^2_2 \leq \frac{\varepsilon^2}{1 + \varepsilon^2} = O(\varepsilon^2).$$

**Corollary 4.** As $\varepsilon \to 0$, the rows of $\tilde{Y}$ and $Y$ are clustered identically with the $k$-means method.

**Proof.** The rows of $\tilde{Y}Q$ and $\tilde{Y}$ are clustered identically since $Q$ maps $\tilde{Y}$ to a set of coordinates that is almost only rotated (see Fig. 2), by the previous theorem. Thus the rows of $\tilde{Y}$ and $Y$ admit the same clustering, as $\varepsilon \to 0$. 

\[\Box\]
4 Comparison with related work

The Hunter and Strohmer result [22]. The result that is most relevant to ours appeared in [22]. Firstly, we follow the exact same framework for measuring the approximability of an algorithm for spectral clustering (see Theorem 5 and Corollary 6 in [22]). Hunter and Strohmer do not suggest a specific approximation algorithm; however, they do provide 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 prove a weaker version of Theorem 7 of our work (see the proof of Theorem 5 in [22]). Putting their bound in our notation translates to the fact that there exists an orthonormal matrix $Q \in \mathbb{R}^{k \times k}$ such that,

$$||Y - \hat{Y}Q||_2 \leq ||P_Y - P_{\hat{Y}}||_2 + ||Y^T\hat{Y}||_2.$$  

Spectral Clustering via Random Projections. The most relevant algorithmic result to ours is [37]. The authors suggest to reduce the dimension of the data points with random projections before forming the similarity matrix $W$. Recall that we apply random projections to a scaled version of $W$. No theoretical results are reported for this approach.

Power Iteration Clustering. Another clustering method similar to ours is in [26]. The authors use the power iteration, as we do, on $W$ but choose $S$ 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 $\rho$ as the number of incorrectly classified data points divided by the total number of points. Suppose $L = L + E$, where $L$ and $\tilde{L}$ 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 by looking at the signs of the computed eigenvector. Denote by $v_1$ and $v_2$, the $i$th (smallest) eigenvectors of $L$ and $L$, respectively. Then, [48, 21, 44] show the following bound: $\rho \leq ||v_2 - v_1||_2 \leq (\lambda_2 - \lambda_1)^{-1} ||E||_2 + O(||E||_2)$. 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 this with the “multiplicative gap”.

The Spielman-Teng iterative algorithm. Motivated by a result of Mihail [29], Spielman and Teng [43] suggest the following definition for an approximate Fiedler vector: For a Laplacian matrix $L$ and $0 < \varepsilon < 1$, $v \in \mathbb{R}^m$ is an $\varepsilon$-approximate Fiedler vector if $v$ is orthogonal to the all-ones vector and $\frac{\lambda v^T v}{v^T v} \leq (1 + \varepsilon) \frac{L_{\mathcal{F}}}{\mathcal{F}} = (1 + \varepsilon)\lambda_{n-1}(L)$. Here $f \in \mathbb{R}^m$ denotes the Fiedler vector (the eigenvector of $L$ corresponding to the second smallest eigenvalue). Theorem 6.2 in [43] gives a randomized algorithm to compute such an $\varepsilon$-approximate Fiedler vector w.p. $1 - \delta$, in time $O\left(\min\{n, m\} \ln^2 \left(\min\{n, m\}\right) \ln \left(\frac{1}{\delta}\right) \ln \left(\frac{1}{\varepsilon}\right)\right)$. 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. [48] 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 [35, 5, 46]. 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 $W$, i.e., use a submatrix of $W$ and hope that it approximates the entire matrix $W$ well. This is known as the Nyström method [31, 3, 17]. The Nyström method relies on a submatrix to approximate the entire similarity matrix $W$. 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 [40]. 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 the suggested algorithm, $\rho \lesssim (\lambda_2(L) - \lambda_3(L))^{-1} \left(n^{-2} + n^{-1}b^{-1}\right)$, where $L$ is the graph Laplacian matrix.
Randomized Numerical Linear Algebra (RandNLA). 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 [36, 6, 15, 10]; (ii) Linear equations with Symmetric Diagonally Dominant matrices [25] (iii) Low-rank approximation of matrices [28, 19, 7, 10]; (v) K-means clustering [8]; (vii) support vector machines [34]. (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 with random projections. However, to prove the theoretical behavior of the algorithm we derive results that could be of independent interest. The main novel technical ingredient is that we show that the difference between \( Y \) and an orthonormal transformation of \( \tilde{Y} \) is bounded (see Theorem 3). Previous work analyzes the approximation error after projecting \( W \) on \( P_Y \). Specifically, [20] argues that \( \| W - P_Y \tilde{W} \|_2 \) is small.

Subspace Iteration. Finally, there are deep connections of our approach with a classical algorithm in the numerical linear algebra literature, known as “subspace iteration”. See for example Section 8.2.4 in [18]. Theorem 8.2.2 in [18] shows a bound similar to our bound in Theorem 10. We leave it as a topic of future work a detailed discussion of those connections.

5 Preliminaries

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

\[
A = \begin{pmatrix} U_k & U_{\rho-k} \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{\rho-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ V_{\rho-k}^T \end{pmatrix},
\]

with singular values \( \sigma_1(A) \geq \ldots \geq \sigma_k(A) \geq \sigma_{k+1}(A) \geq \ldots \geq \sigma_{\rho}(A) > 0 \). The matrices \( U_k \in \mathbb{R}^{m \times k} \) and \( U_{\rho-k} \in \mathbb{R}^{m \times (\rho-k)} \) contain the left singular vectors of \( A \); and, similarly, the matrices \( V_k \in \mathbb{R}^{n \times k} \) and \( V_{\rho-k} \in \mathbb{R}^{n \times (\rho-k)} \) contain the right singular vectors of \( A \). Also, \( A^+ = V_A \Sigma_A^{-1} U_A^T \) denotes the Moore-Penrose pseudo-inverse of \( A \). For a symmetric positive definite matrix (SPSD) \( A = B B^T \), \( \lambda_i(A) = \sigma_i^2(B) \) denotes the \( i \)-th eigenvalue of \( A \).

Given a matrix \( A \) with \( m \) rows, \( P_A \in \mathbb{R}^{m \times m} \) is the projector matrix that projects in the column span of \( A \), i.e., for any matrix \( Z \) that is a basis for \( \text{span}(A) \), \( P_A = ZZ^+ \). For example, \( P_A = U_A U_A^T \). For \( A, B \),

\[
\| P_A - P_B \|_2 = \| (I - P_A) P_B \|_2 = \| (I - P_B) P_A \|_2,
\]

5.1 Random Matrix Theory

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

Lemma 5 (The norm of a random Gaussian Matrix [11]). Let \( A \in \mathbb{R}^{n \times m} \) be a matrix with i.i.d. standard Gaussian random variables, where \( n \geq m \). Then, for every \( t \geq 4 \),

\[
P\{ \sigma_1(A) \geq tn^{1/2} \} \geq e^{-nt^2/8}.
\]

Lemma 6 (Invertibility of a random Gaussian Matrix [38]). Let \( A \in \mathbb{R}^{n \times n} \) be a matrix with i.i.d. standard Gaussian random variables. Then, for any \( \delta > 0 \)

\[
P\{ \sigma_n(A) \leq \delta n^{-1/2} \} \leq 2.35\delta.
\]

6
6 Proof of Theorem 3

First, we show that there is some orthonormal matrix $Q$ such that $\|Y - \tilde{Y}Q\|_2^2 \leq \|P_Y - P_{\tilde{Y}}\|_2^2$.

**Theorem 7.** Let $Y \in \mathbb{R}^{n \times k}$ and $\tilde{Y} \in \mathbb{R}^{n \times k}$ are the matrices described in the exact and the approximate spectral clustering algorithms, respectively. Then, there is an orthonormal matrix $Q \in \mathbb{R}^{k \times k}$ such that,

$$\|Y - \tilde{Y}Q\|_2^2 \leq \|P_Y - P_{\tilde{Y}}\|_2^2$$

**Proof.** Observe that if $O$ is a square orthonormal matrix, then $O^T$, $YO^T$, and $\tilde{Y}O$ are all orthonormal matrices. Recall also that $\|A\|_2 = \|AO\|_2$ when $O$ is orthonormal. Consequently,

$$\inf_{OO^T = I} \|Y - YO\|_2 \leq \inf_{OO^T = I} \|Y - \tilde{Y}(\tilde{Y}^TO)\|_2$$

$$= \inf_{OO^T = I} \|YO^T - \tilde{Y}^T\|_2$$

$$\leq \inf_{OO^T = I} \|YY^T O - \tilde{Y}^T\|_2$$

$$\leq \|YY^T - \tilde{Y}^T\|_2$$

$$= \|P_Y - P_{\tilde{Y}}\|_2.$$  

It follows that there is an orthonormal matrix $Q$ satisfying the claim of the theorem. 

To prove Theorem 3 it only remains to show that $\|P_Y - P_{\tilde{Y}}\|_2^2 \leq O(\varepsilon^2)$. We show this bound in Corollary 11 (specifically, apply this corollary to $A = \tilde{W}$). Corollary 11 is based on few results of independent interest, which we present next.

7 Error bounds for Subspace Iteration

We start with a matrix perturbation bound which we use later to prove our main results for approximate spectral clustering.

**Lemma 8.** Consider three matrices $D, C, E \in \mathbb{R}^{m \times n}$; Let $D = C + E$ and $PC\Sigma = 0_{m \times n}$. Then,

$$\|(I - PD)PC\|_2^2 = 1 - \lambda_{\text{rank}(C)}(C(D^*D)^\dagger C^T).$$

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

$$\|(I - PD)PC\|_2^2 = \|PC(I - PD)PC\|_2$$

$$= \|UU^T - UU^TVV^TU^T\|_2$$

$$= \|U(I - U^TVV^TU)U^T\|_2$$

$$= \|I - U^TVV^TU\|_2.$$  

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

It follows that

$$\|(I - PD)PC\|_2^2 = 1 - \lambda_k(U^TVV^TU) = 1 - \lambda_k(V^TUU^T) = 1 - \lambda_k(UU^TVV^T),$$

where our manipulations are justified by the fact that, when the two products are well-defined, the eigenvalues of $AB$ are identical to those of $BA$ up to multiplicity of the zero eigenvalues. We further observe that

$$\lambda_k(UU^TVV^T) = \lambda_k(PCPD) = \lambda_k(PCDD^\dagger) = \lambda_k(PC(C + E)D^\dagger) = \lambda_k(CD^\dagger).$$
Lemma 9. Let \( \Omega_1 = (\mathbf{A} \mathbf{A}^T)^p \mathbf{A} \mathbf{S} \) and \( \Omega_2 = \mathbf{A}_k \), we obtain
\[
\|\mathbf{P}_{\Omega_1} - \mathbf{P}_{\Omega_2}\|_2^2 = 1 - \lambda_k (\mathbf{A}_k \mathbf{S} (\mathbf{S}^T \mathbf{A} \mathbf{S})^T)^{\dagger} \mathbf{S}^T \mathbf{A}_k \mathbf{S}^{\dagger} \mathbf{S}^T \mathbf{A}_k^T).
\]

Proof. In Lemma 8, take \( \mathbf{C} = \mathbf{A}_k \mathbf{S} \) and \( \mathbf{E} = \mathbf{A}_{p-k} \mathbf{S} \) so that \( \mathbf{D} = \mathbf{A} \mathbf{S} = \mathbf{C} + \mathbf{E} \). Indeed, since \( \text{rank}(\mathbf{A}_k \mathbf{S}) = k = \text{rank}(\mathbf{A}_k) \), the range spaces of \( \mathbf{A}_k \mathbf{S} \) and \( \mathbf{A}_k \) are identical, so
\[
\mathbf{P}_C \mathbf{E} = \mathbf{P}_{A_k} \mathbf{A}_{p-k} \mathbf{S} = 0_{m \times n}.
\]

Lemma 8 gives
\[
\|\mathbf{(I-P_D)} \mathbf{P}_C\|_2^2 = 1 - \lambda_k (\mathbf{A}_k \mathbf{S} (\mathbf{S}^T \mathbf{A} \mathbf{S})^T)^{\dagger} \mathbf{S}^T \mathbf{A}_k \mathbf{S}^{\dagger} \mathbf{S}^T \mathbf{A}_k^T).
\]

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

Theorem 10. Given \( \mathbf{A} \in \mathbb{R}^{m \times n} \), let \( \mathbf{S} \in \mathbb{R}^{n \times k} \) be such that \( \text{rank}(\mathbf{A}_k \mathbf{S}) = k \). Let \( p \geq 0 \) be an integer and define
\[
\gamma_p = \|\sum_{t=k}^{2p+1} \mathbf{V}_{p-k} \mathbf{S} (\mathbf{V}^T_k \mathbf{S})^{-1} \Sigma_k^{-2p-1}\|_2.
\]

Then, for \( \Omega_1 = (\mathbf{A} \mathbf{A}^T)^p \mathbf{A} \mathbf{S} \) and \( \Omega_2 = \mathbf{A}_k \), we obtain
\[
\|\mathbf{P}_{\Omega_1} - \mathbf{P}_{\Omega_2}\|_2^2 = \frac{\gamma_p^2}{1 + \gamma_p^2}.
\]

Proof. Note that \( \mathbf{A}_k \mathbf{S} \) has full column rank, so
\[
\mathbf{S}^T \mathbf{A}^T_k \mathbf{A}_k \mathbf{S} > 0
\]
is invertible. It follows that
\[
\mathbf{S}^T \mathbf{A}^T \mathbf{A} \mathbf{S} = \mathbf{S}^T \mathbf{A}^T_k \mathbf{A}_k \mathbf{S} + \mathbf{S}^T \mathbf{A}^T_{p-k} \mathbf{A}_{p-k} \mathbf{S} > 0
\]
is also invertible. Also \( \mathbf{V}^T_k \mathbf{S} \) 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.
Hence,

\[
\lambda_k(A_k S(SA^T AS)^t S^T A_k^T) = \\
= \lambda_k \left( U_k \Sigma_k V_k^T S(SA^T AS)^{-1} S^T \Sigma_k U_k^T \right) \\
= \lambda_k \left( \Sigma_k V_k^T S(SA^T AS)^{-1} S^T \Sigma_k \right) \\
= \lambda_{\max} \left( \left( \Sigma_k V_k^T S(SA^T AS)^{-1} S^T \Sigma_k \right)^{-1} \right) \\
= \lambda_{\max} \left( \left( S^T V_k \Sigma_k \right)^{-1} \left( SA^T AS \right) \left( \Sigma_k V_k^T S \right)^{-1} \right) \\
= \lambda_{\max} \left( I_k + \left( S^T V_k \Sigma_k \right)^{-1} \left( S A^T A S \right) \left( \Sigma_k V_k^T S \right)^{-1} \right) \\
= \left( 1 + \left\| \Sigma_{\rho-k} V_{\rho-k} S(\Sigma_k V_k^T S)^{-1} \right\|^2_2 \right)^{-1} \\
= \left( 1 + \left\| \Sigma_{\rho-k} V_{\rho-k} S(\Sigma_k V_k^T S)^{-1} \Sigma_k^{-1} \right\|^2_2 \right)^{-1} \\
= (1 + \gamma_p^2)^{-1}.
\]

The result follows by substituting this expression into the expression given in Lemma 9. \( \blacksquare \)

The term \( \gamma_p \) has a compelling interpretation: note that

\[
\gamma_p \leq \left( \frac{\sigma_{k+1}(A)}{\sigma_k(A)} \right)^{2p+1} \left\| V_{\rho-k}^T S \right\|_2 \left( V_k^T S \right)^{-1}_2;
\]

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

To apply Theorem 10 practically to estimate the dominant \( k \)-dimensional left singular space of \( A \), we must choose an \( S \in \mathbb{R}^{n \times k} \) that satisfies rank(\( A_k S \)) = \( k \) and bound the resulting \( \gamma_p \). The next corollary gives a bound on \( \gamma_p \) when \( S \) is chosen to be a matrix of i.i.d. standard Gaussian r.v.s.

**Corollary 11.** Fix \( A \in \mathbb{R}^{m \times n} \) with rank at least \( k \). Let \( p \geq 0 \) be an integer and draw \( S \in \mathbb{R}^{n \times k} \), a matrix of i.i.d. standard Gaussian r.v.s. Fix \( \delta \in (0, 1) \) and \( \varepsilon \in (0, 1) \). Let

\[
\Omega_1 = (AA^T)^p AS,
\]

and

\[
\Omega_2 = A_k.
\]

If \( p \) satisfies

\[
p \geq \frac{1}{2} \ln(4\varepsilon^{-1} \delta^{-1} \sqrt{k(n-k)}) \ln^{-1} \left( \frac{\sigma_k(A)}{\sigma_{k+1}(A)} \right)
\]

then with probability at least \( 1 - \varepsilon^{-n} - 2.35\delta \),

\[
\| P_{\Omega_1} - P_{\Omega_2} \|_2^2 \leq \frac{\varepsilon^2}{1 + \varepsilon^2} = O(\varepsilon^2).
\]
| 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 [9] that were used in our experiments.

Proof. Clearly $A_k S$ 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), so Theorem 10 is applicable. Estimate $\gamma_p$ as

$$\gamma_p \leq \left\| \Sigma_{p-k}^{2p+1} \right\|_2 \left\| V_{p-k}^T S \right\|_2 \left\| (V_k^T S)^{-1} \right\|_2 \left\| \Sigma_k^{-2p+1} \right\|_2$$

$$= \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \frac{\sigma_{\max}(V_{p-k}^T S)}{\sigma_{\min}(V_k^T S)}$$

$$\leq \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \frac{4\sqrt{n-k}}{\delta/\sqrt{k}}$$

$$= \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \cdot 4\delta^{-1} \sqrt{k(n-k)}.$$ 

In the above, the last inequality follows from the fact that $V_{p-k}^T S$ and $V_k^T S$ 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

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

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

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

In this case,

$$\left\| P_Y - P_{\tilde{Y}} \right\|_2^2 \leq \frac{\varepsilon^2}{1 + \varepsilon^2} = O(\varepsilon^2).$$

8 Experiments

Setting expectations. 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 — $(\hat{W} W^T) p WS \in \mathbb{R}^{n \times k}$ — as opposed to $W \in \mathbb{R}^{n \times n}$; 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 $W$, which is out of the scope of this paper.
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 (\(\hat{W}W^T\))^p WS 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 W, which we did not optimize.

| Name    | Exact | Approximate | Approximate |
|---------|-------|-------------|-------------|
|         | CR%   | time (secs) | 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 |

Setup. To conduct our experiments, we developed MATLAB versions of the exact and approximate algorithms (Section 3 and 2.1). To compute W, we use the heat kernel: \(\ln(W_{ij}) = \frac{|x_i - x_j|^2}{\sigma_{ij}}\), where \(x_i \in \mathbb{R}^d\) and \(x_j \in \mathbb{R}^d\) are the data points and \(\sigma_{ij}\) is a tuning parameter; \(\sigma_{ij}\) is determined using the self-tuning method described in [49]. For the partial singular value decomposition of \(W\), we used IRLBA [2] due to its excellent support for sparse decompositions. To compute WS, we used MATLAB’s built-in normrnd function to generate a Gaussian S. For the complete singular value decomposition of \((\hat{W}W^T)^p WS\), 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 [23]. 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 \(CR = \frac{1}{n} \sum_{i=1}^{n} I_i\), where \(I_i\) 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.

Results. To determine the quality of the clustering obtained by our approximate 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 S. In each iteration, we varied p from [0,10]. First, we explore how the error in Theorem 3 behaves while p grows. As we are not aware how to find this optimal \(Q\), we compute \(\hat{Q}\) as the solution of the Frobenius norm version of the problem, which is the famous Procrustes problem, \(\hat{Q} = \text{argmin}_{Q^TQ = I_k} \| Y - Y\hat{Q} \|_F\), i.e., \(\hat{Q} = U_BV_B^T\) (\(B = Y^TV\)Y). Figure 3 confirms that the approximation 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 $W$ 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 $W$ is merely $WS$. As expected, as $p$ increases, we see a linear increase in the time taken.

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 $Y$ and $\tilde{Y}$ are clustered identically as $\varepsilon \to 0$. However, in our setting $\varepsilon$ is not approaching 0 (see Figure 3), hence $k$-means finds (slightly) different clusters for small $\varepsilon$, e.g., $\varepsilon = 0.1$. 

Figure 3: Reduction in the error $||Y - \tilde{Y}Q||_2$ (see Theorem 3 and text below) with increase in $p$. 

Figure 4: Procrustes error with $p$ for different datasets.
Figure 4: Variation of time (normalized to $p=0$) to complete the power iterations and the eigen-decomposition of $\hat{W}\hat{W}^T p\hat{W}S$. 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.

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.
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, A. Zouzias, and P. Drineas. Random projections for k-means clustering. In In NIPS, 2010.

[9] 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.

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

[11] 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.

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

[13] 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.

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

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

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

[17] 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.

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

[19] 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.

[20] 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.

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

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

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

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

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

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

[27] 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.
[28] P. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 2010.

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

[30] 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.

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

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

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

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

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

[36] 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.

[37] 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.

[38] 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.

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

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

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

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

[43] 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.

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

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

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

[47] 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.

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

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