11-14-2012

Spectral Clustering: An Empirical Study of Approximation Algorithms and its Application to the Attrition Problem

B. Cung
University of California, Los Angeles

T Jin
Stanford University

J. Ramirez
Instituto Tecnológico Autónomo de México

A. Thompson
University of Nebraska-Lincoln

C. Boutsidis
IBM T.J. Watson Research Center

See next page for additional authors

Recommended Citation

Cung, B., Jin, T., Ramirez, J., Thompson, A., Boutsidis, C., Needell, D., "Spectral Clustering: An empirical study of Approximation Algorithms and its Application to the Attrition Problem", SIAM Undergraduate Research Journal, vol. 5, pp. 283-303, 2012.

Authors
B. Cung, T Jin, J. Ramirez, A. Thompson, C. Boutsidis, and Deanna Needell

SPECTRAL CLUSTERING: AN EMPIRICAL STUDY OF APPROXIMATION ALGORITHMS AND ITS APPLICATION TO THE ATTRITION PROBLEM

B. CUNG, T. JIN, J. RAMIREZ, A. THOMPSON

ADVISORS: C. BOUTSIDIS, AND D. NEEDELL

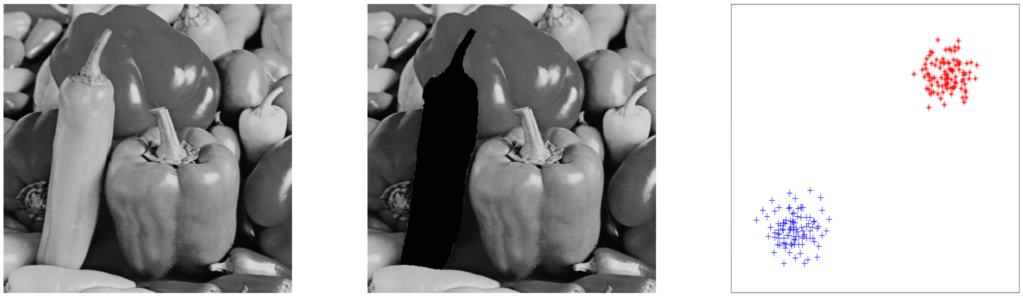
ABSTRACT. Clustering is the problem of separating a set of objects into groups (called clusters) so that objects within the same cluster are more similar to each other than to those in different clusters. Spectral clustering is a now well-known method for clustering which utilizes the spectrum of the data similarity matrix to perform this separation. Since the method relies on solving an eigenvector problem, it is computationally expensive for large datasets. To overcome this constraint, approximation methods have been developed which aim to reduce running time while maintaining accurate classification. In this article, we summarize and experimentally evaluate several approximation methods for spectral clustering. From an applications standpoint, we employ spectral clustering to solve the so-called attrition problem, where one aims to identify from a set of employees those who are likely to voluntarily leave the company from those who are not. Our study sheds light on the empirical performance of existing approximate spectral clustering methods and shows the applicability of these methods in an important business optimization related problem.

1. INTRODUCTION

Clustering or *cluster analysis* addresses the problem of separating a set of objects into *clusters* so that objects within each cluster are more similar to each other than to objects in different clusters. The clustering problem has become ubiquitous in data mining and machine learning with applications ranging from image processing to bioinformatics. What one means by clustering, and the type of clustering desired is application dependent. For example, one may wish to segment an image such as that in Figure 1 (a)-(b). In medical imaging, segmentation may aid in the identification of tumors, assist physicians in surgery and separate anatomical structures. Computer vision applications utilize clustering methods to identify foreign objects in surveillance images or detect road signs for computer piloted vehicles. In statistical analysis, the objects to be clustered may represent individuals in a population viewed as a vector of personal attributes. For example, we will consider the *attrition problem*: from a dataset of employees one wishes to identify which cluster of employees are likely to voluntarily leave the company and which are not. With this problem as our overarching focus, we will consider here and throughout the case in which we wish to identify *two* clusters in the data. One can visualize this type of clustering in low dimensions, for example as seen in Figure 1 (c), where the “correct” cluster identification is obvious.

1.1. **Contributions.** As we discuss later in this section, there are different clustering algorithms such as k-means or spectral clustering. The focus of this article is on *spectral clustering*, a method which utilizes an eigenvector from the so-called data similarity matrix. Computing eigenvectors of such matrices could be potentially a very expensive operation. Thus, faster approximation algorithms for spectral clustering have appeared in the literature. The first contribution of this article is to summarize and experimentally evaluate such approximation algorithms. Our second contribution is to apply spectral clustering to a modern business optimization related problem which we call the *attrition problem*: given a set of

Research performed at the Institute for Pure and Applied Mathematics (IPAM) in the University of California, Los Angeles (UCLA) during the Research in Industrial Projects for Students (RIPS) summer program of 2012. B. Cung (bcung@ucla.edu) is with the University of California, Los Angeles. T. Jin (tonyjin1@stanford.edu) is with Stanford University. J. Ramirez (juan.ramirez.prado@itam.mx) is with the Instituto Tecnológico Autónomo de México, and A. Thompson (aubrey@huskers.unl.edu) is with the University of Nebraska-Lincoln. C. Boutsidis (cboutsi@us.ibm.com) is with the IBM T.J. Watson Research Center and D. Needell (dneedell@cmc.edu) is with Claremont McKenna College.



(a) (b) (c)

Figure 1 Examples of clustering: (a) original peppers image, (b) segmentation of peppers image, (c) two clusters in two dimensions.

employees, we would like to separate those who are likely to voluntarily resign from the company from those who are not. Such information could be of tremendous value to the company because of the high costs to replace the workforce. We present the empirical study of approximation algorithms for spectral clustering in Section 2 and the case study to the attrition problem in Section 4.

1.2. **Clustering via k-means.** The goal of clustering methods is to identify clusters automatically from the data input. The k*-means clustering* method is an approach that separates objects into k clusters so that each object is assigned to the cluster whose *mean* is nearest in the Euclidean sense [8, 19]. That is, given n vectors $\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}$ in d-dimensional space, $\boldsymbol{x_j} \in \mathbb{R}^d$, the k-means method aims to minimize the sum of the squared intra-cluster distances:

$$\sum_{i=1}^{k} \sum_{\boldsymbol{x_j} \in S_i} \|\boldsymbol{x_j} - \boldsymbol{\mu_i}\|_2^2,$$

where, for $i = 1, \dots, k$, S_i contains the indices of vectors in the ith cluster, and $\boldsymbol{\mu_i} \in \mathbb{R}^d$ denotes the mean (center) of vectors in that cluster.

Although this problem is in general NP-Hard [10], efficient iterative algorithms have been developed that often converge to a locally optimal solution (see e.g. Chapter 20 of [9]). Although variations in the method exist, the standard approach due to Lloyd (for $k = 2$ clusters) consists of repeating the two steps described in Algorithm 1. We denote by S^c the complement of the set S.

Algorithm 1 k-means Clustering Method (for $k = 2$)

```
procedure ($\boldsymbol{x_j}$'s, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, $T$)    ▷ data points $\boldsymbol{x_j} \in \mathbb{R}^d$, initial means $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, number of iterations $T$
    for $t = 1, 2, \dots, T$ do
        Cluster by assigning each object to its closest mean:
            $S_1 = \{\boldsymbol{x_j} : \|\boldsymbol{x_j} - \boldsymbol{\mu_1}\|_2 \le \|\boldsymbol{x_j} - \boldsymbol{\mu_2}\|_2\}, \quad S_2 = S_1^c$
        Update the mean vectors:
            $\boldsymbol{\mu_1} = \frac{1}{|S_1|} \sum_{\boldsymbol{x_j} \in S_1} \boldsymbol{x_j}, \quad \boldsymbol{\mu_2} = \frac{1}{|S_2|} \sum_{\boldsymbol{x_j} \in S_2} \boldsymbol{x_j}$
    end for
end procedure
```

To separate the points into more than 2 clusters, one extends the method for k-means in the natural way. The runtime of the method is $O(knT)$. When the mean of each cluster converges toward the true cluster center, the k-means method performs well. This is the case, for example, when the clusters are each of similar size and have a spherical shape as seen in Figure 2 (a). However, when the clusters are not linearly separable, as in Figure 2 (b), k-means may often incorrectly assign points to clusters. Although k-means performs well in many settings, there are also applications where these limitations are apparent, and this leads us to search for other methods that will work for more general purposes.

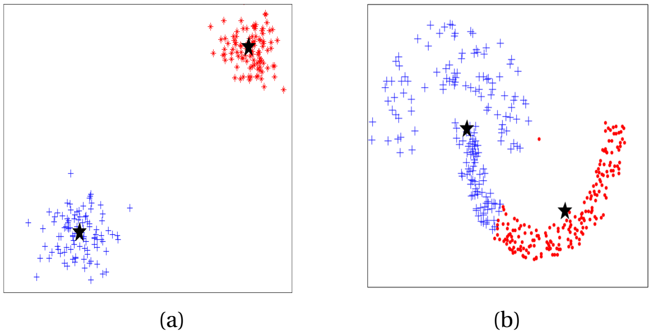


Figure 2 The k-means clustering method: (a) two clusters in two dimensions with cluster means converged to cluster centers (marked with stars); (b) non-spherical clusters are difficult to identify via k-means clustering.

1.3. **Spectral Clustering.** An alternative way to approach the clustering problem is to view the data points as a graph. Each vertex of the graph will represent a data point, and each edge will represent the *similarity* between the two corresponding vertices. To that end, for n data points $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$ in d-dimensional space, denote by $\boldsymbol{X}$ the $n \times d$ data matrix whose rows contain the data vectors $\boldsymbol{x_j^T}$. We construct a *similarity matrix* $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ whose (i,j)th entry gives the similarity between the two corresponding data points:

$$\boldsymbol{W}_{ij} = \exp\left(-\frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{\sigma_{ij}^2}\right), \tag{1}$$

where σ_{ij} is a tuning parameter to be chosen later. The similarity matrix $\boldsymbol{W}$ induces a complete graph $(V, E, \boldsymbol{W})$ where V is the set of vertices (objects) to be clustered, E is the set of edges, and $\boldsymbol{W}$ represents the weights of the edges. The clustering problem can then be viewed as the partitioning of the graph into sets of vertices such that the edges within the sets have large weights, and the edges across sets have small weights. Formally, in the 2-clustering setting, we wish to identify sets A and B which minimize the so-called *normalized cut* objective,

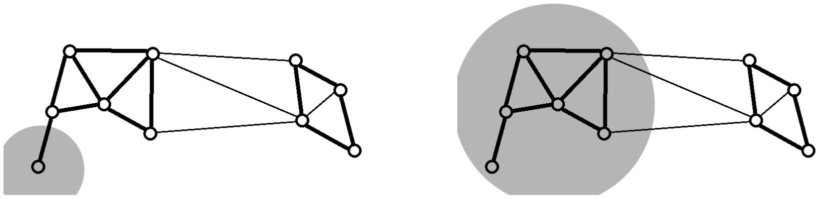
$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,V)} + \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,V)},$$

where the cut and association functions are defined by

$$\mathrm{cut}(A,B) = \sum_{\substack{\boldsymbol{x_i}\in A\\ \boldsymbol{x_j}\in B}} \boldsymbol{W}_{ij}, \qquad \mathrm{assoc}(A,V) = \sum_{\substack{\boldsymbol{x_i}\in A\\ \boldsymbol{x_j}\in V}} \boldsymbol{W}_{ij}, \quad \text{and} \quad \mathrm{assoc}(B,V) = \sum_{\substack{\boldsymbol{x_i}\in B\\ \boldsymbol{x_j}\in V}} \boldsymbol{W}_{ij}.$$

The numerators of Ncut defined in this way guarantee that the weights between the clusters A and B are small. On the other hand, if we simply minimized the cut function, one might obtain cuts for which A is a very small set of vertices (perhaps even just one vertex) and B is the remaining vertices, as shown in Figure 3 (a). To avoid these trivial cuts, we divide by the association function, which sums the weights between a set of vertices and *all* nodes. If a set of vertices in the partition is too small, its association will be small, leading to a large Ncut. With this normalization, one hopes to avoid this type of bias and obtain cuts as in Figure 3 (b).

Minimizing the normalized cut is NP-Complete in general (see [15] for the proof; originally due to Papadimitriou). However, recently a relaxation of the optimization problem has been reduced to an



(a)

(b)

Figure 3 Two examples of graph partitioning (one set is shown shaded and the other unshaded): (a) Minimizing the cut of the graph, (b) minimizing the normalized cut of the graph.

eigenvector problem [15]. Given the $n \times n$ similarity matrix $\boldsymbol{W}$, one defines the normalized Laplacian[1] matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ by

$$\boldsymbol{L} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{D}^{-1/2}, \tag{2}$$

where $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is the diagonal matrix of degree nodes,

$$\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}. \tag{3}$$

Shi and Malik argued that the eigenvector corresponding to the second smallest eigenvalue of $\boldsymbol{L}$ corresponds to a linear transformation of the relaxed solution to the Ncut problem [15]. Indeed, the clustering is then performed by selecting an appropriate threshold, and assigning indices of the eigenvector with large values to one cluster, and indices with small values to the other. For example, with the eigenvector plotted in Figure 4, the first 400 data points would be assigned to one cluster and the second 400 to the other. This gives rise to the following formal definition of the spectral clustering algorithm.

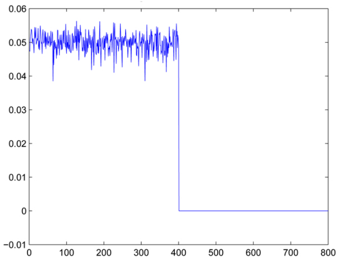


Figure 4 An example of an eigenvector obtained from the spectral clustering method (horizontal axis represents the index, vertical the value of the eigenvector at that index). Here we assign the first 400 objects to one cluster and the second 400 to the other.

Algorithm 2 Spectral Clustering Method (for two clusters)

```
procedure ($\boldsymbol{X}, \sigma$)        ▷ $n \times d$ data matrix $\boldsymbol{X}$, tuning parameter $\sigma$
    Construct the similarity matrix $\boldsymbol{W}$ in (1), degree matrix $\boldsymbol{D}$ in (3) and Laplacian $\boldsymbol{L}$ in (2)
    Compute the eigenvector corresponding to the second smallest eigenvalue of $\boldsymbol{L}$
    Assign the indices in the eigenvector with large values to one cluster, the rest to the other
end procedure
```

The step which is most computationally burdensome is the eigenvector computation. In general this step yields an $\mathrm{O}(n^3)$ running time. This cost is often detrimental for large applications and is one of the biggest drawbacks to spectral clustering methods.

[1] Note that one may also consider the Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.

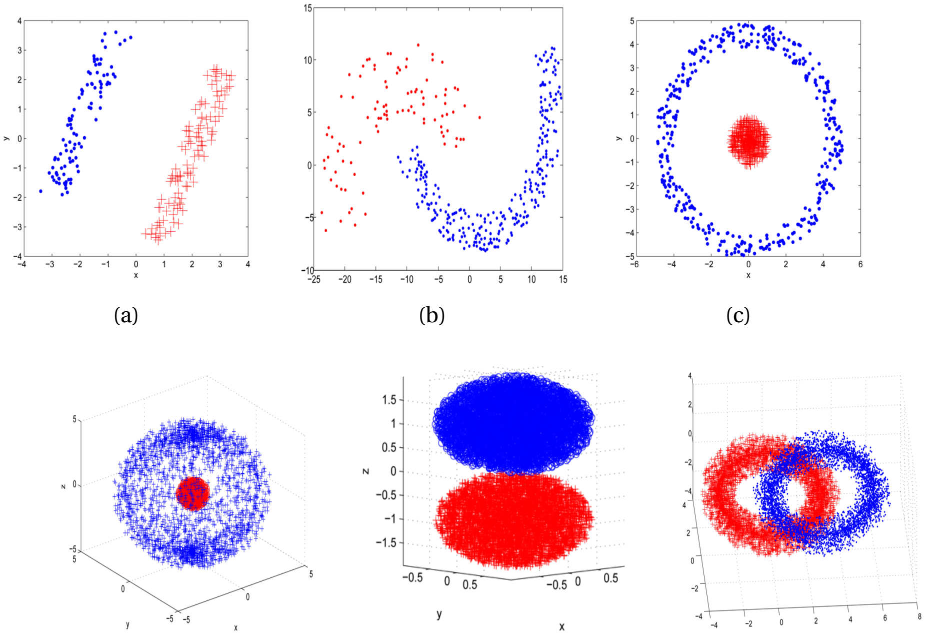
Experimental results using the spectral clustering method are shown in Figure 5. Here we use the self-tuning approach by Zelnik-Manor and Perona [13] for obtaining the similarity matrix. Consider the vector in $\mathbb{R}^n$ defined entrywise by

$$v_i = \|\boldsymbol{x_i} - \boldsymbol{x_{i_K}}\|_2, \tag{4}$$

where $\boldsymbol{x_{i_K}}$ denotes the K^{th} closest neighbor to $\boldsymbol{x_i}$. We then set the scaling parameter σ_{ij} as

$$\sigma_{ij} = v_i v_j. \tag{5}$$

As in the article, we set $K = 7$ for all our experiments with the spectral clustering algorithm. The experiment performed on the interlocked rings, interlocked half rings, and Gaussian strips were run on Intel Core 2 Duo E8500 3.16 GHz machines with 6 MB cache and 16 GB memory. The concentric spheres and concentric rings experiments were run on Intel Core i7 870 2.93 GHz machines with 4 cores, 8 MB cache and 16 GB memory. The tangent spheres experiments were run on an Intel Xeon W3520 2.67 GHz machine with 4 cores, 8 MB cache and 16 GB memory.



(a) (b) (c)

(d) (e) (f)

Figure 5 Accurate clustering results for data sets of many shapes and sizes using spectral clustering: (a) Gaussian strips, (b) Interlocked half rings, (c) Concentric rings, (d) Concentric spheres, (e) Tangent spheres, and (f) Interlocked rings.

The running times and dataset sizes are summarized in Table 1.

Dataset	Size n	Running time (s)
Gaussian strips	200	0.17
Interlocked half rings	373	0.73
Concentric rings	800	8.02
Concentric spheres	5000	855.12
Tangent spheres	10000	6365.4
Interlocked rings	10000	6870.7

Table 1 Running times and dataset sizes for problems of Figure 5.

Although the spectral clustering method performs accurate cluster identification even for datasets challenging to the k-means method, its cubic runtime is a practical obstacle in many applications. For this reason, approximation methods have been developed which efficiently approximate the spectrum of the Laplacian matrix $\boldsymbol{L}$. In this article, we consider four popular approximation methods. The *Fast spectral clustering* method [20] and Extensible spectral clustering method [18] both identify a small set of representative points from the dataset, perform spectral clustering on this much smaller set of points, and extend the identification to the remaining data points. An alternative way to reduce the dimension of the similarity matrix is to randomly sample values from the matrix to obtain a smaller submatrix, which is the basis of the *Spectral clustering on a budget* [14] and Nyström methods [5].

1.4. **Organization.** The remainder of the article is organized as follows. The four approximation methods are described and discussed in Section 2. Section 3 displays numerical results used for a comparison between the methods. In Section 4 we focus on the attrition problem and analyze how each method performs at that task. We conclude in Section 5 with a discussion of the findings.

2. Approximation Methods

The fundamental idea behind efficiently approximating the spectral clustering method is to reduce the problem size to be clustered. To maintain accurate cluster identification, one hopes that the reduced problem preserves the same cluster structure as the original problem. The two main approaches to this goal that we discuss here rely either on randomness to reduce the dimension, or some preprocessing algorithmic step to ensure that the smaller set is a good representation of the original. To evaluate accuracy of the method one uses the results of spectral clustering as the ground truth, and compares the output of the other methods to that. To evaluate in a general sense, rather than example to example, one may wish to compare the eigenvector computed by the approximation method to that of the spectral clustering method. We discuss these notions and describe the methods in the remainder of this section.

The common theme between approximate spectral clustering methods is that the $n \times n$ similarity matrix $\boldsymbol{W}$ is downsampled so that clustering can be performed efficiently. Such downsampling will of course lead to errors in the computed eigenvectors, and one wishes to quantify the magnitude of such perturbations to validate the accuracy of the approximate method. In a general context, we can view this process as the perturbation of the Laplacian) matrix $\widetilde{\boldsymbol{L}} = \boldsymbol{L} + \boldsymbol{E}$ where $\boldsymbol{E}$ is some error matrix and $\widetilde{\boldsymbol{L}}$ is the perturbed Laplacian matrix. Standard results from linear algebra guarantee the following bound on the perturbation of eigenvectors.

Theorem 2.1 (Eigenvector Perturbations [20, 6, 16])**.** *Suppose $\widetilde{\boldsymbol{L}} = \boldsymbol{L} + \boldsymbol{E}$ and denote by $\widetilde{\boldsymbol{v}}_i$ and $\boldsymbol{v}_i$ the ith eigenvectors of $\widetilde{\boldsymbol{L}}$ and $\boldsymbol{L}$, respectively, corresponding to the ith smallest eigenvalue. Then*

$$\|\widetilde{\boldsymbol{v}_2} - \boldsymbol{v}_2\|_2 \le \frac{1}{\lambda_2 - \lambda_3}\|\boldsymbol{E}\| + \mathrm{O}(\|\boldsymbol{E}\|^2),$$

where λ_i denotes the ith smallest eigenvalue of $\boldsymbol{W}$.

This result shows that the perturbation in the eigenvectors is controlled by the (spectral) norm of the perturbation in the matrix, and the *eigengap* $\lambda_2 - \lambda_3$. This theory can be extended to bound the angles between eigenspaces of the original and perturbed matrices as well as the norm of their projections [7].

In analyzing approximation methods, one wishes to determine the tradeoff between accuracy and efficiency. To quantify theoretically and empirically the performance of an approximation method, we define the mis-clustering rate by

$$\rho = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_i, \tag{6}$$

where $\mathbb{1}_i$ is the indicator function which is equal to 1 if object $\boldsymbol{x}_i$ is clustered correctly and zero otherwise. We assume here that the correct clustering is given by the spectral clustering method, and compare the results of the approximation methods against that standard. One can then bound the mis-clustering rate by the difference in the eigenvectors under certain assumptions.

Theorem 2.2 (Misclustering rate [6, 20])**.** *Suppose $\widetilde{\boldsymbol{L}} = \boldsymbol{L} + \boldsymbol{E}$ and denote by $\widetilde{\boldsymbol{v}_2}$ and $\boldsymbol{v}_2$ the 2nd (smallest) eigenvectors of $\widetilde{\boldsymbol{L}}$ and $\boldsymbol{L}$, respectively. Then when both sets of eigenvectors partition the data into two clusters and the perturbations in the eigenvectors satisfy the componentwise assumptions of* [6]*,*

$$\rho \le \|\widetilde{\boldsymbol{v}_2} - \boldsymbol{v}_2\|_2^2.$$

This result motivates the development of approximation methods which yield small perturbations in the eigenvectors of the downsampled matrix.

2.1. **Fast Spectral Clustering.** The *fast spectral clustering algorithm* by Yan et al. [20] consists of two major parts: data preprocessing and spectral clustering. The goal of the data preprocessing is to construct a smaller, but representative set of points to undergo spectral clustering rather than the original large dataset. Since the k-means method itself identifies k representative points (usually the cluster means), it seems a natural way to identify a representative set of points even if k is larger than the number of clusters. One can then perform spectral clustering efficiently on the representative points, and assign clusters to the entire original dataset by simply choosing the cluster containing the closest representative point. Indeed, the algorithm for fast spectral clustering is described in Algorithm 3 below.

Algorithm 3 Fast Spectral Clustering Method

```
procedure ($\boldsymbol{X}$, $k$, $T$) ▷ $n \times d$ data matrix $\boldsymbol{X}$, number of representative points $k$, number of iterations $T$
    Find $k$ representative points (centroids $\boldsymbol{y_1}, \ldots, \boldsymbol{y_k}$) via $k$-means
    Create a correspondence table that associates each $\boldsymbol{x_i}$ with the nearest cluster centroid $\boldsymbol{y_j}$;
    Run spectral clustering on the data matrix $\boldsymbol{Y}$ of centroids to obtain a clustering assignment
    Use the correspondence table to recover cluster membership for each point $\boldsymbol{x_i}$
end procedure
```

The complexity for the k-means step is $\mathrm{O}(knT)$, and since spectral clustering is only run on the k representative points, that step yields a cost of just $\mathrm{O}(k^3)$. The remaining assignment steps cost at most $\mathrm{O}(n)$, yielding an overall runtime of $\mathrm{O}(knT + k^3)$. This is of course a significant improvement over the cubic $\mathrm{O}(n^3)$ of spectral clustering when k and T are chosen small enough. To quantify precisely this tradeoff between efficiency and accuracy, the perturbation theory of Theorem 2.2 can be utilized. Indeed, results on fast spectral clustering guarantee the following bound on the mis-clustering rate.

Theorem 2.3 (Spectral misclustering rate [20])**.** *Assume that the assumptions of Theorem 2.2 hold. Then the mis-clusterting rate ρ* (6) *for fast spectral clustering satisfies*

$$\rho \lesssim \frac{2}{(\lambda_2 - \lambda_3)^2} \|\boldsymbol{L} - \widetilde{\boldsymbol{L}}\|_F^2,$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm, $\boldsymbol{L}$ and $\widetilde{\boldsymbol{L}}$ denote the Laplacian and perturbed Laplacian, and the symbol $\lesssim$ implies that higher order terms are ignored in the relation.

This result demonstrates that the mis-clustering rate is again controlled by the eigengap and the perturbations in the Laplacian incurred via fast spectral clustering. The latter term can be bounded in special cases, see [20] for details.

2.2. **Extensible Spectral Clustering.** The notion of identifying a small representative sample of the data on which to initially perform spectral clustering can also be generalized. This class of methods are given the name *extensible spectral clustering* (eSPEC) [12, 2, 18]. Here, one performs spectral clustering on the representative sample of the data, and assign each object in the original dataset to cluster based on its m closest neighbors within the representative sample. We again use the similarity matrix (1) to measure "closeness." The general model is described in Algorithm 4.

Algorithm 4 Extensible Spectral Clustering Method

```
procedure ($X$, $m$, $S$)    ▷ $n \times d$ data matrix $X$, neighboring parameter $m$, representative sample $S$
    Run spectral clustering on the representative sample $S$ to obtain a clustering assignment
    For each object $i$ in $S^c$, find its $m$ closest neighbors in $S$
    Assign each object $i$ to the cluster containing the majority of its $m$ closest neighbors
end procedure
```

There are of course many ways one can initially obtain the representative sample. As in the fast spectral clustering method, one can utilize the k-means method to identify a good representative sample S. Indeed, if the centroids found by the k-means method coincide with data points in the set, the extensible spectral clustering method with $m = 1$ is the same as the fast spectral clustering method. Alternatively, the representative sample can be chosen randomly. For example, one can simply sample uniformly at random from the dataset (see e.g. [12, 17, 18] and references therein) or according to some other probability distribution such as one that assigns probabilities proportional to the norms of each column [4]. In the experiments section below, we see that using uniform sampling with just $m = 1$ provides accurate results even for reasonably small sample sizes. In this case the running time of the method is dominated by the size of the sample, $O(|S|^3)$.

2.3. **Nyström Method.** Both the fast spectral clusterting method and extensible spectral clustering reduce the dimension of the clustering problem by subsampling the *objects* in the data. An alternative approach is to subsample the similarity matrix $\boldsymbol{W}$ (1) itself. In this case, one uses a submatrix of $\boldsymbol{W}$ and asks that the submatrix approximates the entire matrix $\boldsymbol{W}$ well. This is the motivation behind the *Nyström method* [11, 1, 5].

To that end, we decompose the $n \times n$ similarity matrix $\boldsymbol{W}$ so that

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W_{11}} & \boldsymbol{W_{21}^T} \\ \boldsymbol{W_{21}} & \boldsymbol{W_{22}} \end{pmatrix}, \tag{7}$$

where $\boldsymbol{W_{11}} \in \mathbb{R}^{m\times m}, \boldsymbol{W_{21}} \in \mathbb{R}^{(n-m)\times m}$, and $\boldsymbol{W_{22}} \in \mathbb{R}^{(n-m)\times(n-m)}$. Choosing $m \ll n$, $\boldsymbol{W_{22}}$ is very large, and this is thus the part we wish to approximate.

To do so, one computes the similarity matrix for only the m sampled data points, represented by $\boldsymbol{W_{11}}$. The relationship between the sampled data points and the rest of the points is given by $\boldsymbol{W_{21}}$. Then only $\boldsymbol{W_{11}}$ and the first m columns of $\boldsymbol{W}$, denoted $\boldsymbol{W_m} = (\boldsymbol{W_{11}} \boldsymbol{W_{21}^T})^T$, are needed to compute the Nyström approximation:

$$\hat{\boldsymbol{W}} = \boldsymbol{W_m} \boldsymbol{W_{11}^{-1}} \boldsymbol{W_m^T}.$$

The eigenvectors of $\hat{\boldsymbol{W}}$ are then used as an approximation to the eigenvectors of $\boldsymbol{W}$. Unfortunately, these approximate eigenvectors are not necessarily orthogonal, a property that is necessary for the spectral clustering problem. However, when $\boldsymbol{W}$ is positive semidefinite, these eigenvectors can be orthogonalized efficiently. First, we construct

$$\boldsymbol{Q} = \boldsymbol{W_{11}} + \boldsymbol{W_m}^{-\frac{1}{2}} \boldsymbol{W_{21}^T} \boldsymbol{W_{21}} \boldsymbol{W_m}^{-\frac{1}{2}}.$$

Then we compute the eigendecomposition of $\boldsymbol{Q}$ to obtain a matrix $\boldsymbol{U}$ whose columns are equal to the eigenvectors of $\boldsymbol{Q}$ and a diagonal matrix $\boldsymbol{\Lambda}$ with diagonal entries equal to its eigenvalues. The orthogonalized approximate eigenvectors of $\hat{\boldsymbol{W}}$ are then computed as the columns of

$$\boldsymbol{V} = \boldsymbol{W_m} \boldsymbol{W_{11}}^{-\frac{1}{2}} \boldsymbol{U} \boldsymbol{\Lambda}^{-\frac{1}{2}},$$

which can be used for clustering. The Nyström method is thus described as follows in Algorithm 5.

For this process to work, however, one requires that the similarity matrix $\boldsymbol{W}$ be positive semidefinite. Therefore, the self-tuning approach given in (5) can no longer be utilized since it will not necessarily

Algorithm 5 Nyström Method

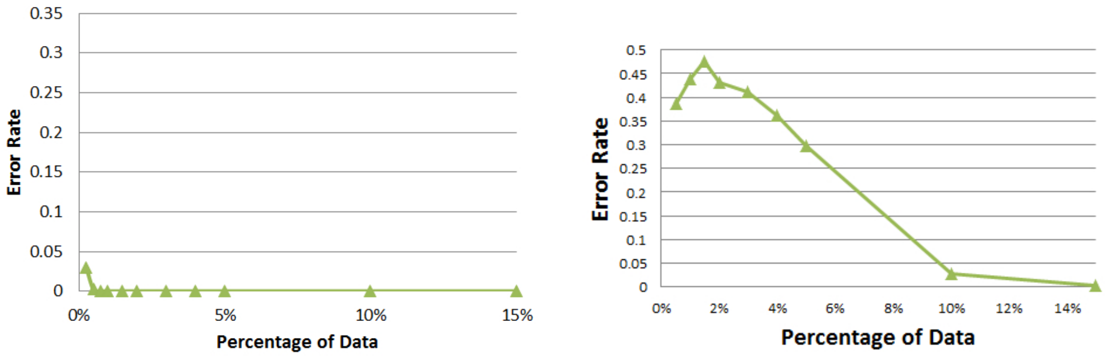
```
procedure ($\boldsymbol{W}$, $m$)    ▷ $n \times n$ similarity matrix $\boldsymbol{W}$, sample size $m$
    Decompose the similarity matrix $\boldsymbol{W}$ as in (7)
    Compute the approximation $\hat{\boldsymbol{W}} = \boldsymbol{W_m} \boldsymbol{W}_{\boldsymbol{11}}^{-1} \boldsymbol{W_m}^T$
    Set $\boldsymbol{Q} = \boldsymbol{W}_{\boldsymbol{11}} + \boldsymbol{W_m}^{-\frac{1}{2}} \boldsymbol{W}_{\boldsymbol{21}}^T \boldsymbol{W}_{\boldsymbol{21}} \boldsymbol{W_m}^{-\frac{1}{2}}$
    Compute the eigendecomposition of $\boldsymbol{Q}$ to obtain eigenvectors $\boldsymbol{U}$ and eigenvalues $\Lambda$
    Compute orthogonalization $\boldsymbol{V} = \boldsymbol{W_m} \boldsymbol{W}_{\boldsymbol{11}}^{-\frac{1}{2}} \boldsymbol{U} \boldsymbol{\Lambda}^{-\frac{1}{2}}$
    Use the columns of $\boldsymbol{V}$ as approximate eigenvectors for spectral clustering
end procedure
```

guarantee positive semidefiniteness. However, using the fact that choosing σ equal to a constant yields a similarity matrix which is positive semidefinite [3], the matrix for this algorithm can be self-tuned by setting

$$\boldsymbol{W}_{ij} = \exp\left(-\frac{\|\boldsymbol{x_i}/\nu_i - \boldsymbol{x}_j/\nu_j\|^2}{c}\right),$$

where ν_i is defined in (4), and c is some fixed constant.

However, this self-tuning approach yielded worse results empirically than simply manually setting the scaling parameter σ. Figure 6 demonstrates the percentage of misclustered data points via the Nyström method with $\sigma = 1$ and the self-tuning approach, for the tangent spheres data (shown in Figure 5 (e)).



(a) (b)

Figure 6 Manually setting σ gives better results than the self-tuning method. An example is shown here for the tangent spheres dataset for (a) $\sigma = 1$ and (b) self-tuning. However, different datasets often favor different values of σ.

The Nyström method is more efficient than the exact algorithm because it is not necessary to compute eigenvectors of the entire dense similarity matrix. It has a time complexity of $O(nm^2 + m^3)$, which for $m \ll n$ is significantly less than the $O(n^3)$ runtime of the standard spectral clustering method.

2.4. **Spectral Clustering on a Budget.** The Nyström method relies on a submatrix to approximate the entire similarity matrix $\boldsymbol{W}$. There, one usually samples blocks or rows/columns at a time. Alternatively, one can simply sample the *entries* themselves at random. This is the approach of the *spectral clustering on a budget* method [14]. 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.

More specifically, the indices for the entries are chosen uniformly at random and without replacement from $\{(i, j) : i < j\}$. A new matrix $\widetilde{\boldsymbol{W}}$ is formed whose entries are given by

$$\widetilde{\boldsymbol{W}}_{ij} = \widetilde{\boldsymbol{W}}_{ji} = \begin{cases} \dfrac{2b}{n(n-1)} & \text{if } i = j \\ \boldsymbol{W}_{ij} & \text{if } (i, j) \text{ is queried} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We thus formulate the spectral clustering on a budget method formally as in Algorithm 6.

Algorithm 6 Spectral clustering on a budget

```
procedure ($\boldsymbol{W}$, $m$)        ▷ $n \times n$ similarity matrix $\boldsymbol{W}$, sample size $m$
    Create $\widetilde{\boldsymbol{W}}$ by selecting $m$ entries of $\boldsymbol{W}$ according to (8)
    Run spectral clustering efficiently using sparsified approximation $\widetilde{\boldsymbol{W}}$
end procedure
```

It is shown that the perturbation in the eigenvectors via the downsampling in this method can be bounded with high probability.

Theorem 2.4 (Spectral clustering on a budget [14])**.** *Suppose that the budget is $b \le \frac{1}{4}(n^2 - n)$, and denote by $\boldsymbol{v}_2$ and $\widetilde{\boldsymbol{v}_2}$ the 2nd (largest) eigenvectors of the Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, and corresponding perturbed Laplacian $\widetilde{\boldsymbol{L}}$, respectively. Then*

$$\min\left(\|\widetilde{\boldsymbol{v}_2} - \boldsymbol{v}_2\|_2, \|-\widetilde{\boldsymbol{v}_2} - \boldsymbol{v}_2\|_2\right) \lesssim \frac{4}{\lambda_2 - \lambda_3}\left(\frac{n^{5/3}}{b^{2/3}} + \frac{n^{3/2}}{b^{1/2}}\right),$$

where λ_i denotes the ith eigenvalue of $\boldsymbol{L}$ and the relation $\lesssim$ ignores lower order logarithmic factors.

This result shows that either $\widetilde{\boldsymbol{v}_2}$ or $-\widetilde{\boldsymbol{v}_2}$ is close to the true eigenvector and can be used for spectral clustering (note that the negative of the eigenvector preserves the same clustering). This closeness is again controlled by the eigengap of the Laplacian and the budget size b. If the data are well-clustered, $\boldsymbol{W}$ can be sparsified to have $O(n \log^{3/2} n)$ nonzero entries, and then spectral clustering can be performed in $O(n \log n)$ time.

3. Numerical Results for Approximation Methods

We next describe experimental results for the approximation methods of Section 2. We run each method using the datasets shown in Figure 5. Each cluster in these sets is clearly defined, and accuracy can thus be easily analyzed. The aim of these experiments is to compare and analyze the relationship between sample size, runtime, and accuracy for the approximation methods. We use the convention that a z% sample size refers to the percentage z of data used in the sample. For the fast spectral clustering method, this size corresponds to the number of centroids utilized, k/n. The error is reported in terms of the misclustering rate (6). Although experiments across different datasets were run on machines of varying specifications, each algorithm for a fixed dataset was run on the same machine to allow for fair comparison. The algorithms were implemented in Matlab as described in the pseudocode of Section 2 and the runtimes were computed via the *cputime* function. The experiments performed on the interlocked rings, interlocked half rings, and Gaussian strips were run on Intel Core 2 Duo E8500 3.16 GHz machines with 6 MB cache and 16 GB memory. The concentric spheres and concentric rings experiments were run on Intel Core i7 870 2.93 GHz machines with 4 cores, 8 MB cache and 16 GB memory. The tangent spheres experiments were run on an Intel Xeon W3520 2.67 GHz machine with 4 cores, 8 MB cache and 16 GB memory. A constant σ value was used for the Nyström method and the spectral clustering on a budget method for the interlocked rings dataset (for the values, see the tables below); otherwise, the self tuning approaches were used.

Table 2 and Figure 7 display the results for each algorithm on the Gaussian strips dataset, depicted in Figure 5 (a). For this dataset, the error rate and time for the fast spectral clustering method tend to decrease with small enough representative points k. This is most likely because at some point, the k-means clustering portion of the algorithm controls how the data clusters. To get a small error rate for a small enough k, the k-means clustering must work well with the dataset, in which case spectral clustering is perhaps not necessary (as is most likely the case for a well-separated dataset like the Gaussian strip set). However, for datasets for which k-means does not work well, such as the eye dataset below, we cannot assume that a very small k will have the same accurate results.

As seen in Figure 7, eSPEC generally performs better than the Nyström method, but fails when we take too small of a sample size. Spectral clustering on a budget performs worst overall, yielding the highest error rate if too small of a budget is used. A sufficiently large budget will allow the algorithm to run faster than the original spectral clustering algorithm, but a larger or slightly smaller budget does not significantly change the runtime. The Nyström method and eSPEC reach a point beyond which an increase in running time fails to produce a commensurate decrease in error rate. For small sample sizes, time does not change as much, but error rate can increase substantially. Since the Gaussian data consists of only $n = 200$ points, taking a sample as low as 5% might be too small for Nyström and eSPEC to work well. This is not a problem for fast spectral clustering since the data can be clustered with k-means clustering.

($n = 200$)	Fast		Budget		Nyström ($\sigma = 1$)		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
2%	0.0122	0.0036	0.2044	0.4914	0.0097	0.208		
5%	0.0156	0.068	0.1017	0.1308	0.0112	0.1344	0.0287	0.0379
10%	0.0181	0.0548	0.078	0.0015	0.014	0.1183	0.0271	0.0656
15%	0.0212	0.0046	0.0889	0	0.0218	0.0719	0.0275	0.0364
20%	0.0225	0	0.0858	0	0.0281	0.0321	0.0293	0.005
25%	0.0271	0	0.0952	0	0.0312	0.0249	0.0318	0.0098
30%	0.0300	0	0.088	0	0.0415	0.0042	0.0337	0
35%	0.0343	0	0.0924	0	0.0546	0.0088	0.0396	0
40%	0.0371	0	0.0877	0	0.0621	0	0.0446	0
50%	0.0771	0	0.0952	0	0.1026	0	0.0805	0

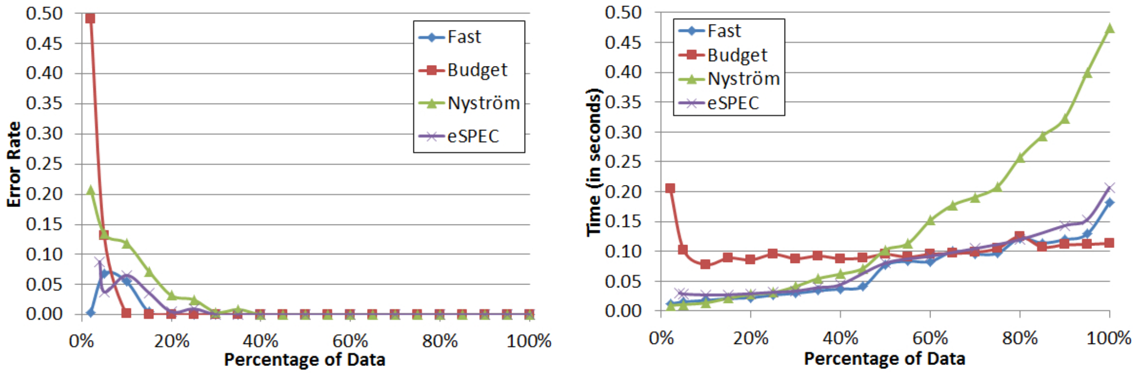
Table 2 The run time and error rate of each sample size for each approximation algorithm, ran on the Gaussian strip dataset.

Table 3 and Figure 8 depict the experimental results for the interlocked half rings dataset, depicted in Figure 5 (b). For this dataset, the Nyström method is ideal across the board. It has the smallest error rate and requires the least amount of time. Fast spectral clustering and eSPEC generally follow the same trend and are very similar in performance. For small datasets in general, even if k-means clustering does not work well with the dataset, fast spectral clustering is effective. Cells without numerical entries indicate regimes for which the parameters were too small for the algorithm to perform.

The results obtained from the concentric rings dataset (depicted in Figure 5 (c)) are displayed in Table 4 and Figure 9. The table shows that spectral clustering on a budget is very inaccurate with this dataset when the sample size is below 3%, as it misclusters 1 in 5 points. We see that all methods identify the clusters exactly once 10% of the data is used in the sample.

Table 5 and Figure 10 contain the results of the experiments on the concentric spheres dataset, depicted in Figure 5 (d). As the three dimensional analog of the concentric rings, it is not surprising we see similar results. It is to be noted that all algorithms cluster perfectly when the sample size is 5% or larger.

Table 6 and Figure 11 depict the results of the algorithms run on the tangent spheres dataset, depicted in Figure 5 (e). This dataset gives us a prime example of how approximate spectral clustering algorithms are ideal for large, structured data. The exact algorithm takes $6,365.4$ seconds, or almost 2 hours, to give



(a) (b)

Figure 7 Graphs show the (a) error rate and (b) CPU running time (in seconds) when using different sample sizes for all approximation algorithms. Gaussian strip dataset has $n = 200$ data points and sample sizes range from 0–100%.

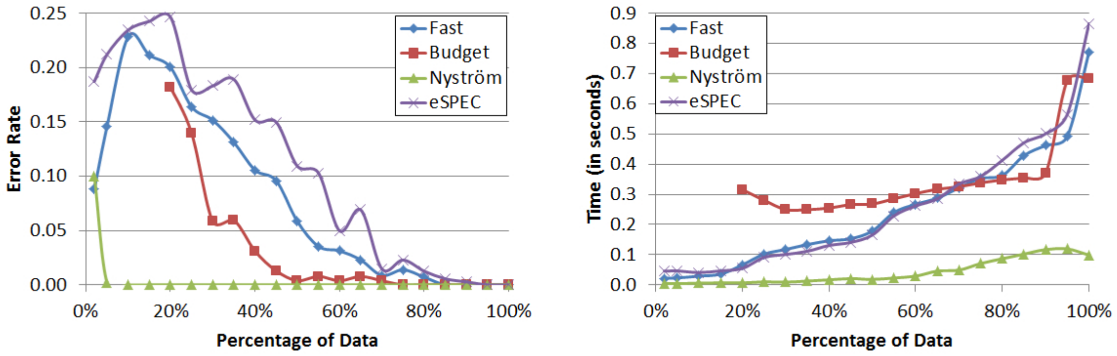
($n = 373$)	Fast		Budget		Nyström ($\sigma = 0.2$)		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
2%	0.0215	0.0881	-	-	0.005	0.1	0.0452	0.1872
5%	0.0231	0.1461	-	-	0.0044	0.002	0.0462	0.2122
15%	0.0356	0.2116	-	-	0.0069	0	0.0465	0.2426
25%	0.1017	0.1639	0.2805	0.1396	0.01	0	0.0908	0.1795
35%	0.1335	0.1312	0.2493	0.0595	0.0128	0	0.1114	0.1892
45%	0.1529	0.0959	0.2658	0.0129	0.0209	0	0.1407	0.1498
55%	0.2399	0.0354	0.2855	0.0078	0.0228	0	0.2287	0.1031
65%	0.2874	0.0229	0.3170	0.0077	0.0456	0	0.2852	0.069
75%	0.3526	0.0139	0.3382	0	0.0708	0	0.3594	0.0229
85%	0.4287	0	0.3547	0	0.102	0	0.4711	0.0056

Table 3 The run time and error rate of each sample size for each approximation algorithm, ran on the interlocked half rings dataset.

($n = 800$)	Fast		Budget		Nyström ($\sigma = 0.5$)		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
1.0%	0.0187	0.2699	0.3289	0.4979	0.0362	0.2404	0.0646	0.3076
2.0%	0.0259	0.0792	0.3017	0.4341	0.0356	0.1311	0.0661	0.1176
2.5%	0.0291	0.012	0.2839	0.2841	0.0356	0.1127	0.0646	0.0441
3%	0.0324	0.0062	0.2689	0.2061	0.0427	0.0682	0.0633	0.0102
4%	0.0674	0	0.2424	0.0564	0.0378	0.0258	0.0643	0.0046
5%	0.0627	0	0.2409	0.0131	0.0415	0	0.0655	0.0001
10%	0.1186	0	0.2792	0	0.0615	0	0.0764	0
15%	0.171	0	0.341	0	0.0967	0	0.127	0

Table 4 The (a) run time and (b) error rate of each sample size for each approximation algorithm, ran on the concentric rings dataset.

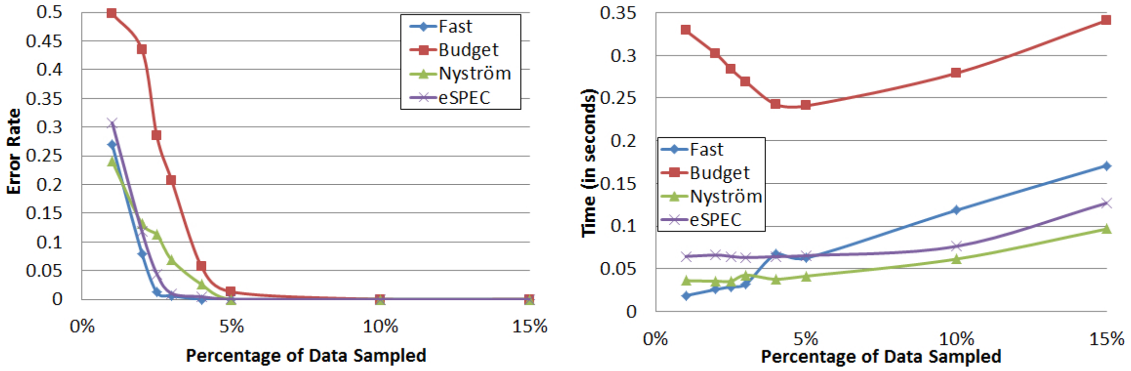
results, when nearly identical results can be given in seconds using approximations. When just 4.25% of the data is sampled, all of the algorithms perform with error less than 0.01, and all under a minute.



(a)

(b)

Figure 8 The (a) error rate and (b) CPU running time in seconds when using different sample sizes for all approximation algorithms. Interlocked half rings dataset has $n = 373$ data points and sample sizes range from 0–100%.



(a)

(b)

Figure 9 The (a) average error rate and (b) running time when using different sample sizes for all approximation algorithms. Concentric rings dataset has $n = 800$ data points and sample sizes range from 0–15%.

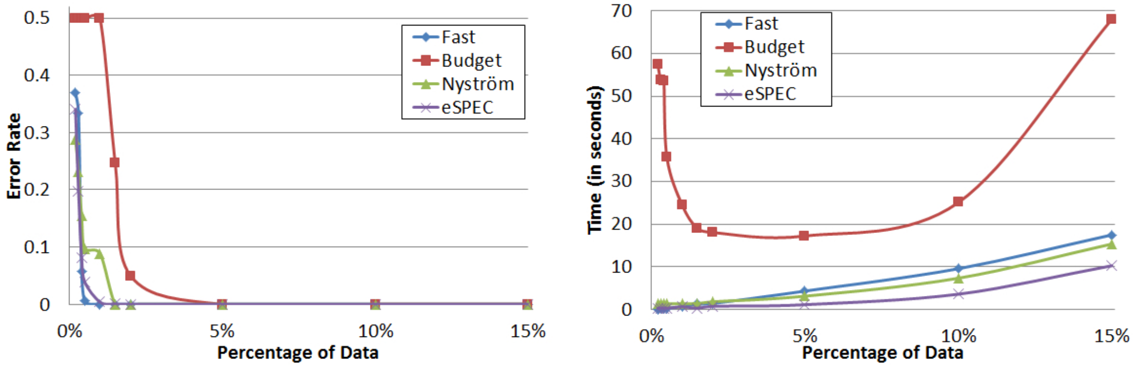
Finally, Table 7 and Figure 12 demonstrate the results of the algorithms run on the interlocked rings dataset of Figure 5 (f). We again see similar results to the tangent spheres dataset, most likely because the structure of the clusters are similar.

4. Case Study: The Attrition Problem

Clustering methods have a wide range of applications, ranging from image segmentation to social network identification. An application we focus on here is the attrition problem. In this setting, the data objects correspond to employees of a particular company, and each data vector contains a list of employee attributes. For example, age, salary, years at the company, number of children, age of children, etc. may all be relevant attributes. From this data, one wishes to distinguish a cluster of employees who are likely to leave the company from the other cluster of employees are likely to stay with the company. Such a classification allows companies to invest resources appropriately in an effort to maintain desired employees, saving significant expense and training time. We analyze here how this problem can be solved using spectral clustering methods. Because the number n of employees may be very large, and the number of attributes collected about the employees may also be very large, approximation methods

$(n = 5,000)$	Fast		Budget		Nyström $(\sigma = 1)$		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
0.2%	0.1292	0.3688	57.4396	0.4998	1.3934	0.2884	0.356	0.3405
0.3%	0.2296	0.3334	53.7683	0.4996	1.3672	0.2309	0.3588	0.1967
0.4%	0.3404	0.0569	53.6019	0.4997	1.3837	0.1539	0.3622	0.0812
0.5%	0.3644	0.0064	35.6188	0.4995	1.3937	0.0976	0.3469	0.0394
1.0%	0.7033	0	24.5065	0.4993	1.4212	0.089	0.6964	0.0041
1.5%	1.3407	0	18.9463	0.246	1.4546	0	0.3716	0.0006
2.0%	1.487	0	18.0827	0.0498	1.87	0	0.805	0
5.0%	4.3699	0	17.2624	0	3.1868	0	1.1784	0
10.0%	9.6112	0	25.099	0	7.3857	0	3.6448	0
15.0%	17.528	0	67.9837	0	15.3998	0	10.3045	0

Table 5 The run time and error rate of each sample size for each approximation algorithm, ran on the concentric spheres dataset.



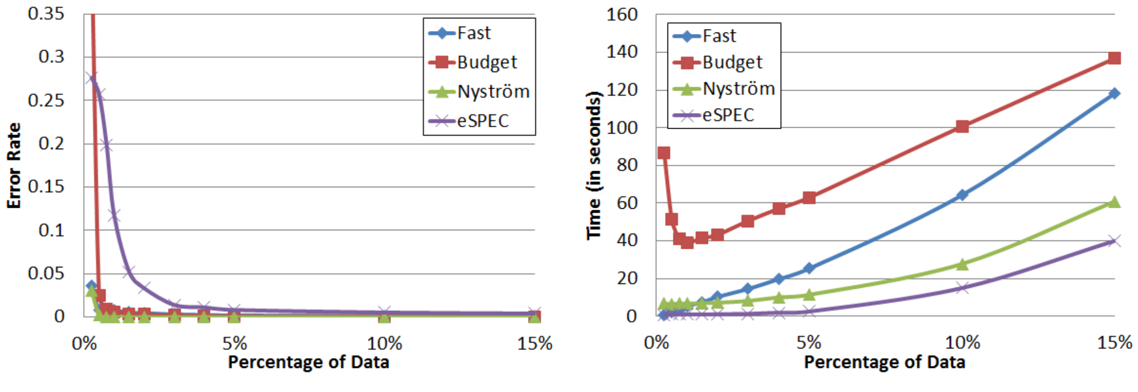
(a) (b)

Figure 10 The (a) average error rate and (b) running time when using different sample sizes for all approximation algorithms. Concentric spheres dataset has $n = 5,000$ data points and sample sizes range from 0–15%.

$(n = 10,000)$	Fast		Budget		Nyström $(\sigma = 1)$		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
0.25%	0.8022	0.036	86.5802	0.4268	6.647514612	0.030166	0.8833	0.2763
0.50%	1.9678	0.0075	51.4463	0.0247	6.592	0.0021	0.9316	0.2571
0.75%	3.428	0.0097	41.0183	0.0087	6.6759	0.00037	0.9382	0.1987
1%	5.5	0.0081	39.0598	0.0061	6.7492	0.000118	0.9734	0.1177
1.5%	7.4556	0.0052	41.432	0.0039	6.9648	0.000078	1.0056	0.0524
2%	10.3042	0.0047	43.2666	0.003	7.2534	0.000072	1.1082	0.0334
3%	14.5371	0.0032	50.6176	0.002	8.1726	0.00009	1.3419	0.0141
4%	19.7435	0.0025	57.1082	0.0015	10.0714	0.0001	1.9394	0.0112
5%	25.3318	0.0019	63.0219	0.0011	11.4333	0.0001	2.5556	0.0083
10%	64.279	0.0016	100.6472	0.000656	27.7863	0.0001	15.0004	0.0054
15%	118.1536	0.0013	136.7211	0.000466	60.9053	0.0001	40.0202	0.0042

Table 6 The run time and error rate of each sample size for each approximation algorithm, ran on the tangent spheres dataset.

are crucial to solve this problem efficiently. In contrast to the examples of Section 3, datasets in this setting are not only high dimensional, but accuracy is often difficult to quantify since there may no longer



(a) (b)

Figure 11 The (a) average error rate and (b) CPU running time in seconds when using different sample sizes for all approximation algorithms. Tangent spheres dataset has $n = 10{,}000$ data points and sample sizes range from 0–15%.

($n = 10{,}000$)	Fast		Budget ($\sigma = 0.5$)		Nyström ($\sigma = 0.5$)		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
0.5%	1.136	0.3298	96.986	0.3064	7.429	0.1676	0.8998	0.3252
1%	3.554	0.209	56.876	0.0048	7.575	0.0059	0.9284	0.2958
1.5%	4.538	0.1302	54.718	0.0027	7.759	0.0011	0.9466	0.2335
2%	6.21	0	57.23	0.0018	7.966	0.0009	0.9962	0.1579
3%	9.41	0	60.79	0.0014	8.595	0.001	1.1128	0.0691
4%	13.552	0	66.9	0.001	9.947	0.0009	1.3298	0.0217
5%	16.64	0	80.278	0.0008	10.778	0.001	1.9226	0.0039
10%	49.814	0	118.72	0.0008	24.164	0.001	11.185	0
15%	93.782	0	176.44	0.0007	56.83	0.001	29.976	0

Table 7 The run time and error rate of each sample size for each approximation algorithm, ran on the interlocked rings dataset.

be a notion of "correct" clusters. To overcome this last challenge, we utilize historical data about teacher attrition which will allow us to properly identify the appropriate clustering.

Each year, the National Center for Education Statistics sends out a follow-up survey to teachers originally selected for the Teacher Questionnaire in a Schools and Staffing Survey. 4,528 teachers were given one of two surveys according to their employment status. Teachers were classified as either *stayers*, *movers*, or *leavers*. Stayers are teachers who stayed at their current position, movers are teachers who continued teaching, but transferred schools, and leavers are teachers who left the position entirely. Leavers took the former teacher questionnaire while stayers and movers took the current teacher questionnaire. Both surveys contained different sets of questions; the dataset used in our experiments is made up of common questions in both surveys from the 1994–1995 school year. The attributes include household income (broken into intervals), marital status (coded as 0/1/2 for never married / married / separated), number of dependent children, age of youngest child, and dissatisfaction ratings. Because many teachers had the same responses in the six variables, a dummy variable was added so that the algorithm would recognize that the teachers are different people. The dummy variable was drawn uniformly between 0 and 1.

Spectral clustering of the entire dataset yielded a small cluster on 197 teachers who were never married and did not have children or other dependents. All of their household incomes ranged from $60,000 to

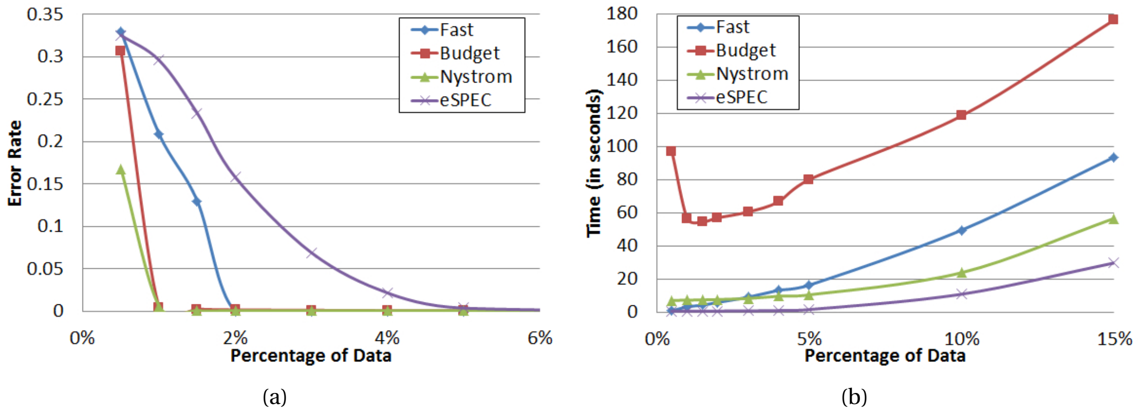


Figure 12 The (a) average error rate and (b) CPU running time in seconds when using different sample sizes for all approximation algorithms. Interlocked rings dataset has n = 10,000 data points and sample sizes range from 0–15%.

$74,999 and none of the teachers expressed dissatisfaction in the survey. The number of stayers, movers, and leavers are summarized in Table 8.

Drawing conclusions about the likelihood of attrition for a group of teachers depends on the classification of movers. From a school's point of view, a mover is an attritor, but from the state's point of view, a mover is still a teacher. In Table 8, if movers are considered leavers, then the proportion of attritors in Cluster 1 is significantly different from the proportion of attritors in Cluster 2. The results yield a one-tailed p-value of 0.01 where teachers with the same characteristics as Cluster 1 are less likely to resign. However, if movers are considered stayers, then the proportion of attritors in Cluster 1 is not significantly different from the proportion of attritors in Cluster 2. In this case, the one-tailed p-value is 0.1950 where teachers with the same characteristics as Cluster 1 are more likely to resign. Because conclusions were affected by the classification of teachers who moved, these teachers were not included in most experiments. However, one can easily use the applications of spectral clustering to cases where teachers are considered either one or the other.

While the age of a teacher's youngest child can provide valuable information about the teacher's age itself, setting different values for this variable for teachers without children gave varying results. For example, if the youngest child's age is set to 50, a large distance away from the maximum value of 38, spectral clustering tends to group teachers with older children together with childless teachers. If the youngest child's age is set to −1, a closer but impossible age, spectral clustering tends to group teachers with newborns together with childless teachers. In the following experiments, we used −1 as the age for teachers without children, though the technicalities of this variable bring us to question if interaction terms ought to be considered when working with data of this nature.

Spectral clustering was applied on subgroups of teachers, such as teachers without children or unmarried teachers. Most experiments yielded clusters with a mix of teachers who stayed and teachers who left. If teachers who moved were included, they would generally be mixed in both clusters as well. An example of this can be seen in Table 8 where we applied spectral clustering to teachers with children; movers were removed. Although both clusters contained a mix of stayers and leavers, in a two-proportion Z-test, we obtain a one-tailed p-value of about 0.0023, providing evidence that Cluster 2 has a greater proportion of teachers who will quit. We conclude that teachers with similar characteristics as those in Cluster 2 are more likely to quit.

One of the more interesting results that spectral clustering produced on the teacher data was the case where one of two clusters contained only teachers who left. In this run, movers were removed and only teachers without children were considered. Spectral clustering grouped 173 teachers together, all who

Status	Cluster 1	Cluster 2
Stayer	92	1,666
Mover	24	1,016
Leaver	81	1,649
Total	197	4,331

Status	Cluster 1	Cluster 2
Stayer	100	870
Leaver	48	699
Total	148	1,569

Status	Cluster 1	Cluster 2
Stayer	0	788
Leaver	173	810
Total	173	1,598

Table 8 Resulting clusters of the entire teacher dataset (left), with movers eliminated (center) and with teachers with children and movers eliminated (right).

had quit. They were all married and had household incomes in the $60,000 to $74,999 range. The teachers in this group expressed at most one dissatisfaction in the survey. In Table 8, this group is labeled Cluster 1. Cluster 2 consists of all other teachers that were not movers and did not have children. Among the 1,598 teachers in Cluster 2, 810 quit. In a two-proportion Z-test, this gave us a one-tailed p-value of less that 0.0001, which supports the idea that teachers similar to those in Cluster 1 are more likely to quit than those similar to teachers in Cluster 2.

To ensure that spectral clustering worked and would give us accurate clusters, the clustering algorithm was applied to a 1/3 sample of a subgroup of teachers. Results were used to try to predict the remaining 2/3. For example, in the case of the 488 unmarried teachers, 163 teachers were sampled (Table 9). Spectral clustering grouped 23 of the teachers in one group because they all did not express complaints and did not have children or other dependents. Among this first cluster, 17 had quit while 38 of the 140 teachers in the second cluster quit. This yielded a one-tailed p-value of less than 0.0001 where teachers in the first cluster are more likely to quit. Going through the 325 unsampled teachers, 55 displayed the same characteristics as the first cluster. Proportionally, our prediction that 41 of the 55 teachers would quit was not bad considering that in actuality 38 teachers quit. Our prediction for the second group of teachers was further off, but still supported the finding that teachers similar to those in the first cluster are more likely to quit than teachers similar to those in the second cluster. Spectral clustering was able to group the 55 teachers together in its run with the remaining 2/3 data points.

	With Children Cluster	Without Children Cluster
SC on 1/3	17/23	38/140
Predicted	41/55	73/270
SC on 2/3	38/55	94/270

Table 9 Prediction given by spectral clustering for the teacher dataset (married teachers and movers eliminated).

Although [13] recommended using the 7th nearest neighbor to help determine the similarity bandwidth of each point, using different values of nearest neighbor for this dataset yielded different clusters that provided valuable information. We applied spectral clustering to married teachers who were not movers using the 7th, 50th, and 100th nearest neighbor. All teachers in the smaller cluster (Cluster A) using the 7th nearest neighbor were found in the same cluster (with other teachers) when using the 100th nearest neighbor. That same cluster, with the newly added teachers, in the 100th nearest neighbor was also found in the same cluster with almost the rest of the teachers when using the 50th nearest neighbor. Define Cluster B as teachers grouped with teachers in Cluster A using the 100th nearest neighbor. Define Cluster C as teachers grouped with teachers in Cluster A and B using the 50th nearest neighbor. The remaining teachers make up Cluster D. A summary of the characteristics of teachers in each cluster can be found on Table 10. Note that the table does not reflect correlations; for example, a dissatisfaction value of 1 only appears in Cluster B if the teacher does not have a child.

We found that the difference between all four clusters is statistically significant with a p-value of less than 2.2E-16. With this, we can obtain a ranking of the teachers where teachers with 2 or 3 points of dissatisfaction are most likely to resign. Teachers with a young only child and teachers without children, excluding those with the exact characteristics of Cluster A, are very likely to resign. Teachers with the exact characteristics of Cluster A could possibly resign, while all other teachers with at most one dissatisfaction point are not likely to resign. In summary, a ranking of teachers who are most likely to quit teaching is achievable with spectral clustering. It brings us to consider multiple clusters in spectral clustering.

Characteristics	Cluster A	Cluster B	Cluster C	Cluster D
Household Income	60,000–74,999	Varies	Varies	Varies
Children	None	0 to 1	Varies	Varies
Youngest Child	-1	-1 to 4	Varies	Varies
Other Dependents	None	None	0 to 1	0 to 3
Dissatisfaction	None	0 to 1	0 to 1	2 to 3
Stayer	92	398	772	0
Leaver	81	567	480	214
Total	173	965	1,252	214

Table 10 Four clusters given by altering the tuning parameter on the teacher dataset (unmarried teachers and movers eliminated).

Our analyses of spectral clustering on the teacher data was facilitated by looking at subgroups of teachers. It is perhaps that the variables that create the divide for the subgroups (i.e. unmarried teachers only, teachers without children only, etc.) interact with other variables, such as age of youngest child. Variable transformations, interaction terms, multiple clusters, and weighting are thus important points of consideration when working with spectral clustering on attrition-like data.

4.1. **Approximation Results.** To measure the effectiveness of the approximation algorithms on the teacher dataset, we ran each one given a different sample size 10 times and compared average run time and error rate. The "movers" category was removed for simplicity. As the ground truth, we use the answers obtained by the exact spectral clustering algorithm. In other words, we measure the ability of the approximation algorithms to give the same answer as exact spectral clustering in a shorter amount of time (the exact algorithm ran in 476.47 seconds).

Although the results are similar to those obtained using visually apparent clusters, we see two major differences. First, as seen in Table 11, spectral clustering on a budget was not necessarily the slowest or most inaccurate algorithm of the group. Secondly, as seen in Figure 14, it took a longer time for the algorithms to reach zero error. This is potentially due to the use of proximity of the clusters. Perhaps spectral clustering on a budget handles less structured clusters better than the others. Still, it displays an odd progression in terms of run time – one that is not entirely upward sloping, as seen in Figure 14. This leads us to believe it may be unstable in this setting. For this kind of dataset, fast spectral clustering or eSPEC may offer more advantages. Alternatively, if it is acceptable for the error rate to be up to 5%, Nyström gives adequate results the quickest. A depiction of the tradeoff between efficiency and accuracy for this dataset is given in Figure 13.

5. Discussion

Fast spectral clustering frequently gives the most accurate results in the shortest running time for small datasets using a small k. For easily clustered data, this may be due to the k-means algorithm overpowering the fast spectral clustering algorithm for really small k values. The Nyström method often performs quickly and accurately as well, especially on the larger or more complicated datasets. eSPEC is the fastest when n is extremely large, but also the most inaccurate.

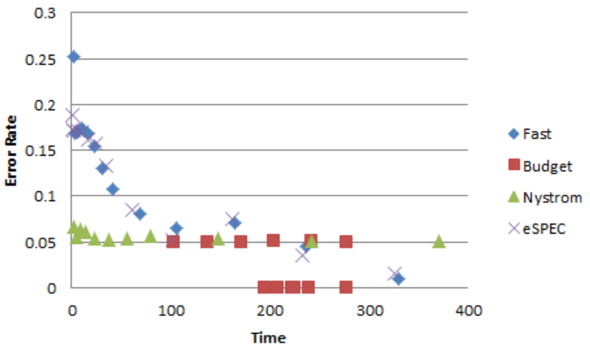
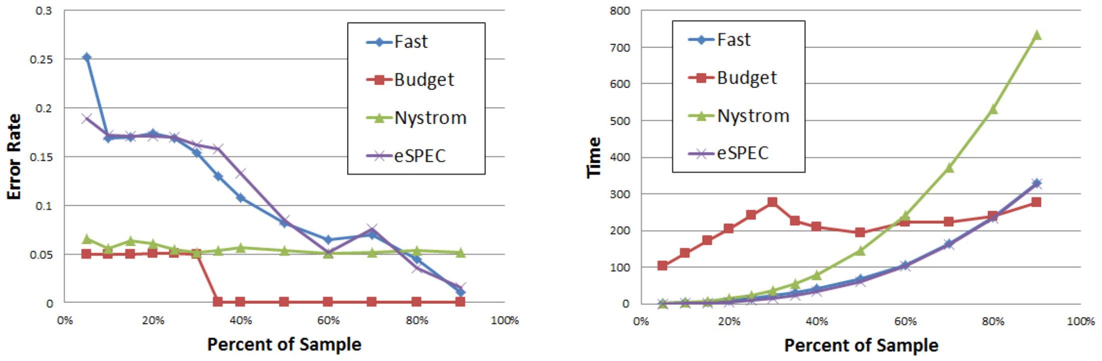


Figure 13 Error rate versus time (in seconds) plot for the teacher data (without movers).



(a) (b)

Figure 14 Sample sizes range from 5–100% of the teacher dataset with $n = 3,488$, and without the classification of movers.

($n = 3{,}488$)	Fast		Budget		Nyström		eSPEC	
Sample Size	Time	Error	Time	Error	Time	Error	Time	Error
5%	1.9906	0.2518	102.9295	0.0499	1.7644	0.066	0.4243	0.1889
10%	3.4086	0.1691	136.6257	0.0499	4.1699	0.0551	0.8143	0.1721
20%	10.3694	0.1735	204.2521	0.0502	14.2835	0.0601	5.4663	0.1709
30%	22.8323	0.1541	276.855	0.0493	37.1969	0.0517	16.3941	0.1618
40%	41.5103	0.1079	208.5421	0	80.0753	0.0567	34.2765	0.1328
50%	69.0788	0.0812	194.2056	0	146.9904	0.0536	61.5502	0.0843
60%	106.2819	0.065	222.941	0	241.4677	0.0506	102.1619	0.052
70%	164.4188	0.07	223.8302	0	371.103	0.0511	162.096	0.0752
80%	236.8828	0.045	239.8047	0	532.3269	0.0533	232.819	0.035
90%	329.4538	0.01	277.4946	0	733.4184	0.0511	326.2121	0.015

Table 11 The run time and error rate of each sample size for each approximation algorithm, ran on the teacher dataset.

Intuitively, we sacrifice accuracy for efficiency when we run the approximation algorithms on a relatively small set of points compared to the dataset size. However, the trend is not so apparent for spectral clustering on a budget. The other algorithms face approximately the expected tradeoff. Across all datasets, spectral clustering on a budget often takes longer than the other three with limited advance in accuracy. Interestingly, it consistently reaches a point where smaller sample sizes actually make it increase in running time. Thus, it may be preferable to utilize one of the other three algorithms, depending on the size of the data and the goal of the clustering results.

In addition, when given the right data, spectral clustering can find similarities in individuals that may point to employees at high risk of attrition. Using a subset of the teacher dataset, we found we could predict with some accuracy which teachers had left. This shows that if possible, breaking the data down and running spectral clustering on smaller groups is very useful. Approximation methods did not perform quite as well on the teacher dataset, but they do give accurate results and cut down run time by a few minutes. If run on a larger employee dataset, they would likely increase efficiency by a greater factor. With finer tuning of parameters and variable choices, even more improvements may be possible.

Acknowledgements

We would like to thank our advisors Christos Boutsidis and Deanna Needell. We also thank Mike Raugh, Stacey Beggs, Dimi Mavalski, and all the faculty at the Institute of Pure and Applied Mathematics for directing and coordinating this summer. Lastly, thank you to IBM and NSF for funding this project.

References

[1] C.T.H. Baker and CTH Baker. *The numerical treatment of integral equations*, volume 13. Clarendon press Oxford, 1977.

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

[3] M. Cuturi. Positive definite kernels in machine learning. *arXiv preprint arXiv:0911.5367*, 2009.

[4] P. Drineas, R. Kannan, and M.W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[5] 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.

[6] L. Huang, D. Yan, M.I. Jordan, and N. Taft. Spectral clustering with perturbed data. *Advances in Neural Information Processing Systems (NIPS)*, pages 705–712, 2008.

[7] B. Hunter and T. Strohmer. Performance analysis of spectral clustering on compressed, incomplete and inaccurate measurements. Submitted, 2011.

[8] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[9] D.J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[10] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. *WALCOM: Algorithms and Computation*, pages 274–285, 2009.

[11] EJ Nystr "o m. "U ber practical aufl ö solution of integral equations with applications to boundary value problems. *Acta Mathematica*, 54(1):185–204, 1930.

[12] M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. *Advances in Neural Information Processing Systems*, 17:1057–1064, 2005.

[13] P. Perona and L. Zelnik-Manor. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17:1601–1608, 2004.

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

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

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

[17] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

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

[19] X. Wu and V. Kumar. *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis, 2009.

[20] D. Yan, L. Huang, and M.I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 907–916. ACM, 2009.