A Sequential Sampling Framework for Spectral *k*-Means based on Efficient Bootstrap Accuracy Estimations: Application to Distributed Clustering

DIMITRIOS MAVROEIDIS, Radboud University Nijmegen, The Netherlands
PANAGIS MAGDALINOS, National and Kapodistrian University of Athens, Greece

The scalability of learning algorithms has always been a central concern for Data Mining Researchers and nowadays, with the rapid increase of data storage capacities and availability, its importance has grown further. To this end, sampling has been studied by several researchers in an effort to derive sufficiently accurate models using only small data fractions. In this paper we focus on Spectral *k*-Means, i.e. the *k*-Means approximation as derived by the spectral relaxation, and propose a sequential sampling framework that iteratively enlarges the sample size until the *k*-Means results (objective function and cluster structure) become indistinguishable from the asymptotic (infinite-data) output. In the proposed framework we adopt a commonly applied principle in Data Mining research that considers the use of minimal assumptions concerning the data generating distribution. This restriction imposes several challenges mainly related to the efficiency of the sequential sampling procedure. These challenges are addressed using elements of Matrix Perturbation Theory and Statistics. Moreover, although the main focus is on Spectral *k*-Means, we also demonstrate that the proposed framework can be generalized to handle Spectral Clustering.

The proposed sequential sampling framework is consecutively employed for addressing the Distributed Clustering problem, where the task is to construct a global model for data that reside in distributed network nodes. The main challenge in this context is related to the bandwidth constraints that are commonly imposed, thus requiring that the distributed clustering algorithm consumes a minimal amount of network load. This illustrates the applicability of the proposed approach as it enables the determination of a minimal sample size that can be used for constructing an accurate clustering model that entails the distributional characteristics of the data. As opposed to the relevant distributed *k*-means approaches, our framework takes into account the fact that the choice of the number of clusters has a crucial effect on the required amount of communication. More precisely, the proposed algorithm is able to derive a statistical estimation of the required relative sizes for all possible values of *k*. This unique feature of our distributed clustering framework enables a network administrator to choose an economic solution that identifies the crude cluster structure of a dataset and not devote excessive network resources for identifying all the “correct” detailed clusters.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms: Spectral, Clustering, Asymptotic Convergence, Sampling

Additional Key Words and Phrases: Matrix Perturbation Theory, Bootstrapping, Distributed Clustering

1. INTRODUCTION

An important practical problem in Data Mining is related to the determination of the sufficient sample size that is required such that an accurate model, that reflects the distributional characteristics of the data is constructed [Domingo et al. 2002; Provost and Kolluri 1999; Provost et al. 1999;

Part of this work was conducted while the first author was employed at the European Commission, Joint Research Center (JRC), Ispra, Italy.

Author’s addresses: D. Mavroeidis, Faculty of Science, University of Nijmegen, Postbus 9010, 6500GL Nijmegen, The Netherlands; email: mavroeidis@cs.ru.nl; P. Magdalinos, Department of Informatics and Telecommunications, Faculty of Science, National and Kapodistrian University of Athens, Panepistimiopolis, Ilissia, Athens 15784, Greece ;email: panagis@di.uoa.gr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© YYYY ACM 0000-0000/YYYY/01-ARTA $10.00

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Scheffer and Wrobel 2003; Scholz 2005]. Depending on the nature of the data (such as stream data, dynamic data or static data) and the properties of the learning algorithms (such as asymptotically convergent or inconsistent) various approaches have been proposed. However, albeit the importance and significance of this problem, there exist certain popular data mining paradigms, such as Spectral Clustering [von Luxburg 2007] and Spectral *k*-means [1] [Ding and He 2004; Zha et al. 2001; Gordon and Henderson 1977] that have not been adequately analyzed in this respect.

The relatively small attention that Spectral *k*-means has received can be attributed to the fact that the Lloyd's standard *EM*-style *k*-means algorithm [Lloyd 1982] presents an efficient and easy to implement approach for approximating the minimum sums of squares clustering problem. If we attempt to make a high-level comparison between Spectral *k*-means and Lloyd's *k*-means the arguments will boil down to the standard dilemma between transforming the original clustering formulation to an easy-to-solve, deterministic and convex optimization problem, as opposed to using a heuristic, local-minima algorithm that requires certain tuning (such as the initialization of cluster centers) but performs remarkably well in practice. Due to the popularity of Lloyd's algorithm, several efficient sampling strategies have been proposed in various application contexts (such as [Ailon et al. 2009; Datta et al. 2009; Zhou et al. 2007; Bradley et al. 1998]).

These methods generally consider the desired number of clusters *k* as input and aim to derive a sufficiently accurate estimation of the cluster centers or the cluster objective. One issue that is commonly overlooked is the fact that the choice of *k* can have a significant effect on the required sample size for approximating the cluster results. As we analyze in detail in Sections 7.2 and 7.3, the discovery of the detailed cluster structure or even a wrong choice of *k* that attempts to split a dense cluster, can require large sample sizes, much larger than when *k* is correctly configured to identify the crude cluster structure of the data. Thus, if we consider that the data gathering process is associated with a cost, it is natural to desire a mechanism that is able to provide us with the comparative sample size requirements for all possible choices of *k*. As we analyze in Sections 7.2 and 7.3 the proposed framework has this property and can derive that the construction of a reliable clustering for *k* clusters requires a smaller bandwidth than for and other number of clusters.

The little attention that Spectral Clustering has received with respect to sufficient sample size determination can be attributed to the fact that its asymptotic behavior, i.e. its behavior as sample size tends to infinity has only recently been characterized [von Luxburg et al. 2008]. The recent results in [von Luxburg et al. 2008] demonstrate that Spectral Clustering is consistent, i.e. converges under mild assumptions to a steady partition of the whole data space, thus motivating the consideration of algorithms that aim in determining the required sample size such that the clustering algorithm approximates sufficiently the asymptotic-infinite data cluster structure.

Sampling strategies have been extensively considered in the application area of Distributed Data Mining where the main task is to construct a reliable clustering (such as [Datta et al. 2009] and references therein) of the available network data while using a minimal amount of bandwidth resources. The consumption of bandwidth resources is necessary since each network node has only a certain portion of the available data and thus, the fragmented information needs to be accumulated in order to construct a reliable cluster model that reflects the global distributional characteristics of the network data. The role of a distributed clustering algorithm is to ensure that the data accumulation process will be performed in an economic manner, consuming a minimal amount of network resources. This illustrates the direct applicability of the proposed sequential sampling framework to Distributed Clustering, since it allows for the determination of the minimal sample size that needs to be communicated such that a reliable clustering is constructed.

It should be noted that although this work presents the first approach that considers the problem of Distributed Spectral Clustering and Distributed Spectral *k*-means, there exists a large body of literature on Distributed Lloyd's *k*-means for several types of networks [Datta et al. 2009; Bandyopadhyay et al. 2006; Datta et al. 2006; Forman and Zhang 2000; Dhillon and Modha 2000]. A

[1] Throughout the rest of this paper we will refer to the continuous relaxation approach for approximating *k*-Means, as Spectral *k*-Means.

shortcoming of these approaches is that they are committed to a fixed number of clusters *k* and do not take into account the effect that the choice of *k* has to the required bandwidth consumption. We should note here that there exist some works that aim in detecting the number of well separated clusters in a distributed manner [Tasoulis and Vrahatis 2004], however these approaches do not quantitatively relate the "correct" number of clusters with the required sample size of *k*-means. As we have stated earlier, the choice of *k* can have a significant effect on the sample size requirements, and thus an inappropriate *k* selection can lead to excessive network load consumption. Based on this observation it can be argued that a distributed clustering algorithm should have the ability to estimate the relevant bandwidth requirements for all *k*, thus providing a network administrator with the ability to select a *k* that derives an economic crude cluster structure of a dataset. The proposed Distributed Spectral Clustering and Distributed Spectral *k*-means approaches have this feature and in fact this constitutes a distinct advantage they have over the relevant Lloyd-type Distributed *k*-means approaches.

Before we present the contributions of this work we will provide a brief non-technical summary of the proposed framework. The sequential sampling algorithm initially considers as input a large dataset that cannot be directly analyzed and randomly splits it in smaller samples. Consequently, these samples are iteratively merged in a sequential manner, until our theoretical analysis guarantees that the desired approximation levels with respect to the objective function and the cluster results are reached. In the heart of the proposed approach lies an efficient Bootstrap-based methodology that assesses at each sequential step whether the input approximation requirements are achieved. The efficiency of the proposed methodology is based on Matrix Perturbation Theory results that allow us to relate the accuracy of the elements of the input data matrix to the accuracy of its spectrum. We also demonstrate that our framework can be generalized to handle Normalized Spectral Clustering, when the object-similarity (which is an input in Spectral Clustering) is defined in the form of an inner-product. Experiments demonstrate the convergent behavior of the proposed framework and also provide insights on the appropriate choice of the input parameters. More precisely, the experimental results lead to the definition of an automated selection process for the input requirements such that the quality of the sub-sample considered at the termination of the sequential sampling process tightly approximates the asymptotic classes-cluster performance. Based on the automatic tuning of the input-requirements, our approach can be considered as a stand-alone algorithm that automatically determines the required sample size such that the clustering performance does not further improve when larger data sizes are considered. With regards to the application focus, we conduct extensive experiments against distributed *k*-Means approaches and demonstrate the superiority of our approach with respect to bandwidth consumption.

The contributions of this paper can be summarized in the following:

- **A new perspective to sequential sampling k-means:** We introduce a novel perspective to the sequential sampling problem for Spectral *k*-means and demonstrate that it can be reduced to the statistical estimation of the appropriate feature-to-feature similarities. This view is different than most sampling approaches for *k*-means that aim in accurately estimating the relevant cluster centers or objective function. As we will analyze subsequently in more detail, this is an important distinction and allows our framework to be independent of an ad-hoc prior selection of parameter *k*.
- **Efficient statistical accuracy estimation of the appropriate quantities:** In the proposed framework, we do not make any assumptions regarding the data generating distribution, thus a challenge that arises is concerned with the efficient computation of the appropriate statistical accuracy estimates. In this context we propose an efficient bootstrap-based methodology that presents an improvement in terms of efficiency over the direct application of Bootstrapping on the spectral solution.
- **Number of clusters and required sample size:** Another novel feature of the proposed framework is that it provides at each step of the sequential sampling process an estimation of the required relative sample sizes for all possible values of *k*. I.e. based on our framework we can identify

the number of clusters *k* that attains the smallest sample size requirements, and we can also draw conclusions such as: “the reliable identifications of a three cluster structure requires less data than a two cluster structure but more data than a four cluster structure”. This is a unique feature that, to the extend of our knowledge, is not provided by other relevant sampling-based clustering frameworks.

— **Number of clusters and required bandwidth:** In the context of Distributed clustering, our approach offers the unique feature of providing a statistical estimation of the relative bandwidth requirements for all possible values of *k*. This is an important feature that provides a network administrator with better control over the Distributed Clustering process.

2. DISTRIBUTED CLUSTERING AND SAMPLING APPROACHES

In the Data Mining literature, the term “Distributed Clustering” is largely overloaded and is employed to refer to diverse distributed data mining problems. Thus, in order to clarify the application context of this work, we will initially provide a brief categorization of the distributed clustering literature and also present the central problems that are considered. In this analysis we will also highlight the relevance of sampling approaches and justify why it is natural to consider the application of distributed clustering to the proposed sequential sampling framework.

An initial categorization of the Distributed Clustering literature can be made on the basis of the type of distributed network that is considered. Several Distributed Clustering approaches have been proposed for structured and unstructured Peer-to-Peer networks (such as [Datta et al. 2009; Bandyopadhyay et al. 2006; Hammouda and Kamel 2007]) and Sensor networks (such as [Younis and Fahmy 2004; Bandyopadhyay and Coyle 2003]). These networks specify several different requirements, for example in sensor networks, due to the low energy resources, it is required that a minimal number of local (sensor level) computations are performed. Apart from the application specific approaches, there exist more generic works that define a set of requirements for the structure of the network or the data that are contained (such as [Datta et al. 2006; Januzaj et al. 2004; Kargupta et al. 2000; Klusch et al. 2003; Kriegel et al. 2005; Zhang et al. 2008]) and then design distributed clustering algorithms that satisfy these requirements. The diversity of distributed data mining methods can be observed even in specific application areas, such as P2P networks, where there exist several differentiations between various types of P2P networks such as structured, unstructured or semi-structured. Albeit the large diversity that exists, a requirement that is commonly imposed is related to the minimization of the required bandwidth resources. This requirement highlights the relevance of sequential sampling that allows for the determination of the minimal sample size that needs to be communicated for constructing a representative clustering model of the whole network. The relevance of random sampling in distributed networks can also be illustrated by the fact that it has been considered as a separate research problem (i.e. in [Arai et al. 2007; Awan et al. 2006]).

Due to the large volume of work that exists in the topic of distributed clustering, prior to presenting the specific technical details of the proposed framework we will carefully identify the research problems that still remain open in the area. An open problem can be considered as the definition of “distributed-versions” of centralized algorithms that have not yet been introduced. The “distributalization” of centralized algorithms would enhance the tools that networks administrations can employ and possibly define new, or highlight the importance of old research problems in Distributed Data Mining. Based on this observation we consider in this paper the sampling-based “distributalization” of Spectral *k*-means and Spectral Clustering. To the extend of our knowledge there do not exist sampling based distributed versions for these algorithms. The proposed framework introduces a novel perspective to the Distributed *k*-means problems which is reduced to the statistical estimation of the feature-to-feature similarities as opposed to the relevant Distributed *k*-means approaches that focus on the estimation of the respective cluster centers or cluster objective. This introduces certain novel insights and could lead to a development of network-specific communication efficient algorithms for feature-feature similarity estimations.

Another open problem that can be considered is related to the analysis of the effect of clustering parameters to the required bandwidth consumption. For certain parameters, such as the number of clusters, it is known that clearly separated clusters require less data/bandwidth for the statistical estimation of their cluster centers [Guha et al. 1998]. To the extend of our knowledge, this qualitative knowledge has not been quantitatively analyzed for specific distributed clustering algorithms. Based on this observation, we consider in this paper a Distributed Clustering algorithm, that can automatically assess the relative required sample sizes for all possible choices of k (number of clusters). This is a "built-in" feature of the proposed framework and no extra resources need to be devoted for this estimation.

Now that we have presented the main open problems that will be considered in the application area of Distributed Clustering, we can move on and provide a brief introduction to the algorithms we study, i.e. Spectral k-means and Spectral Clustering. Their introduction will clarify the differentiations between Spectral k-means and Lloyd's k-means that will eventually lead to the formulation of the proposed sequential sampling framework.

3. K-MEANS, SPECTRAL K-MEANS AND SPECTRAL CLUSTERING

k-Means clustering, is one of the most popular methods for identifying groups in data. It considers as input the number k of clusters and aims in retrieving the k clusters that minimize the following objective function.

$$J_k = \sum_{j=1}^{k} \sum_{i \in C_k} ||x_i - m_j||^2$$

where x_i are the input data and m_k are the cluster centroids. The most well known heuristic for k-Means is Lloyd's algorithm [Lloyd 1982]. Due to its wide use and practical effectiveness, Lloyd's algorithm is commonly referred to as the k-Means algorithm.

Another approach that has been proposed, considers the spectral relaxation for approximating the k-Means objective [Ding and He 2004; Zha et al. 2001; Gordon and Henderson 1977]. These approaches are based on the fact that the k-Means optimization problem is equivalent to the following trace maximization problem.

$$\begin{aligned} &min_Y(\mathbf{Tr}(XX^T) - \mathbf{Tr}(Y^T XX^T Y)) \equiv \\ &max_Y(\mathbf{Tr}(Y^T XX^T Y)) \end{aligned} \tag{1}$$

Where X is the *object* × *feature* matrix [2] and Y is a matrix with size $n \times k$ (n is the number of objects and k is the number of clusters). Y is defined as:

$$Y_{ic} = \begin{cases} \frac{1}{\sqrt{|\pi_c|}} & \text{if object i} \in \pi_c \\ 0 & \text{otherwise} \end{cases}$$

with $|\pi_c|$ denoting the size of cluster c. It can be observed that Y is an orthonormal matrix that contains the discrete cluster assignments for the data objects.

The formulation of k-Means as a trace maximization problem makes apparent the relevance of spectral techniques, since if we relax matrix Y to be any orthogonal matrix, the continuous relaxation solution can be derived by the k dominant eigenvectors[3] of the object-similarity matrix XX^T. More precisely, the matrix Y that maximizes the objective function (with the continuous relaxation and the constraint $Y^T Y = I_k$) contains the k dominant eigenvectors as columns. Since the results are continuous, and do not correspond to crisp cluster assignments, an additional step is required for discretizing the results. To this end several approaches have been proposed for discretizing the continuous solutions ([von Luxburg 2007] and references within), with the most popular choice being Lloyd's k-Means.

[2] In the context of this work we will always denote X as the *object* × *feature* matrix

[3] In the context of this work we will refer to the k dominant eigenvectors as the eigenvectors that correspond to the k largest eigenvalues

Normalized Spectral clustering works in a similar manner and aims in retrieving the k clusters that minimize the Normalized Cut objective function (which is also NP-Hard):

$$NCut(A_1, ..., A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

where $cut(A, \overline{A}) = \sum_{i \in A, j \notin A} W(i, j)$, $vol(A) = \sum_{i \in A} \sum_{j=1}^{n} W(i, j)$, n is the number of objects and $W(i, j)$ is the similarity between objects i and j.

This problem can also be stated as a Trace minimization problem [Shi and Malik 2000] in the form:

$$min_Y \mathbf{Tr}(Y^T(I - D^{-1/2} W D^{-1/2})Y) \tag{2}$$

where W is the object similarity matrix, D is the degree matrix as induced by W and Y is the orthogonal matrix with size $n \times k$ (n number of objects and k the number of clusters) defined in a similar manner as above. Essentially Y contains the discrete cluster assignments for the data objects. If we relax the matrix Y to be any orthogonal matrix, the continuous relaxation solution can be derived by the k eigenvectors that correspond to the k smallest eigenvalues of the normalized Laplacian $L = I - D^{-1/2} W D^{-1/2}$. It should be noted that in the case of 2-way clustering, the eigenvector that corresponds to the second smallest eigenvalue should be employed.

4. SPECTRAL LEARNING BASED ON FEATURE-SIMILARITY MATRICES

4.1. Spectral k-Means

It can be observed that both Spectral Clustering and Spectral k-Means are based on *object* × *object* matrices. More precisely, Spectral k-Means is based on the object inner-product similarity matrix XX^T and (normalized) Spectral Clustering is based on the *object* × *object* "distance" matrix L. As the sample size grows, the sizes of these matrices change accordingly, thus enhardening the study of the asymptotic clustering behavior.

In order to facilitate the study of Spectral k-Means with growing sample sizes we make the observation that the algorithm's output can be derived by a feature-similarity matrix that remains constant in size as the sample size grows. More precisely, we can observe that if λ_i and u_i is an eigenvalue-eigenvector pair of the feature inner-product similarity matrix $X^T X$, then λ_i and $X u_i / ||X u_i||$, is an eigenvalue-eigenvector pair of the object inner-product similarity matrix. This observation illustrates that we can derive the cluster solutions by simply projecting the data matrix X onto the eigenvectors of the feature inner-product similarity matrix. This is a crucial observation and allows us to confine our study to the constant in size feature similarity matrix.

It can be observed that as the sample size grows, the objective function of k-Means becomes larger, not converging to a constant value. In order to address this issue, we consider the normalized feature inner-product similarity matrix $\frac{1}{n} X^T X$ (n is the number of objects), that produces exactly the same eigenvectors (and thus continuous solutions) as $X^T X$. As we will analyze in section 6, the factor $\frac{1}{n}$ can guarantee the convergence of the objective function in the context of the law of large numbers as $n \to \infty$. Based on these considerations we can state the Spectral k-Means optimization problem as:

$$max_Y(\mathbf{Tr}(Y^T[\frac{1}{n} X^T X]Y)) \tag{3}$$

The connection between Spectral K-Means and the spectrum of feature similarity matrices was also taken into account in the work of [Ding and He 2004], where the authors have demonstrated that by conducting an appropriate continuous relaxation to the original clustering problem, the solution of k-Means can be derived by the projections of the data on the *k-1* principal vectors. Their main result is summarized in the following theorem.

THEOREM 4.1 ([DING AND HE 2004]). *When optimizing the k-Means objective function, the continuous solution for the transformed discrete cluster membership indicator vectors*[4] *are given by* $(v_1, ..., v_{k-1})$*, where* $v_i = \frac{1}{\sqrt{\lambda_i}} X_c u_i$*. The* λ_i *and* u_i*,* $i = 1, ..., k-1$ *are the* $k-1$ *largest eigenvalues and the respective eigenvectors of the input covariance matrix, and* X_c *is the centered object* × *feature data matrix.*

It can be observed in the above theorem that the authors employ $k - 1$ (and not k) eigenvectors for solving the k-Means clustering problem, and also that they employ the feature covariance matrix (which can be considered as a centered inner-product similarity for the features) and not the feature inner-product similarity. The reason is that the relaxation is performed on a slightly different objective function.

$$\begin{aligned} &max_Y \mathbf{Tr}(Y^T X_c X_c^T Y) - \mathbf{Tr}(X_c X_c^T) \equiv \\ &max_Y \mathbf{Tr}(Y^T X_c X_c^T Y) \end{aligned} \tag{4}$$

where Y is an orthogonal $n \times (k-1)$ matrix and X_c is the centered data matrix. For details on the derivation, the interested reader can refer to [Ding and He 2004]. As it will become apparent in the subsequent sections, our methodological approach can be equally applied to both spectral k-Means formulations.

4.2. Normalized Spectral Clustering

Similar results can be derived for the normalized Laplacian, when the instance-similarity matrix W can be expressed as an inner product matrix at a fixed feature space. By using the word "fixed" we refer to a feature space that remains constant as the sample size grows, i.e. Gaussian and Polynomial Kernels do not fall into this category. Examples of valid W choices include the simple inner product $W = XX^T$ and other inner-product variations of W such as the normalized inner-product $W = XD_Y^{-1}X^T$ (with D_Y being a diagonal matrix and $D_Y(j, j) = \sum_i X(i, j)$).

In order to demonstrate that the Spectral Clustering results can be derived by a feature-similarity matrix when $W = XX^T$, we define the weighted feature-similarity matrix $TermSim$ as:

$$TermSim = (X^T D^{-1} X) \tag{5}$$

where D is the graph degree matrix as derived by matrix W. Now if we consider λ_i and u_i, $i = 1, ..., n$ to be the eigenvalues and the respective eigenvectors of the $TermSim$ matrix, then it can be easily shown that $1 - \lambda^i$ is an eigenvalue and $c \cdot (D^{-1/2}X)u^i$ the respective eigenvector of normalized Laplacian L, where c is a constant that guarantees that the norm of the eigenvector is equal to 1. Similar results can be derived for other inner-product versions of W.

Based on the above, we have established the direct connection between the eigenvectors of $TermSim$ and the eigenvectors of normalized Laplacian L. Taking into account this observation we can study the behavior of the clustering results as the sample size grows, using the fixed size feature similarity matrix $TermSim$.

5. QUALITY MEASURES: OBJECTIVE FUNCTION AND CLUSTER RESULTS

5.1. Objective Function

Recall that the Spectral k-Means and the Spectral Clustering optimization problems are stated as trace maximization problems (equations 2,3,4) and the dominant eigenvectors of the respective feature similarity matrices are employed for deriving the continuous cluster solutions. Thus, the appropriate objective function can be derived by the sum of the eigenvalues that correspond to eigenvectors employed for the (continuous) clustering solution. More precisely we can state the following three observations that are a direct consequence of a popular theorem of Ky Fan (theorem 3.2 in [Ding and He 2004]).

[4] The theorem refers to the continuous relaxation defined in [Ding and He 2004]

OBSERVATION 1 (SPECTRAL k-MEANS, BASED ON EQ. 3). *Given an input object-feature data matrix X, the objective function for the continuous relaxation of optimization problem 3 is derived by $\sum_{i=1}^{k} \lambda_i$, where λ_i, $i = 1, ..., k$ are the k dominant eigenvalues of the feature-similarity matrix $X^T X$.*

OBSERVATION 2 (SPECTRAL k-MEANS, BASED ON EQ. 4). *Given an input object-feature data matrix X, the objective function for the continuous relaxation of optimization problem 4 is derived by $\sum_{i=1}^{k-1} \lambda_i$, where λ_i, $i = 1, ..., k-1$ are the $k-1$ dominant eigenvalues of the feature-covariance matrix.*

OBSERVATION 3 (SPECTRAL CLUSTERING, BASED ON EQ. 2). *Given an input object-feature data matrix X, and $W = XX^T$, the objective function for the continuous relaxation of the optimization problem 2 for $k > 2$ clusters is derived by $\sum_{i=1}^{k} \lambda_i$, where λ_i, $i = 1, ..., k$ are the k dominant eigenvalues of the $TermSim = X^T D^{-1} X$ matrix. When $k = 2$ the objective is derived by λ_2, where λ_2 is the second largest eigenvalue of matrix $TermSim$.*

Based on the above, it is evident that in order to measure the asymptotic (infinite-limit data) approximation level for the objective function of Spectral k-Means and Normalized Spectral Clustering, one should measure the proximity of the sample-based eigenvalues to the asymptotic ones. Statistics provides us with a formal framework for studying the proximity of the sample-based estimates to their expected value results. More precisely, based on statistical accuracy and asymptotic analysis we can derive that a sample size is sufficient for producing adequately accurate estimations with high confidence i.e. if we draw different samples (of the same size) from the data generating distribution the approximation requirement will hold with high probability (i.e. in 95% of experiments). This is an extensively studied issue in the statistical literature, and depending on the assumptions that one can make concerning the data generating distribution, there exist several approaches for deriving the statistical accuracy and asymptotic properties of the sample eigenvalues (a literature review for certain types of random matrices can be found in [Bai 1999]).

5.2. Clustering Results

In the afore subsection, we have demonstrated that the objective function approximation can be cast as an eigenvalue estimation problem. Since, the (continuous) clustering results are derived by the appropriate eigenvectors, one can analogously consider the problem of measuring the asymptotic (infinite-limit data) approximation level of the clustering results as an eigenvector estimation problem. However, this a slightly harder problem than one needs to solve, since the continuous results actually depend on the space spanned by the employed eigenvectors, rather than the eigenvectors themselves. This is because the cluster results are derived by projecting the original data onto the estimated eigenvectors, thus any basis of the space that is spanned by these eigenvectors would suffice to produce *exactly the same distances between the projected objects*. The Euclidean distances between the projected data is employed by many authors for deriving the discrete cluster solutions ([von Luxburg 2007] and references therein), thus preserving the same distances in the projected space would suffice to produce the same clustering results.

In order to illustrate the problematic nature of studying the behavior of eigenvectors, consider a feature similarity matrix that converges (asymptotically) to a matrix whose largest eigenvalue λ has algebraic multiplicity 2 (i.e. the largest eigenvalue corresponds to two eigenvectors u,u'). It can be easily observed that any basis of the space that is spanned by u and u', produces a valid pair of eigenvectors that correspond to eigenvalue λ. Although the eigenvectors in this case are highly unstable, the projection of a data matrix to their eigenspace produces constant distances, independent of the basis chosen for the projection. This example illustrates that there can be cases where the eigenvectors do not converge to a stable solution, while the eigenspaces exhibit a coherent behavior.

This observation allows us to cast the problem of approximating the asymptotic cluster results as a statistical-estimation problem of the appropriate eigenspace (i.e. the space spanned by the eigenvectors that take part in the clustering solution). Based on the analysis presented we can elaborate on the original research goals.

Original Goal:

- Find a sufficiently large sub-sample of the dataset such that the desired approximation thresholds are achieved with high confidence.

Equivalent Goals:

- Find a sufficiently large sub-sample of the dataset such that the appropriate sample-based eigenvalues approximate the asymptotic eigenvalues with high confidence.
- Find a sufficiently large sub-sample of the dataset such that the appropriate sample-based eigenspace approximates the asymptotic eigenspace with high confidence.

6. APPROXIMATING THE ASYMPTOTIC RESULTS

6.1. Objective Function

We will now present the proposed methodological approach for deriving the sample-based approximations to the asymptotic Objective function, which as analyzed in the previous section, can be cast as an eigenvalue estimation problem. The issue of statistical estimation of sample eigenvalues has been extensively studied and several methodological approaches and algorithms have been proposed. However, in the context of this work we assume that we do not have any knowledge of the data generating distribution, thus severely reducing the range of methods that can be employed.

A popular approach for measuring the accuracy of statistical estimates without making unnecessary distributional assumptions is Bootstrapping [Efron and Tibshirani 1993]. Given a random sample, generated by an unknown probability distribution and a statistic of interest, the bootstrapping procedure generates several independent bootstrap samples by sampling with replacement and consequently computes the appropriate standard errors and confidence intervals based on the variation exhibited by the statistic of interest. The theoretical justification for Bootstrapping relies on the Glivenko-Cantelli theorem (can be found in [Chung 1974]), that in the context of an iid sample, asserts that the empirical distribution, as derived by sampling with replacement, converges uniformly with probability 1 to the unknown data generating distribution. Moreover, a smoothness condition on the function used for estimating the statistic is required such that the convergence to the asymptotic results is guaranteed.

Bootstrapping has been previously used for computing the statistical accuracy of eigenvalues [Efron and Tibshirani 1993]. However, it is evident that the Bootstrapping approach would impose a significant computational overhead as it would require the computation of the eigenvalue-decomposition multiple times (1000-2000 bootstrap samples are commonly required for constructing confidence intervals). In order to address this issue, we employ Matrix Perturbation Theory that allows us to relate the statistical accuracy of the elements of a matrix to its eigenvalues.

THEOREM 6.1 (WEYL'S THEOREM [STEWART AND SUN 1990]). *Let A be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ and E a symmetric perturbation with eigenvalues $\epsilon_1 \geq \epsilon_2 \geq ... \geq \epsilon_n$. Then for $i = 1, ..., n$ the eigenvalues $\overline{\lambda}_i$ of $A + E$ will lie in the interval $[\lambda_i + \epsilon_n, \lambda_i + \epsilon_1]$.*

In the context of this work A is the appropriate feature similarity matrix that is used for computing the objective function. Weyl's theorem allows us to initially evaluate the statistical accuracy of the elements of the input matrix (as encoded by error-matrix E), and consequently assess the effect of error matrix E to the eigenvalues. In order to compute matrix E, we can consider the task of estimating the statistical accuracy of all the feature-similarity pairs (i.e. all the elements of the feature-similarity matrix), by means of confidence intervals. Recall that confidence intervals present us with a standard approach for determining the range of values a statistic of interest will assume around its expected value, with high confidence. Having computed the confidence intervals, we can define E in the same manner as in [Mavroeidis and Vazirgiannis 2007; Mavroeidis and Bingham 2008; 2010], i.e. as the maximum difference between the feature similarities and the endpoints of the corresponding confidence interval. Thus, high values for elements of matrix E will correspond to wide confidence intervals, while small values will correspond to highly accurate estimates. Having defined E, we can employ Weyl's theorem and directly assess the effect of the (in)-accuracy of the input matrix elements to its eigenvalues. The size of the eigenvalues of matrix E will determine the upper bound on the objective function estimation. It is evident that this process avoids the multiple eigenvalue computations, thus significantly reducing the computational overhead imposed by the bootstrap process.

6.2. Clustering Results

We will now present the proposed methodological approach for deriving the sample-based approximations to the asymptotic Clustering results, which as analyzed in the previous section, can be cast as an eigenspace estimation problem. It can be observed that in order to measure the approximation to the asymptotic eigenspace, the definition of a distance measure between subspaces is required. In the context of this work we employ the norm-difference between the respective projection operators that is a popular measure for evaluating the distance between subspaces. As in the case of eigenvalues, bootstrapping directly the eigenspaces would impose a significant computational overhead since it would require the computation of the eigenvector-decomposition multiple times. In order to address this issue, we employ Strewart's theorem on the perturbation of invariant subspaces. The subsequent theorem presents a slightly modified version of the original Stewart's theorem, as presented in [Papadimitriou et al. 1998]:

THEOREM 6.2 (STEWART'S THEOREM [STEWART AND SUN 1990]). *Let A and $A+E$ be $n\times n$ symmetric matrices and let $V = [V_1\ V_2]$ be an orthogonal matrix, with $V_1 \in d \times n$ and $V_2 \in (n-d) \times n$, where range(V_1) is an invariant subspace for A. Partition the matrices V^TAV and V^TEV as follows:*

$$V^TAV = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

$$V^TEV = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}$$

if

$$\delta = \lambda_{min} - \mu_{max} - ||E_{11}||_2 - ||E_{22}||_2 > 0$$

where λ_{min} is the smallest eigenvalue of Q_1 and μ_{max} is the largest eigenvalue of Q_2 and $||E_{12}||_2 \leq \delta/2$, then there exists a matrix $P \in (n-d) \times d$ with $||P||_2 \leq \frac{2}{\delta}||E_{21}||_2$, such that the columns of $V_1' = (V_1 + V_2P)(I + P^TP)^{\frac{1}{2}}$ form an orthonormal space that is invariant for $A + E$. Moreover, concerning the distance between the projection operators corresponding to V_1 and V_1' we have that

$$||P_{V_1} - P_{V_1'}||_2 \leq \frac{2}{\delta}||E_{21}||_2$$

Given matrices A and E, Stewart's upper bound requires the computation of: V_1, V_2, λ_{min} μ_{max}, E_{11}, E_{22} and E_{21}. In Spectral k-means the solution is derived by the k dominant eigenvectors of the input matrix, thus V_1 is defined by the top-k eigenvectors of A (as columns) and V_2 is defined by the rest $n-k$ eigenvectors. Based on these definitions for V_1 and V_2 we will have that $\lambda_{min} = \lambda_k$, i.e. the k^{th} largest eigenvalue of matrix A and $\mu_{max} = \lambda_{k+1}$, i.e. the $k + 1$ largest eigenvalue of matrix A. The above specifications clarify how Stewart's upper bound can be computed given the input matrices A and E.

We will also derive here two simplified expressions of Stewart's bound. One simplified bound includes only terms λ_k, λ_{k+1} and E_{21}, while the other includes solely λ_k, λ_{k+1} and E. In order to derive these bounds we need to make a stronger assumption for the size of the eigengap than the one employed in Stewart's theorem. The eigengap requirement in Stewart's theorem is expressed in the formula $\delta = \lambda_{min} - \mu_{max} - ||E_{11}||_2 - ||E_{22}||_2 > 0$, which translates in our context as: $\lambda_k - \lambda_{k+1} - ||E_{11}||_2 - ||E_{22}||_2 > 0$. If we now impose a stronger assumption for the size of the eigengap,

$\lambda_k - \lambda_{k+1} > 2(||E_{11}||_2 + ||E_{22}||_2)$ we can derive for the upper bound employed in Stewart's theorem that $\frac{2}{\delta}||E_{21}||_2 \leq \frac{4||E_{21}||_2}{\lambda_k-\lambda_{k+1}}$. Thus, $\frac{4||E_{21}||_2}{\lambda_k-\lambda_{k+1}}$ can serve as an upper bound to the continuous results. Notice that this bound can be computed using solely the appropriate E_{21} matrix and the eigenvalues of A. If we further observe that $||E_{21}||_2 \leq ||E||_2$ then we can derive that $\frac{4||E_{21}||_2}{\lambda_k-\lambda_{k+1}} \leq \frac{4||E||_2}{\lambda_k-\lambda_{k+1}}$, thus $\frac{4||E||_2}{\lambda_k-\lambda_{k+1}}$ can also serve as an upper bound to the continuous results.

Matrices A and E that are required as input for computing Stewart's upper bound are derived in a similar manner as in [Mavroeidis and Vazirgiannis 2007; Mavroeidis and Bingham 2008; 2010] using the following procedure:

- Employ Bootstrapping (of objects) and compute confidence intervals for the elements of the appropriate feature-similarity matrix S.
- Define perturbation matrix E such that $E(i, j)$ contains the maximum difference between the $S(i, j)$ and the endpoints of the respective confidence interval.
- Compute an upper bound on the difference of the eigenvalues between S and $S + E$ based on Weyl's theorem.
- Compute an upper bound on the difference of the eigenspaces between S and $S + E$ based on Stewart's theorem.

The efficiency of this procedure is based on Matrix Perturbation Theory results that allows us to perform the bootstrap process on the elements of the appropriate feature-similarity matrix and consequently measure the effect of the variability of the matrix elements to the matrix's spectrum. Thus, this method does not require the computation of the eigen-decomposition of S as opposed to the naive application of Bootstrapping that would require 1000-2000 such computations. Although this approach provides us with an efficient Bootstrap-based proximity estimation of the sample-based spectrum to its expectation, it can be argued that it is not practically efficient in the cases where a large number of features is used. This is because the sequential sampling procedure would require the bootstrap-estimation of all the feature-to-feature confidence intervals multiple times until convergence. In Section 7.1 we address this issue and enhance the efficiency of this procedure by demonstrating that the desired bound can be derived by computing at each sequential step $k \cdot m$ confidence intervals (k is the number of clusters and m the number of features), instead of m^2 that are computed by the aforementioned approach.

It is evident that the afore approximation bounds are derived for the continuous cluster results, thus it is natural to inquire as to whether these bounds extend to the discrete cluster solutions. Based on the favorable empirical performance of Spectral Clustering and Spectral k-means, also reported in the experimental section of this paper, it can be argued that in practice the spectral clustering output can serve as a good approximation to the discrete cluster results. From the theoretical point of view, recent results [Huang et al. 2009] have demonstrated that under certain assumption the norm-distance between the continuous solutions can serve as an upper bound for the difference in the discrete cluster results. Thus, there exist empirical and theoretical evidence, that justify the use continuous bounds for measuring the proximity between two spectral solutions.

7. THE SEQUENTIAL SAMPLING ALGORITHM

7.1. The Algorithm

We will now formally define the sequential sampling process that terminates when the theoretical analysis, as described in the previous section, guarantees that the required approximation levels are reached. This procedure consists of two components, one accounting for the sequential sampling process and the other accounting for the efficient computation of the Bootstrap confidence intervals. In the first component there are several practical issues that need to be resolved. One such issue is related to the sequential sampling scheduling, i.e. the determination of the initial sample size as well as the specification of the increase of the sample size at each sequential step. Several approaches have been proposed in the relevant literature for addressing these issues [Guha et al. 1998; Banerjee and Ghosh 2002; Domingos and Hulten 2001; Provost et al. 1999]. These approaches employ the popular Hoeffding and Chernoff inequalities for determining the initial (or directly the required) sample size and also consider sophisticated sampling strategies, such as the geometric increase of the sample size at each sequential step.

In the context of this work, we do not utilize Hoeffding or Chernoff type bounds for estimating the initial (or required) sample size since these are worst-case bounds and commonly overestimate the required sample size. Moreover, with respect to the sampling scheduling mechanism, we consider a simple sampling procedure that enlarges linearly the sample size until the convergence criteria are met. In relevant approaches, there exist more sophisticated sampling scheduling mechanisms, such as geometrical sampling [Provost et al. 1999], however the empirical evidence in the experiments section suggests that linear sampling suffices to achieve a quick converge to the asymptotic results.

The sequential sampling algorithm is illustrated in Algorithm 1, while the Bootstrap-based accuracy estimation process, described in the previous section, is summarized in Algorithm 2. We should stress here that there exist various Bootstrap procedures for generating the desired confidence intervals[Efron and Tibshirani 1993] but this choice does not affect the general intuitions of the proposed approach.

With regards to the time complexity of Algorithm 1, each sequential sampling step requires $O(m^3 + m^2 \cdot n)$ time, where m is the number of features and n is the number of objects. The m^3 factor refers to the required eigendecomposition for computing the bounds based on Weyl and Stewart Theorems, while $m^2 \cdot n$ refers to the computation of the feature similarity matrix and the cost of constructing the confidence intervals. It is evident that as soon as the number of objects becomes larger than the number of features, i.e. $n > m$, the component that dominates the time complexity is $m^2 \cdot n$. A hidden computational burden that is not apparent in the O-based analysis is related to the estimation of the bootstrap confidence intervals. This is because it is generally accepted that 1000-2000 bootstrap samples are required for computing reliable confidence intervals[Efron and Tibshirani 1993]. Thus, the bootstrap process requires $B = 1000 - 2000$ estimations of the feature similarity matrix, resulting in a total of $B \cdot m^2 \cdot n$ such computations. It should be noted that the time complexity of both Bias Corrected and accelerated (BCa) confidence intervals as well as percentile intervals (that are consecutively employed in the experiments) is dominated by the multiple computations of the feature similarities. For details and the appropriate formulas for computing these confidence intervals the interested reader can refer to [Efron and Tibshirani 1993].

It can be observed that a large portion of the computational burden for deriving Stewart's upper bound is associated with the Bootstrap confidence intervals. This is due to the fact that bootstrap confidence intervals have to be computed for all feature-to-feature similarities repeatedly (1000-2000 times at each sequential sampling step). Thus, it would be desirable if one could avoid this burden and compute solely an "informative" subset of these confidence intervals. This potential arises if we observe that one of the simplified bounds derived in the previous paragraph, $\frac{4\|E_{21}\|_2}{\lambda_k - \lambda_{k+1}}$ employs matrix E_{21} which is smaller in size matrix than E. The use of this bound does not attain any direct advantages since E_{21} is a submatrix of $V^T EV$ and not E, thus it requires the prior computation of the full E matrix. However, as we will demonstrate subsequently, the norm of E_{21} can be approximated using a submatrix of E, thus requiring in the computation of solely a subset of the feature-feature similarities.

In order to achieve this goal we will firstly make some observations regarding matrices E and $V^T EV$. In the context of our work V contains as columns the eigenvectors of the appropriate feature-similarity matrix, thus the matrices E and $V^T EV$ are similar, in the linear algebra sense, i.e. they have exactly the same eigenvalues and thus also the same Frobenius and spectral norms. Moreover, Gerschgorin's theorem [Stewart and Sun 1990] (commonly referred to as Gerschgorin's disks) asserts that each eigenvalue λ of E and $V^T EV$ is bounded by the sum of the absolute value of the elements of a certain line (or column), i.e. $|\lambda| \leq \sum_i |a_{ij}|$. Based on these three observations, i.e. eigenvalue equality, Frobenius norm equality (which also implies the equality of the element-wise squared sums), and also Gerschgorin's theorem, we can assert that the submatrices of E and $V^T EV$

will have a similar structure and $V^T EV$ will not tend to overconcentrate the values of E in certain submatrices.

Based on the above observations, we consider the use of an appropriate submatrix of the original E matrix and not of the transformed $V^T EV$ for estimating the desired bound. This would enhance the efficiency of the bootstrap process, since we would avoid computing all the m^2 feature-similarity confidence intervals, and concentrate solely on $m \times (k - 1)$ intervals. One issue that is immediately raised is concerned with the choice of the appropriate E_{21} submatrix (of the original E matrix). Since each row and column of E corresponds to a specific feature, this question is essentially related to the selection of the appropriate features that will be utilized in the computation of E_{21}. In the context of this work we employ the $(k - 1)$ features that exhibit the highest variance. The justification of this choice is based on the relationship between the sampling variance of a covariance estimate and the variance of the individual features, that asserts that features with high variance are expected to have larger confidence intervals. Thus, selecting the $k - 1$ features with the highest variance follows the general intuition of selecting a worst-case submatrix of the original E matrix.

The proposed efficient computation of Stewart's bound is illustrated in Algorithm 3. It should be noted that Algorithm 3 does not compute the full E matrix, thus it cannot derive Weyl's upper bound on the objective function. Albeit the theoretical justifications of the efficient Bootstrap-based computation of the cluster approximation bounds, it is easy to construct counter examples where the appropriate submatrix of the original E matrix will underestimate the norm of the respective submatrix of $V^T EV$. This illustrates the need for extensive experimental evaluation. The empirical results, presented in Section 9 demonstrate that the proposed efficient approach (illustrated in Algorithm 3) can provide us with reliable estimates of the convergence to the asymptotic cluster results and also enhance substantially the efficiency of the sequential sampling framework.

ALGORITHM 1: Sequential Sampling Spectral *k*-Means

```
Input:
Training data $D$ generated by unknown probability distribution.
Required Approximation Level for Objective Function $ThresObj$.
Required Approximation Level for Cluster Results $ThresClust$.
Cardinality of sequential sampling step $c$.
Algorithm:
Generate a random sequence of sub-samples $\{d_1, d_2, ..., d_n\}$, with $d_i \subseteq D$ and $\#d_i = c$
$step = 0; dataset = \emptyset$
repeat
    $step \leftarrow step + 1$
    $dataset \leftarrow dataset \cup d_{step}$
    converged=(Efficient)BootCheck(dataset,ThresObj,ThresClust)
until ( Coverged==True OR step==n )
```

7.2. Factors that affect Convergence

Since the sequential sampling process terminates when the approximation requirements are met, a question that naturally arises is related to the conditions under which termination is achievable (i.e. the algorithm does converge as the sample size grows). In this section we demonstrate that the algorithm converges both with respect to the objective function, as well as to the cluster results under quite general assumptions. It is also demonstrated that the convergence of cluster results is harder and depends on the existence of a cluster structure in the dataset under study. The existence of a clear cluster structure will result in fast convergence for the algorithm, while the absence of a cluster structure will result in slow convergence or even divergence.

ALGORITHM 2: BootCheck

```
Input:
Training data sample $d_i$.
Threshold for Objective Function $ThresObj$.
Threshold for Cluster Results $ThresClust$.
Algorithm:
Compute feature-similarity matrix $S$.
Compute Bootstrap confidence intervals for elements of $S$.
Compute error-perturbation matrix $E$.
Compute upper bound on the Objective Function based on Weyl's theorem.
Compute upper bound on Cluster Results based on Stewart's theorem.
if Thresholds are achieved then
    Return True.
else
    Return False.
end if
```

ALGORITHM 3: EfficientBootCheck

```
Input:
Training data sample $d_i$.
Threshold for Cluster Results $ThresClust$.
Algorithm:
Compute feature-similarity matrix $S$.
Compute the $k-1$ features with highest variance.
Compute Bootstrap confidence intervals for the $k-1$ features with all $m$ features.
Compute error-perturbation matrix $E_{21}$.
Compute upper bound on Cluster Results based on Stewart's theorem.
if Thresholds are achieved then
    Return True.
else
    Return False.
end if
```

7.2.1. Convergence of Objective Function. Recall that the upper bound that quantifies the divergence of the sample-based Objective Function is based on error Matrix E, as derived by the lengths of the feature-similarity confidence intervals. Large E value entries signify highly inaccurate similarity estimations, while small values indicate that the similarities have almost converged to their expectations. It is evident that if the feature-similarities indeed converge as the sample size grows, the lengths of the respective confidence intervals will become smaller at each sequential step eventually converging to zero. This means that the eigenvalues of E matrix will also be decreased in absolute value, until they also converge to zero. We can summarize these observations in the following corollary:

COROLLARY 7.1. *Algorithm 1 will achieve the input requirements related to the Objective Function with a finite data sample if the theoretical assumptions of Bootstrapping hold and the elements of the appropriate Feature-Similarity Matrix converge asymptotically to their "true values".*

Recall that in the case of optimization problem 3 we consider the matrix $\frac{1}{n}X^TX$ where X is the *object* × *feature*, and in the case of the optimization problem 4 the feature covariance matrix. Since the elements of both matrices are averaged by n, i.e. the number of objects, the convergence to the "true feature-similarities" is guaranteed under mild assumptions by the Law of Large Numbers. Similar considerations have to be taken into account for $TermSim$ in the case of normalized clustering. It should be noted that for complex W and $TermSim$ definitions we can use the sequential sampling

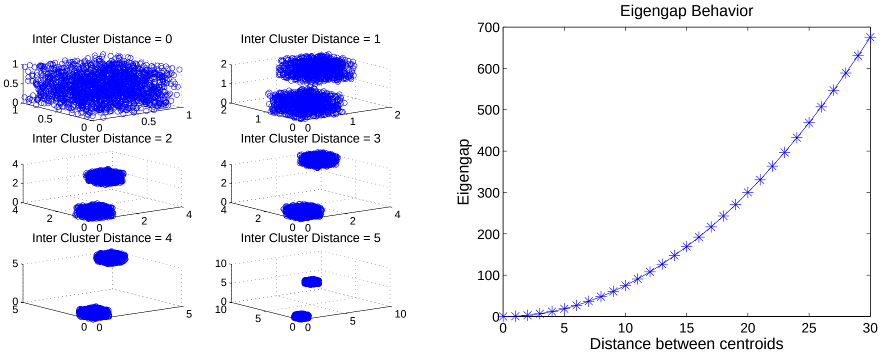


Fig. 1: Relevant Eigengap vs. Cluster centroid Distance

process to determine whether the Feature-similarities converge to their “true values”. An indicator for divergence would be that the values of the feature-similarities change as the sample size grows, while the sizes of the confidence intervals get smaller.

7.2.2. Convergence of Cluster Results. In an analogous manner we can state that the convergence of the eigenspaces (i.e. cluster results) is achieved when the elements of the feature-similarity matrix converge to their expectations and the *relevant eigengap* converges to a non-zero number. We use the term *relevant eigengap* to refer to the minimum difference between the eigenvalues employed in the spectral solution with the rest. Thus the *relevant eigengap* in the case of Spectral k-Means, as derived by optimization problem 3, is the difference between the k and the $k + 1$ eigenvalues; while in Spectral k-Means, as derived by optimization problem 4, the *relevant eigengap* is the difference between the $k - 1$ and the k eigenvalue (eigenvalues shorted in decreasing order). In order to understand why we need the relevant eigengap to converge to a positive number, one should observe that in the prerequisites of Stewart’s theorem, the relevant eigengap is required to be strictly larger than some expression of the norm of the error-perturbation matrix. Thus, if the eigengap is 0 then the prerequisites of Stewart’s theorem will not be satisfied for any error-perturbation matrix E. Moreover, when the eigengap is small, larger samples would be required such that the confidence intervals become small enough to satisfy the prerequisites of Stewart’s theorem.

The size of the *relevant eigengaps* also provides us with a measure of the cluster structure exhibited in the dataset. More precisely, if the dataset has dense and well separated clusters, then small perturbations will not affect the cluster structure. On the other hand, a Spectral Clustering solution with a large *relevant eigengap*, will also not be severely affected from small perturbations of the input (this is a direct derivation of Stewart’s theorem). Thus, if the Spectral Clustering algorithm indeed succeeds in identifying the correct cluster structure, then the size of the eigengaps can be employed as a heuristic for measuring the cluster structure exhibited in the dataset.

In order to demonstrate this behavior empirically we have considered a two-cluster scenario where the data is generated by a mixture of two Gaussians with prior $1/2$ each. It is evident that in this context the cluster structure depends on the distance between the two cluster centers. In Figure 1 we report the *relevant eigengap* of Spectral k-Means as the distance between the two clusters becomes larger. As expected, the enlargement of the cluster distances increases the relevant eigengap. We can summarize the discussion of this subsection in the following corollary:

COROLLARY 7.2. *Algorithm 1 will achieve the input requirements related to the cluster results with a finite data sample if the theoretical assumptions of Bootstrapping hold, the elements of the appropriate Feature-Similarity Matrix converge their “true values” and also if the relevant eigengap does not converge to 0.*

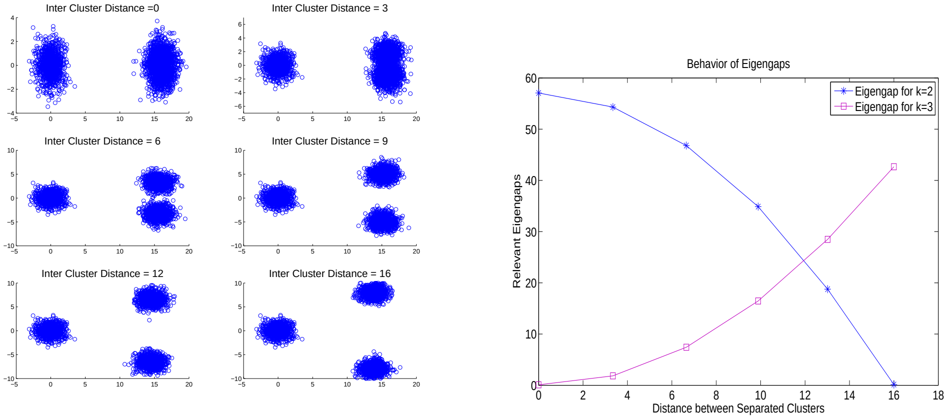


Fig. 2: Relevant Eigengap vs. Separated Cluster Distance

7.3. Number of Clusters and Sample Size/Bandwidth Requirements

In Stewart's theorem we can observe that a requirement for deriving a stable solution is that the relevant eigengap is larger than some expression of the norm of the E perturbation matrix. As we have analyzed earlier, the norm of E will be reduced as sample size becomes larger because of the increase in the accuracy of the feature-similarity estimates. Thus, one can consider selecting the appropriate k that maximizes the relevant eigengap, since a large relevant eigengap will require a smaller sample to converge to the asymptotic infinite-data solution. In the data mining literature the heuristic of selecting the number of clusters that maximizes the relevant eigengap has been employed by several authors (see [von Luxburg 2007] and references therein). These approaches are commonly justified based on perturbation theory or graph theoretic arguments. To the extend of our knowledge sample size arguments in the context of Spectral k-means, have not been employed in the discussion of this heuristic.

Based on the analysis of the sample size requirements, we can consider that the goal should not be to identify of the "correct" number of clusters, but rather to select among a set of plausible clusterings the one that is easier to model. In order to illustrate the notion of multiple plausible clusterings and their relation to the relevant eigengap, we provide the following example for Spectral k-means in Figure 2. In this example, we have generated three 3-dimensional gaussian clusters each containing 1000 objects, projected in 2-dimensional space for the needs of visualization. In the left part of Figure 2 we report the position change of the three clusters that shifts from an initially observable 2 cluster structure in the upper-left image (2 cluster centers initially overlap) to finally reach a clear 3 cluster structure in the bottom-right image. In the right side of Figure 2 we report the evolution of the relevant eigengap for $k = 2$ and $k = 3$. It can be observed that in the cases where the 2 clusters are very close to each other, the eigengap for $k = 2$ dominates. This illustrates that a smaller sample size is required for constructing a reliable 2-cluster model for the data. This clustering solution would group together the two clusters that are situated closely together. On the other hand, as the two clusters become well-separated the relevant eigengap for $k = 3$ dominates and a smaller sample size is required for modeling the three-cluster structure.

An interesting observation in Figure 2 is that the relevant eigengap for $k = 2$ not only becomes at some point smaller than the relevant eigengap for $k = 3$ but reaches almost a zero value. This can be justified if we observe that in the bottom right clustering, the cluster centers lie on the vertices of an equilateral triangle. Thus, setting $k = 2$ would force the Spectral k-means to group two of the three clusters together. However, due to the symmetrical positioning of the clusters, this grouping would be highly unstable as there does not exist a pair of clusters that are closer together and different samples will produce different groupings due to the small differences in each sample. Since a fully symmetrical dataset cannot be constructed (and is also highly unlikely to exist in practical applications), the 2-cluster solution would eventually converge to the asymptotic solution, but it would require a very large data sample.

8. RELATED WORK

We will now summarize the research work that is relevant to the proposed framework. The related work section is divided in two subsections: The first subsection presents the recent developments on the characterization of the asymptotic behavior of Spectral Clustering and also also summarizes the relevant sequential sampling approaches, while the second presents the relevant distributed PCA and *k*-Means approaches. Although the latter are not conceptually related to the proposed framework their summarization is required, since we compare against them in the experimental section.

8.1. Sequential Sampling and Asymptotic Behavior

Although, Spectral Clustering algorithms have received significant attention from data mining researchers, only recently has their asymptotic behavior been characterized [von Luxburg et al. 2008]. In the work of von Luxburg et al. the infinite-limit data behavior of Normalized and Unormalized Spectral Clustering is studied and the convergence requirements are analyzed. Interestingly it is derived that Normalized Spectral Clustering converges under more general conditions than unormalized spectral clustering, thus providing a theoretically justified preference for Normalized Spectral Clustering. To the extend of our knowledge there have been no attempts to define sequential sampling algorithms that aim in achieving a pre-defined approximation to the asymptotic behavior of Spectral *k*-Means. Such sequential sampling algorithms have been proposed for several other data mining paradigms [Domingos and Hulten 2001; Provost et al. 1999; Banerjee and Ghosh 2002].

Although not directly relevant, there exist several efficient sampling strategies for Lloyd's *k*-means in various application contexts (such as [Ailon et al. 2009; Datta et al. 2009; Zhou et al. 2007; Bradley et al. 1998]), that provide rigorous approximation guarantees to the clustering objective. A key difference of the proposed approach is that we take advantage of the “built-in” feature of Spectral *k*-means (and Spectral Clustering) that can provide, through the appropriate eigengap, an estimation of the relevant sample size requirements for all possible values of *k*. Naturally, we could consider patching Lloyd's *k*-means with a preprocessing step that selects (based on a certain objective) the appropriate number of clusters. However, this would impose an extra computational cost and we are not aware of any such approach that explicitly aims in specifying the number of clusters that can be reliably modeled with a small sample size. Moreover, we are not aware of any such approach that can quantitatively assess the relevant sample size requirements for all *k* values.

We should also clarify that there exists a vast bibliography on sampling and N*ÿ*strom approximation methods (such as [Fowlkes et al. 2004; Drineas et al. 1999]) that aim in approximating the a fixed size matrix and not the asymptotic infinite-data results. However, these approaches are conceptually different than the problem we address in this work.

8.2. Distributed k-Means and PCA

As we have stated in the introductory section, to the extend of our knowledge there do not exist any relevant Distributed Spectral *k*-means and Distributed Spectral Clustering algorithms. This highlights that a contribution of this work can be considered as the “distributalization” of an algorithm that has not been introduced in this application context. It should be noted though that with respect the popular Lloyd's *k*-means algorithm and other clustering algorithms there exists a large body of literature for a diverse range of distributed networks [Datta et al. 2009; Bandyopadhyay et al. 2006; Hammouda and Kamel 2007; Younis and Fahmy 2004; Zhang et al. 2008; Bandyopadhyay and Coyle 2003; Datta et al. 2006; Januzaj et al. 2004; Kargupta et al. 2000; Klusch et al. 2003; Kriegel et al. 2005].

As compared to these approaches, a key difference is that our framework is able to derive the relative bandwidth requirements for all possible values of *k*. This allows for the selection of the appropriate k-value that requires the minimal bandwidth. Naturally, one could consider employing as a preprocessing step the distributed selection of the appropriate number of clusters [Tasoulis and Vrahatis 2004]. However, the use of such algorithms would require the consumption of bandwidth and moreover, these are not specifically tailored for identifying the number of clusters that minimizes the required bandwidth consumption of a clustering algorithm.

In order to demonstrate the appropriateness of sampling in the distributed Clustering framework, we will compare our approach against certain Distributed Lloyd's type k-Means algorithms. These algorithms consider the task of computing the cluster structure of a dataset that is distributed among nodes in a network. That is, each node contains a fraction of the dataset and the goal is to approximate the full-data solution, while minimizing the amount of data that needs to be transmitted across the network. Since, to the extend of our knowledge, no Distributed Spectral Clustering algorithms have been proposed, we will compare against Lloyd's type Distributed k-Means approaches.

A prominent approach in this context was proposed by [Datta et al. 2006] (P2PKMeans). P2PKMeans is an adaptation of the classic k-Means algorithm especially designed for application in peer-to-peer networks. Each network node applies k-Means iteratively on its dataset and combines the resulting centroids with the centroids of other peers. The algorithm halts when all nodes have reached a stable state (i.e. the computed centroids in iteration i are the same as those of $i - 1$ or exhibit insignificant distortion).

Since we employ PCA to derive the continuous k-Means solution we will also refer to distributed PCA approaches. The intuition behind most distributed PCA approaches is based on the aggregation of a fragmented covariance matrix. A prominent distributed PCA approach is Collective PCA (CPCA [Kargupta et al. 2000]). In CPCA, each network node forwards a sample of its projected dataset together with its set of local eigenvectors to an aggregator node. Afterwards the aggregator combines the projected data from all sites and calculates the global eigenvectors. CPCA was also employed as an integral step of the distributed clustering methodology, described in [Kargupta et al. 2000]. CPCA requires $O((cf)^2 + \sum_{i=1}^{nodes} d_i k_i + skn)$ network load (c is the overall sample size, d_i the sample size of location i, k_i the number of principal components retained in site i and $f = \sum_{i=1}^{nodes} k_i$ the dimensionality of the aggregated array).

One significant drawback of CPCA is that it is only applicable in vertically distributed datasets. Global PCA (GPCA [Qi et al. 2004]) addresses this issue by providing a simple covariance aggregation scheme for the horizontal case. GPCA assumes centered data (i.e. mean=0), and derives that if u is an eigenvector of matrix $(m - 1)cov(X) + (p - 1)cov(Y)$, then u is also an eigenvector of $(m + p - 1)cov([X^T Y^T]^T)$ (here cov denotes the covariance matrix, X and Y are the data matrices contained in each peer and m and p the respective cardinalities). Based on this observation, each pair of peers can combine their eignevectors and define their locally global set of eigenvectors. By iteratively applying this procedure a network wide global set of eigenvectors can be defined and communicated to all nodes. GPCA requires $O((sn)^2 + skn)$ network resources, where s is the number of nodes, k the number of retained eigenvectors and n the number of dimensions.

It is evident that the aforementioned distributed PCA and k-Means approaches aim in approximating the full-data solution that is contained in a distributed network. Thus, their scalability depends crucially on the size of the network as well as the size of the data collection. Moreover, there are issues related to the required model updates for dynamic data. In the experimental section we will demonstrate that the proposed sampling-based approach can obtain high quality solutions with significantly lower bandwidth consumption. Since the proposed framework relies on Bootstrap confidence intervals we should also note that there exist several algorithms for accumulating uniform data samples in distributed networks with irregular degrees of connectivity and different data sizes (such as [Arai et al. 2007]).

9. EXPERIMENTS

In order to validate and assess the quality of our approach we have conducted a series of experiments on a set of large, real life and artificial datasets. The aim of this process is threefold:

(1) Demonstrate the convergent behavior and the efficiency of the Sequential Sampling Framework.
(2) Consider automated tuning strategies of input parameters.
(3) Show the virtues of sampling in a distributed setup, where restrictions are imposed on the amount of data that can be communicated.

In order to demonstrate the convergence behavior of the sequential sampling framework as the sample size grows, we used benchmark datasets that enjoy a clustered structure. As we have analyzed theoretically in section 7.2, if the datasets are clearly clustered, it is expected that the algorithm will converge rapidly to the required approximation levels. This behavior is indeed demonstrated in the experimental results of Section 9.3. In Section 9.3 we also report the execution time of our algorithm that empirically certifies the efficiency claims made earlier in this paper.

In order to illustrate the need for automatically tuning the input parameters, recall that the approximation requirement for the cluster results is provided by means of an upper bound on the difference between the respective projection operators. It is evident that this measure is related to the continuous results and does not provide us with a direct evaluation of the approximation to the asymptotic discrete cluster assignments. Thus, in subsection 9.4, we empirically assess how small this upper bound should be such that the clustering performance approximates sufficiently the discrete asymptotic cluster results. Interestingly, based on the derived parameter tuning process, it is demonstrated that datasets with millions of instances require solely a few thousand for converging to the asymptotic cluster results. This signifies that the consideration of larger data samples does not further improve the clustering performance. It can be observed that with the automatic tuning of the input parameter (as derived by subsection 9.4), our approach can be considered as a stand-alone algorithm that automatically determines the required sample size for approximating the asymptotic cluster results.

Based on the observation that our algorithm converges with solely a small fraction of the available data, we consider in subsection 9.5 the problem of Distributed Clustering. In this context it is commonly assumed that a large dataset is distributed among nodes in a network and the task is to derive a global data model (such as clustering) of the whole dataset. The naive approach would be to collect all the data centrally (to a network node), however this is usually not possible due to bandwidth limitations, that allow only a small fraction of the available data to be communicated. The imposed limitations make apparent the relevance of our approach to Distributed Clustering problems. As we have analyzed earlier in this paper, a distinct feature of the proposed framework as compared to the relevant distributed clustering approaches, is that is is able to estimate the relative sample size requirements for all possible values of k. However, in the experiments we consider the correct number of clusters as input and compare our framework against relevant approaches that attempt to approximate the full-data k-means model. The experiments demonstrate the superiority of our approach with respect to bandwidth consumption.

9.1. Datasets and Clustering Quality Measures

We have experimented with four real world and artificial datasets. Three of them were acquired from the UCI Machine Learning Repository [5] and one was acquired from the Large Scale Challenge that took place in ICML 2008 [6]. All of them contained a large number of instances, a feature that enabled us to highlight both the theoretic and practical merits of the proposed approach. The first dataset is the MAGIC Gamma Telescope Data Set that contains the simulated readings of a Gamma Telescope. The dataset contains two classes, one corresponding to normal and the other to noise readings. The second and third dataset were generated by the Waveform Database Generator. These datasets contain three classes corresponding to three types of waves. Based on the Waveform generator we have produced two noiseless dataset of 10000 and 1000000 instances respectively. The fourth

[5] http://archive.ics.uci.edu/ml/
[6] http://largescale.first.fraunhofer.de/about/

Table I: Datasets used in the evaluation

Dataset	Objects	Features	Classes	Description
Waveform	10000	21	3	Artificial dataset
Waveform1M	1000000	21	3	Artificial dataset
Magic	19020	10	2	Gamma telescope readings
delta	500000	500	2	Large Scale Challenge ICML'08

dataset is the delta dataset that was employed in the Large Scale Challenge. The datasets along with a brief description are summarized in Table I.

The assessment of *k*-means clustering results can be performed using various quality measures (such as F-measure, Mutual Information etc.). Although certain measures are generally regarded by practitioners as more appropriate, the issue of adapting the right measures for *k*-means clustering is still considered an interesting research topic, with a recent relevant publication appearing on ACM KDD 2009 [Wu et al. 2009]. In their conclusions, the authors of [Wu et al. 2009] identify the "best" measures that can successfully evaluate the clustering results even under "extreme" situations (such as imbalanced cluster sizes, lack of knowledge of correct number of clusters etc.). In our experimental setup all the datasets (with the exception of MAGIC dataset) have balanced cluster sizes and the correct number of clusters is provided as input. Based on this setup, we have employed the measures: Purity (P), Normalized Mutual Information (NMI) and F-measure (F_m) that are commonly used by researchers and practitioners for evaluating the *k*-means clustering performance.

Purity considers the mapping of a cluster($C_i, i = 1...k$) to a class ($S_j, i = 1...k$) based on the highest observed overlap. The quality of this assignment is measured by counting the number of correctly classified instances and dividing by the total number of instances (N). Normalized Mutual Information on the other hand attempts to quantify the information gain by the assignment of an instance belonging to class S_i to cluster C_j where $i, j = 1...k$. Finally, F-measure is the weighted harmonic mean of precision and recall. Table **??** provides an overview of these metrics. In the case of NMI, $I(C,S) = \sum_{i,j=1}^{k} \frac{|C_i \cap S_j|}{N} log \frac{N|C_i \cap S_j|}{|C_i||S_j|}$, $H(C) = -\sum_{i=1}^{k} \frac{|C_i|}{N} log \frac{|C_i|}{N}$ and $H(S) = -\sum_{i=1}^{k} \frac{|S_i|}{N} log \frac{|S_i|}{N}$. For F-measure,

$Recall = \frac{TruePositives}{TruePositives+FalseNegatives}$ and $Precision = \frac{TruePrositives}{TruePositives+FalsePositives}$.

Table II: Clustering Quality Evaluation Metrics

Abbreviation	Name	Definition
P	Purity	$\frac{1}{N} \sum_{i,j=1}^{k} max(\|C_i \cap S_j\|)$
NMI	Normalized Mutual Information	$\frac{2I(C,S)}{H(C)H(S)}$
F_m	F-measure	$\frac{2RecallPrecision}{Recall+Precision}$

Before we move on to analyze the empirical results of the proposed sequential sampling framework, we present some experiments that compare the performance of Spectral *k*-means (SKM), Spectral Clustering (SC) and Lloyd's *k*-means (KM). In these experimental results all three algorithms have almost identical performance on all four datasets we have employed.

9.2. Sampling Approximation Quality Measures

In order to assess the quality of the approximation of the sequential sampling process, we have defined a set of evaluation metrics. These measures are briefly outlined in Table IV.

The first quality measure we employ is the *Stability Factor* (*SF*). *SF* is based on the size of the *relevant eigengap* as well as the norm of the error-perturbation matrix E. Formally, concerning Spectral *k*-Means as derived by optimization problem 4, *SF* is defined as $\frac{\lambda_{k-1} - \lambda_k}{4 \cdot \|E\|_2}$, where $\lambda_{k-1} - \lambda_k$

Table III: Comparison of Loyd's k-Means (KM) with Spectral Clustering (SC) and Spectral k-Means (SKM).

(a) Comparison of KM, SC and SKM with Magic

	KM	SC	SKM
F_m	0.58	0.51	0.56
Pur	0.65	0.60	0.63
NMI	0.012	0.015	0.014

(b) Comparison of KM, SC and SKM with Waveform

	KM	SC	SKM
F_m	0.51	0.51	0.50
Pur	0.39	0.39	0.39
NMI	0.37	0.37	0.37

(c) Comparison of KM, SC and SKM with Waveform 1M

	KM	SC	SKM
F_m	0.50	0.50	0.50
Pur	0.39	0.39	0.39
NMI	0.37	0.37	0.37

(d) Comparison of KM, SC and SKM with Delta

	KM	DPCA/SKM
F_m	0.50	0.50
Pur	0.50	0.50
NMI	$4 * 10^{-5}$	$5 * 10^{-5}$

Table IV: Evaluations Metrics

Abbreviation	**Name**	**Definition**
SF_m	Mean Stability Factor	$SF_m = average[(\frac{relevant\ eigengap}{4 \cdot \|E\|_2})_i]$
ESF_m	Mean Efficient Stability Factor	$ESF_m = average[(\frac{relevant\ eigengap}{4 \cdot \|E_{21}\|_2})_i]$
F_m^{ub}	Objective Function's Mean Upper Bound	$F_m^{ub} = average[(\#eigs \cdot \lambda_1^E)_i]$
ΔPO_m	Projection Operators' Mean Difference	$\Delta PO_m = average[\|PO_i - PO_{fd}\|_2]$

is the relevant eigengap and E is the respective error-perturbation matrix. In the context of our experiments, in order to derive SF_m for a fixed sample size m, we draw 10 random sub-samples of size m and compute the average *Stability factor*. Formally, SF_m is defined as:

$$SF_m = average[(\frac{\lambda_{k-1} - \lambda_k}{4 \cdot \|E\|_2})_i]$$

where $(\frac{\lambda_{k-1}-\lambda_k}{4 \cdot \|E\|_2})_i$ denotes the SF as derived in the i^{th} sample. In order to understand the semantics of *SF* one should observe that the prerequisites of Stewart's theorem hold when $\frac{\lambda_{k-1}-\lambda_k}{4 \cdot \|E\|_2} > 1$ [Mavroeidis and Vazirgiannis 2007] and moreover, as the fraction becomes larger, the upper bound on the sample eigenspace becomes tighter. It should also be noted that this quantity has been employed by [Mavroeidis and Bingham 2008; 2010] to study the stability of eigenspaces.

In the experiments we have also employed the efficient version of the stability factor that is described in Section 7.1. Recall that the formula for computing the efficient stability factor is:

$$ESF_m = average[(\frac{relevant\ eigengap}{4 \cdot \|E_{21}\|_2})_i]$$

where E_{21} does not contain all the m^2 confidence interval lengths, but solely a subset of them thus enhancing the efficiency of the bootstrap-process.

Another metric, directly derived by Weyl's theorem, is the objective function's upper bound. The latter is defined as $F^{ub} = \#eigs \cdot (\lambda_1^E)$ where $\#eigs$ is the number of eigenvectors employed in the cluster solution and (λ_1^E) is the largest eigenvalue of the error-perturbation matrix E. F_m^{ub} is defined for a fixed sample size m and is also calculated as the average of multiple runs.

$$F_m^{ub} = average[\#eigs \cdot (\lambda_1^E)_i]$$

The difference of projection operators ΔPO computes the difference between the sample-based projection operator and the full-data solution. It is defined as $\Delta PO = ||PO_s - PO_{fd}||_2$ where P_s is the projection operator of the sample and P_{fd} is the projection operator of the full-data solution. The projection operator is defined by the eigenvectors employed in the cluster solution, i.e. in Spectral k-Means, based on equation 3, the projection operator is derived by VV^T, where the columns of V contain the k dominant eigenvectors of the respective feature similarity matrix. This metric aims in demonstrating the convergence of the projection operators to the full-data solution. Although, we have stated that the aim of this work is to guarantee converge to the asymptotic solution, the convergence to the full-data solution can be achieved as a by-product when convergence takes place for a sub-sample of the original dataset.

In order to derive the mean difference of projection operators ΔPO_m for a fixed sample size m, we draw 10 random sub-samples of size m, and compute for each the corresponding ΔPO value.

$$\Delta PO_m = average[||PO_i - PO_{fd}||_2]$$

where P_i is the projection operator in the i^{th} sample of size m and P_{fd} is the projection operator of the full-data solution.

9.3. Convergent Behavior and Efficiency

In order to study the convergence behavior of the proposed algorithm we present the evolution of the upper bounds on the objective function and the clustering results, i.e. SF_m ESF_m and F_m^{ub}. In Figure 3 we illustrate the evolution of SF_m of Spectral k-means with respect to the data sample. Sample size steps were configured to depict the rate of convergence of the clustering algorithm in each dataset. Consequently in the case of the Waveform datasets the sampling step was set to 200 instances and in the case of Magic 100 instances. With regards to SF_m we employ solely the three UCI datasets that possess a small number of features.

In Figure 7 we report the evolution of the efficient stability factor ESF_m of Spectral k-means with respect to the data sample for all four datasets. It can be observed that the behavior of ESF_m is similar to SF_m and the convergence rates are very similar. The efficiency enhancements of ESF_m are consecutively illustrated in Figure 8, where we report the total time that is required for a single step of the sequential sampling process. The time requirements are reported for an Intel Core 2 Duo, 2Ghz, 4GB RAM running Ubuntu 9.10.

In the Spectral k-Means experiments we relied on optimization problem 4. Moreover the appropriate error perturbation matrices E were derived by Bias Corrected and accelerated (BCa) confidence intervals [Efron and Tibshirani 1993], based on 1000 bootstrap samples. In Spectral Clustering the confidence intervals were derived by the percentile method using also 1000 bootstrap samples. Finally the coverage in all experiments was set to .95.

In Figure 4 we depict the evolution of F_m^{ub} of Spectral k-means with respect to the data sample size. We can observe that F_m^{ub} is influenced by the evolution of SF_m. More precisely, we notice the minimization of the objective function's mean upper bound when the stability factor value is maximized. In parallel, fluctuations, due to sampling variance, in the stability factor evolution are also observable in the behavior of F_m^{ub}. The same conclusions are drawn by observing the graphs derived by the application of our sequential sampling framework on Normalized Spectral Clustering. Figures 5,6 depict the corresponding results.

9.4. Automated Tuning of input parameters

Recall that the approximation requirement for the cluster results is provided by means of an upper bound for the difference between the respective projection operators. Although it has been demonstrated that under certain assumption a bound on the continuous results can be meaningful for bounding the difference in the discrete cluster assignments [Huang et al. 2009], in this section we will seek to verify this claim empirically. Moreover, we will explore how tight the continuous bound should be such that the asymptotic discrete cluster structure is sufficiently approximated. In all experiments we have observed that if we require that $SF_m = 1$, then the resulting cluster quality does not further improve when larger samples are considered. Intuitively, since the Stability Factor depends on both the coherence of cluster structure (as encoded by the relevant eigengap), as well as the approximation accuracy of the sample feature-similarities, we can derive that $SF_m = 1$ achieves the correct balance between the accuracy of the feature similarities and the cluster structure. A coherent cluster structure (large eigengap) requires less accurate feature-similarity estimations while in the absence of a clear cluster structure highly accurate feature-similarities must be derived. With respect to ESF_m, experiments illustrate that a value of $ESF_m = 2$ is required. In order observe this phenomenon we report in Figures 9, 10 the ΔPO_m measures for all four datasets for Spectral *k*-means and for the Magic and Waveform for Spectral Clustering.

The evolution of ΔPO_m and its relation to SF_m is depicted in all Figures, however it is more evident in the 9(c) when evaluated together with 3(c). The minimization rate of ΔPO_m is decreased as soon as SF_m exceeds 1 (when sample size reaches 2000) and continues to decrease at a constant rate as sample continues to grows. The same analysis holds for $ESF_m = 2$.

Additionally we measured the evolution of clustering quality throughout the sampling procedure. In each sampling step we used the derived projection operators and acquired the projection of the dataset on the corresponding space. Afterwards we discretized the solution using Lloyd's *k*-Means. Figure 11 presents the results for spectral *k*-means. The derived box-and-whisker plots highlight the variance in the clustering results. The red line highlights the maximum value exhibited in each sample iteration (the largest exhibited cluster quality value in the 10 iterations of sample size *m*), the green line the minimum value while the blue line the mean value. In the cases where we simply report the average values, the variance was negligible from the initial sample. In certain figures we can observe that initially the variance of clustering results is high and is decreased with the addition of more data samples, until $SF_m \geq 1$ (or $ESF_m \geq 2$) is satisfied. The latter is clearly demonstrated in the case of the Magic dataset. In all Figures it is depicted that $SF_m = 1$ can be considered as a sufficient condition for the convergence of the clustering quality. In the Waveform case it is shown that this is not a necessary condition since convergence is achieved even before $SF_m = 1$. The same experiments are also reported for Spectral Clustering in Figure 12. Again in this case we notice the same behavior that verify the validity of automatically tuning $SF_m = 1$ (or $ESF_m = 2$) as an input approximation requirement.

It can be observed that the quality of the converged cluster output for the delta dataset in terms of NMI is very low. This can be explained by the fact that even when the whole dataset is employed for deriving a cluster solution, the performance of all the algorithms considered in this paper remains very low (as illustrated in Table III). Thus, the low cluster quality can be attributed to the poor performance of *k*-means on the whole dataset. This can be verified if one inspects Figure 9(d). This Figure illustrates that the projection operator employed by the sequential sampling process quickly becomes very similar to the projection operator of the full data matrix. Thus, the low cluster quality can be attributed to the poor performance of *k*-means on the whole dataset.

9.5. Distributed Clustering

Based on the observation that our algorithm converges with solely a small fraction of the available data, we consider the problem of Distributed Clustering. In order to execute these experiments we assumed that a large dataset is distributed among the nodes of a peer-to-peer network and the task was to derive the global clustering model without communicating the whole dataset. Both Sequential Sampling Spectral *k*-Means (S^3KM) and Sequential Sampling Distributed Spectral Clustering (S^3DC) methods have been experimentally validated against distributed clustering and distributed PCA approaches that aim in approximating the full-data solutions. The methods (S^3KM) and (S^3DC) exploit the automated tuning methodology analyzed in the previous section and terminate the network sampling process as soon as they reach $SF_m = 1$. Moreover, in order to validate the utility the efficient stability factor ESF_m we have also experimented with terminating the sampling procedure as soon as $ESF_m = 2$. When the ESF_m termination criterion is used we will denote our algorithms as S^3_FKM and S^3_FDC.

(a) Clustering quality and network requirements for Magic. Network of 500 peers

	KM	SC	P2P KM	S^3KM	$S^3_F KM$	S^3DC	$S^3_F DC$	DPCA/SKM
F_m	0.58	0.51	0.59	0.60	0.60	0.55	0.54	0.56
Pur	0.65	0.60	0.62	0.63	0.63	0.60	0.53	0.63
NMI	0.012	0.015	0.015	0.015	0.015	0.015	0.014	0.014
NL_{MB}	N/A	N/A	1.71	0.5	**0.35**	1.48	1.00^s	0.42

(b) Clustering quality and network requirements for Waveform. Network of 500 peers

	KM	SC	P2P KM	S^3KM	$S^3_F KM$	S^3DC	$S^3_F DC$	DPCA/SKM
F_m	0.51	0.51	0.54	0.51	0.51	0.51	0.51	0.50
Pur	0.39	0.39	0.60	0.39	0.39	0.40	0.40	0.39
NMI	0.37	0.37	0.37	0.37	0.37	0.38	0.38	0.37
NL_{MB}	N/A	N/A	3.61	**0.09**	0.30	1.50^s	0.34	1.84

(c) Clustering quality and network requirements for Waveform 1M. Network of 5000 peers

	KM	SC	P2P KM	S^3KM	$S^3_F KM$	S^3DC	$S^3_F DC$	DPCA/SKM
F_m	0.50	0.50	0.53	0.51	0.51	0.51	0.50	0.50
Pur	0.39	0.39	0.60	0.40	0.40	0.41	0.39	0.39
NMI	0.37	0.37	0.37	0.37	0.37	0.38	0.37	0.37
NL_{MB}	N/A	N/A	508.7	**0.09**	1.81	3.35^s	1.76	20.14

(d) Clustering quality and network requirements for Delta. Network of 5000 peers

	KM	P2P KM	$S^3_F KM$	DPCA/SKM
F_m	0.50	0.51	0.50	0.50
Pur	0.50	0.50	0.50	0.50
NMI	$4 * 10^{-5}$	$3.5 * 10^{-5}$	$1.6 * 10^{-5}$	$5 * 10^{-5}$
NL_{MB}	N/A	$7.71 * 10^3$	**38**	$9.55 * 10^3$

Table V: Clustering quality and network requirements as obtained from the experiments. S^3KM corresponds to Sequential Sampling Spectral k-Means while S^3DC to Sequential Sampling Spectral k-Means. Subscript F identifies their fast vast version. Superscript s signifies that although the experiment was not concluded ($SF < 1$ or $ESF < 2$) the behavior of the sampling procedure indicated that approximately this value would appear.

As a first evaluation benchmark we used the clustering quality of the algorithms executed centrally on the whole datasets. Given the distributed nature of our approach we also evaluated our algorithms against P2PKMeans and GPCA. Unfortunately, CPCA is not directly comparable to our approach since it is specifically designed and tuned to address cases of vertically distributed datasets while we focus on the horizontal case.

All experiments took place in a simulated peer-to-peer environment where topology was randomly generated with nodes being connected with 5% probability. In the case of the two largest datasets we have created a network of 5000 nodes while for the two smaller sets we have used 500 nodes. It should be stressed out at this point that all the reported results are averaged over 10 executions. All algorithms, except from P2PKMeans, assume the existence of a star overlay network, where each peer communicates its sample (or result) to an aggregator node that undertakes the task of performing any subsequent computations. Finally, the aggregator node forwards the final result to all peers.

In Tables IV(a), IV(b),IV(c), IV(d) we present the results of all experiments. The Network Load is reported in Megabytes (NL_{MB}). Apart from the network requirements we report the cluster quality in terms of F-measure, Purity and NMI. It can be observed that GPCA provides results of equal quality to that of centralized *k*-Means while exhibiting low bandwidth requirements. S^3KM always produces the same clustering quality results but with significantly lower (in two out of three experiments) bandwidth consumption. It worths noting the fact that in the case of Waveform1M S^3KM requires a couple of KBs while GPCA requirements are in the order of MBs. Although S^3DC requires additional resources compared to S^3KM, it is still in an acceptable level, and in two out of three experiments requires less resources than GPCA. Despite its excellence in cluster performance, P2PKMeans exhibits excessively larger requirements in terms of network bandwidth. It is worth noting that although only centroids are communicated during the P2PKMeans execution, the exhibited network load marginally reaches the size of the dataset itself. In Tables IV(a), IV(b),IV(c), IV(d) bold values signify the minimum exhibited network load

We should stress here that in these experiments we have not evaluated the full extent of the capabilities of the proposed distributed spectral clustering framework. This is because we have provided all our algorithms with the correct number of clusters as input. As opposed to the relevant distributed *k*-means approaches, our algorithm would not have the danger of consuming a large bandwidth due to an inappropriate *k* input. This is achieved by its “built-in” ability to estimate the appropriateness of each *k* through the computation of the relevant eigengap. We should recall here that these arguments apply when the goal is to derive a good approximation of the cluster results and not when the target is to derive a good approximation of solely the objective function. As we have analyzed in Section 7.2 the former depends on the cluster structure of the dataset while the latter does not.

10. CONCLUSIONS AND FURTHER WORK

In conclusion, we have proposed a sequential sampling framework for Spectral *k*-Means that terminates when the algorithm’s output is indistinguishable from the asymptotic results. In order to formulate our approach we assume that the data is generated by an unknown probability distribution and consequently employ an efficient-bootstrap based methodology for assessing the convergence of the cluster results. Extensive experiments have demonstrated the convergent behavior of the proposed approach and also promote our approach as a viable solution to distributed clustering problems where bandwidth restrictions commonly impose limitations on the amount of data that can be communicated.

Concerning further work, we aim to extend the proposed approach to handle Kernel *k*-Means as well as Spectral Clustering based on Kernel object-similarities. Moreover, we will investigate the potential of defining sequential sampling Clustering algorithms for Time Series and stream data, where the dependence structure enhardens the application of bootstrapping.

REFERENCES

Ailon, N., Jaiswal, R., and Monteleoni, C. 2009. Streaming k-means approximation. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds. 10–18.

Arai, B., Lin, S., and Gunopulos, D. 2007. Efficient data sampling in heterogeneous peer-to-peer networks. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 23–32.

Awan, A., Ferreira, R. A., Jagannathan, S., and Grama, A. 2006. Distributed uniform sampling in unstructured peer-to-peer networks. In *HICSS ’06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, Washington, DC, USA.

Bai, Z. D. 1999. Methodologies in spectral analysis of large dimensional random matrices. *Statistica Sinica 9*, 611 677.

Bandyopadhyay, S. and Coyle, E. J. 2003. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*. INFOCOM ’03.

Bandyopadhyay, S., Giannella, C., Maulik, U., Kargupta, H., Liu, K., and Datta, S. 2006. Clustering distributed data streams in peer-to-peer environments. *Inf. Sci. 176,* 14, 1952–1985.

Banerjee, A. and Ghosh, J. 2002. On scaling up balanced clustering algorithms. In *Proceedings of the Second SIAM International Conference on Data Mining*. SDM '02. SIAM.

Bradley, P. S., Fayyad, U. M., and Reina, C. 1998. Scaling clustering algorithms to large databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. KDD '98. 9–15.

Chung, K. 1974. *A Course in Probability Theory*. Academic Press.

Datta, S., Giannella, C., and Kargupta, H. 2006. K-means clustering over a large, dynamic network. In *Proceedings of the Sixth SIAM International Conference on Data Mining*. SDM '06. SIAM.

Datta, S., Giannella, C. R., and Kargupta, H. 2009. Approximate distributed k-means clustering over a peer-to-peer network. *IEEE Transactions on Knowledge and Data Engineering 21,* 10, 1372–1388.

Dhillon, I. S. and Modha, D. S. 2000. A data-clustering algorithm on distributed memory multiprocessors. In *Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD*. Springer-Verlag, London, UK, 245–260.

Ding, C. and He, X. 2004. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*. ICML '04. ACM, New York, NY, USA.

Domingo, C., Gavaldà, R., and Watanabe, O. 2002. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery 6,* 131–152.

Domingos, P. and Hulten, G. 2001. A general method for scaling up machine learning algorithms and its application to clustering. In *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 106–113.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., and Vinay, V. 1999. Clustering in large graphs and matrices. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. SODA '99. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 291–299.

Efron, B. and Tibshirani, R. 1993. *An introduction to the bootstrap*. Chapman Hall.

Forman, G. and Zhang, B. 2000. Distributed data clustering can be efficient and exact. *SIGKDD Explor. Newsl. 2,* 2, 34–38.

Fowlkes, C., Belongie, S., Chung, F. R. K., and Malik, J. 2004. Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell. 26,* 2, 214–225.

Gordon, A. and Henderson, J. 1977. An algorithm for euclidean sum of squares classification. *Biometrics*.

Guha, S., Rastogi, R., and Shim, K. 1998. Cure: an efficient clustering algorithm for large databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. ACM, New York, NY, USA, 73–84.

Hammouda, K. M. and Kamel, M. S. 2007. Hp2pc: Scalable hierarchically-distributed peer-to-peer clustering. In *Proceedings of the Seventh SIAM International Conference on Data Mining*. SDM '07. SIAM.

Huang, L., Yan, D., Jordan, M. I., and Taft, N. 2009. Spectral clustering with perturbed data. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*. NIPS '08. MIT Press, 705–712.

Januzaj, E., Kriegel, H.-P., and Pfeifle, M. 2004. Scalable density-based distributed clustering. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*. PKDD '04. Springer-Verlag New York, Inc., New York, NY, USA, 231–244.

Kargupta, H., Huang, W., Sivakumar, K., Park, B.-H., and Wang, S. 2000. Collective principal component analysis from distributed, heterogeneous data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*. PKDD '00. Springer-Verlag, London, UK, 452–457.

Klusch, M., Lodi, S., and Moro, G. 2003. Distributed clustering based on sampling local density estimates. In *Proceedings of the 18th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 485–490.

Kriegel, H.-P., Kroger, P., Pryakhin, A., and Schubert, M. 2005. Effective and efficient distributed model-based clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*. ICDM '05. IEEE Computer Society, Washington, DC, USA, 258–265.

Lloyd, S. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory 28*, 129–137.

Mavroeidis, D. and Bingham, E. 2008. Enhancing the stability of spectral ordering with sparsification and partial supervision: Application to paleontological data. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, DC, USA, 462–471.

Mavroeidis, D. and Bingham, E. 2010. Enhancing the stability and efficiency of spectral ordering with partial supervision and feature selection. *Knowl. Inf. Syst. 23,* 2, 243–265.

Mavroeidis, D. and Vazirgiannis, M. 2007. Stability based sparse lsi/pca: Incorporating feature selection in lsi and pca. In *Machine Learning: ECML 2007, 18th European Conference on Machine Learning*. ECML '07. Springer, 226–237.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., and Vempala, S. 1998. Latent semantic indexing: a probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. PODS ’98. ACM, New York, NY, USA, 159–168.

Provost, F., Jensen, D., and Oates, T. 1999. Efficient progressive sampling. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD ’99. ACM, New York, NY, USA, 23–32.

Provost, F. J. and Kolluri, V. 1999. A survey of methods for scaling up inductive algorithms. *Data Min. Knowl. Discov. 3,* 2, 131–169.

Qi, H., Wang, T., and Birdwell, D. 2004. Global principal component analysis for dimensionality reduction in distributed data mining. *Chapter 19 in Statistical Data Mining and Knowledge Discovery, CRC Press*, 327–342.

Scheffer, T. and Wrobel, S. 2003. Finding the most interesting patterns in a database quickly by using sequential sampling. *J. Mach. Learn. Res. 3,* 833–862.

Scholz, M. 2005. Sampling-based sequential subgroup mining. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. KDD ’05. ACM, New York, NY, USA, 265–274.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22,* 8, 888–905.

Stewart, G. and Sun, J.-G. 1990. *Matrix perturbation theory*. Academic Press.

Tasoulis, D. K. and Vrahatis, M. N. 2004. Unsupervised distributed clustering. In *Parallel and Distributed Computing and Networks*, M. H. Hamza, Ed. IASTED/ACTA Press, 347–351.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing 17,* 4, 395–416.

von Luxburg, U., Belkin, M., and Bousquet, O. 2008. Consistency of spectral clustering. *Annals of Statistics 36,* 2, 555–586.

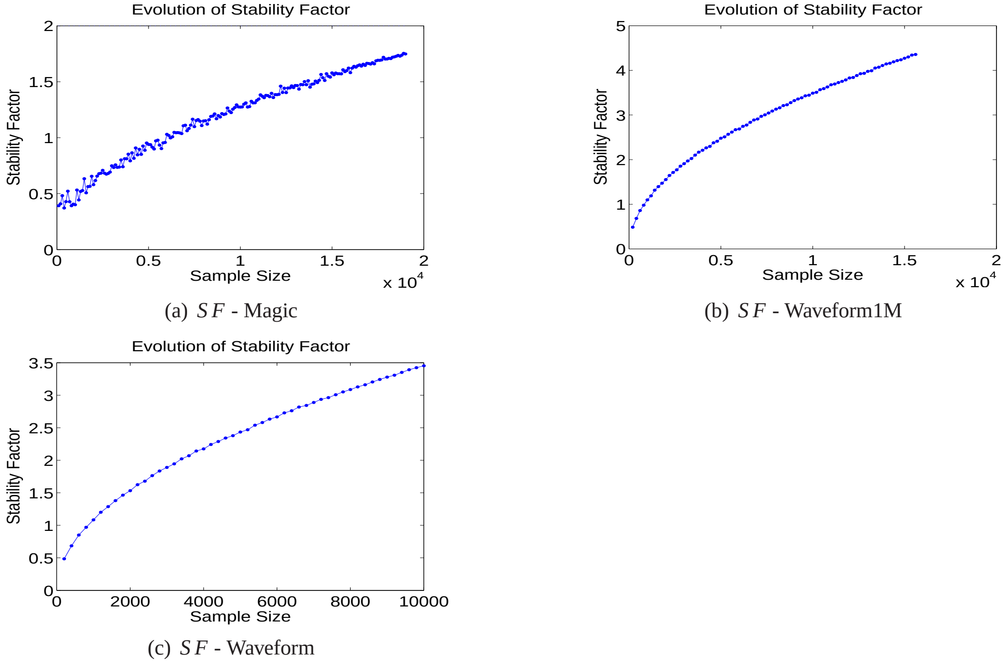
Wu, J., Xiong, H., and Chen, J. 2009. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD ’09. ACM, New York, NY, USA, 877–886.

Younis, O. and Fahmy, S. 2004. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*. INFOCOM ’04.

Zha, H., He, X., Ding, C. H. Q., Gu, M., and Simon, H. D. 2001. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14*. NIPS ’01. MIT Press, 1057–1064.

Zhang, Q., Liu, J., and Wang, W. 2008. Approximate clustering on distributed data streams. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*. IEEE Computer Society, Washington, DC, USA, 1131–1139.

Zhou, A., Cao, F., Yan, Y., Sha, C., and He, X. 2007. Distributed data stream clustering: A fast em-based approach. In *Proceedings of the 23rd International Conference on Data Engineering*. ICDE ’07. IEEE, 736–745.

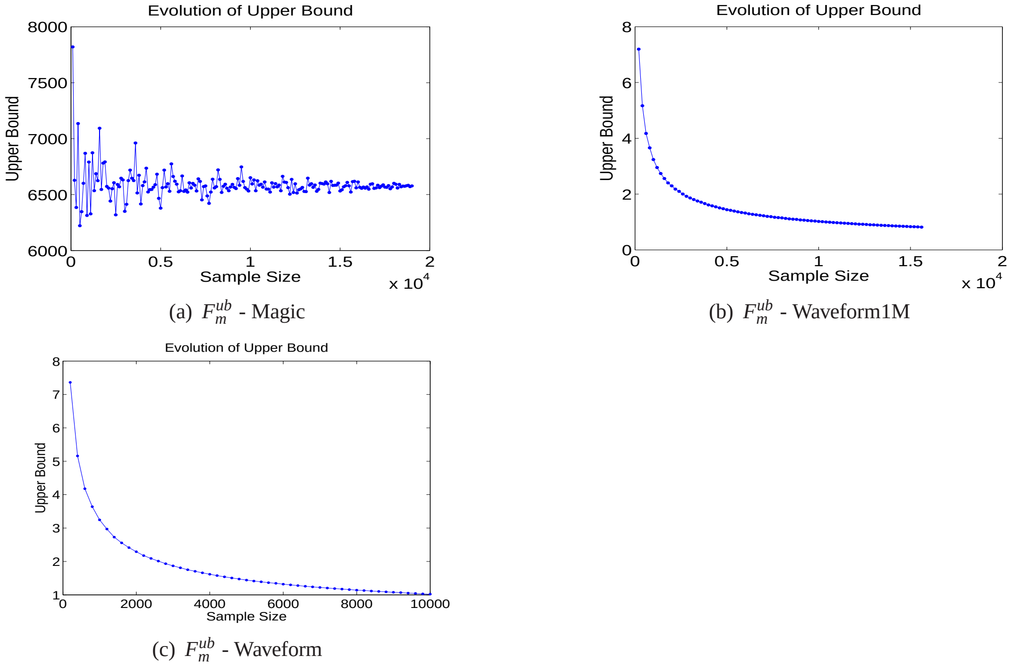


(a) SF - Magic

(b) SF - Waveform1M

(c) SF - Waveform

Fig. 3: Stability Factor Evolution

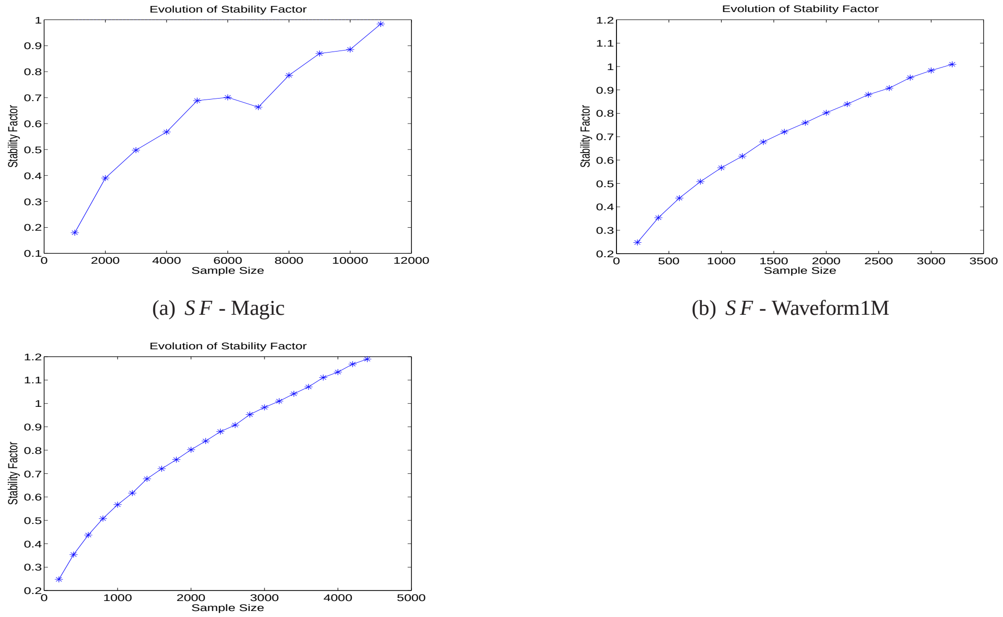


(a) F_m^{ub} - Magic

(b) F_m^{ub} - Waveform1M

(c) F_m^{ub} - Waveform

Fig. 4: Objective Function Upper Bound Evolution

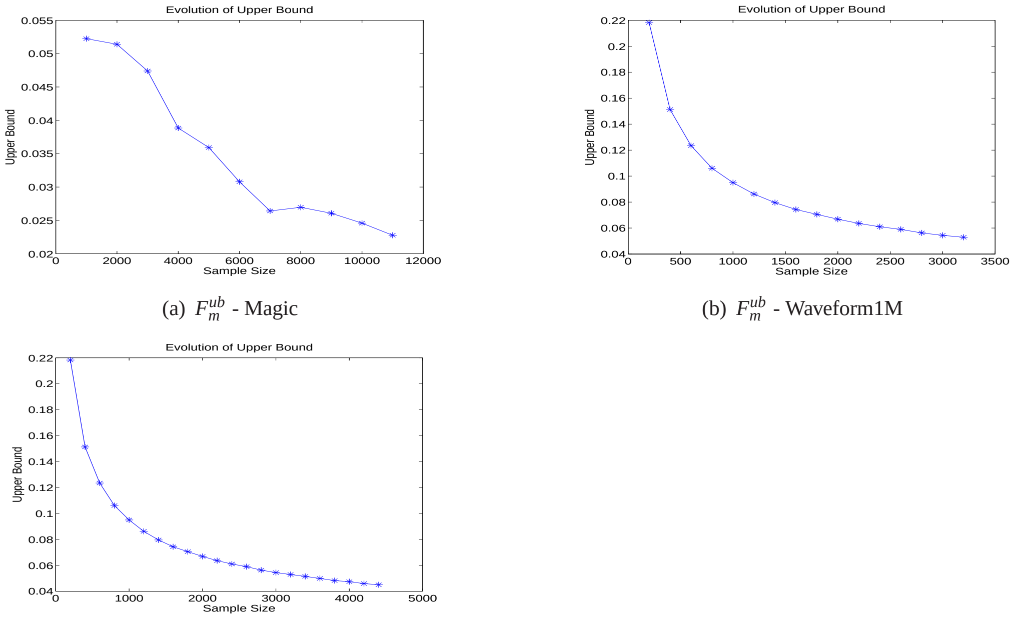


(a) SF - Magic

(b) SF - Waveform1M

(c) SF - Waveform

Fig. 5: Stability Factor Evolution - Spectral Clustering

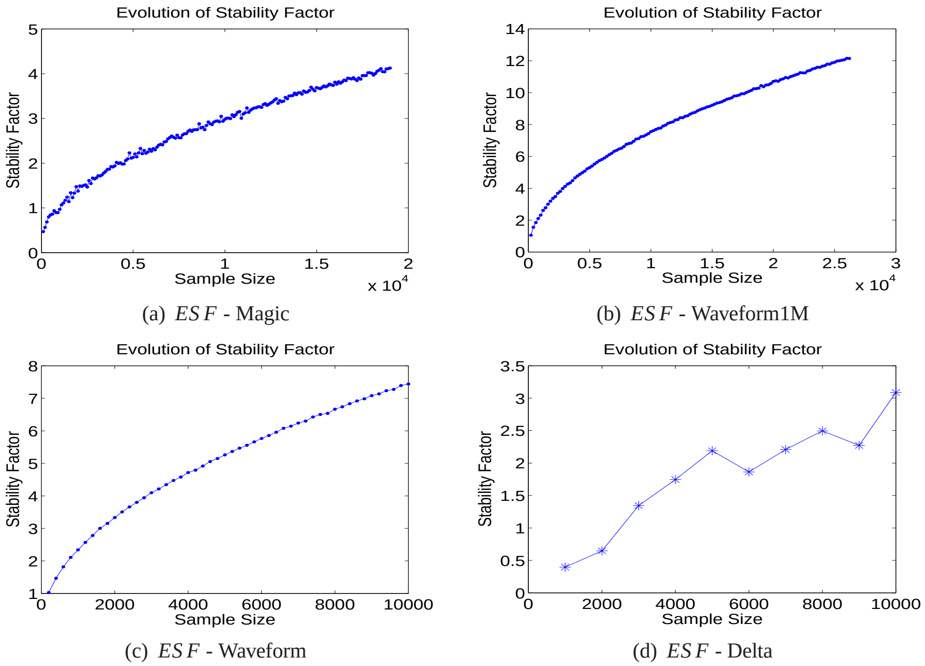


(a) F_m^{ub} - Magic

(b) F_m^{ub} - Waveform1M

(c) F_m^{ub} - Waveform

Fig. 6: Objective Function Upper Bound Evolution - Spectral Clustering



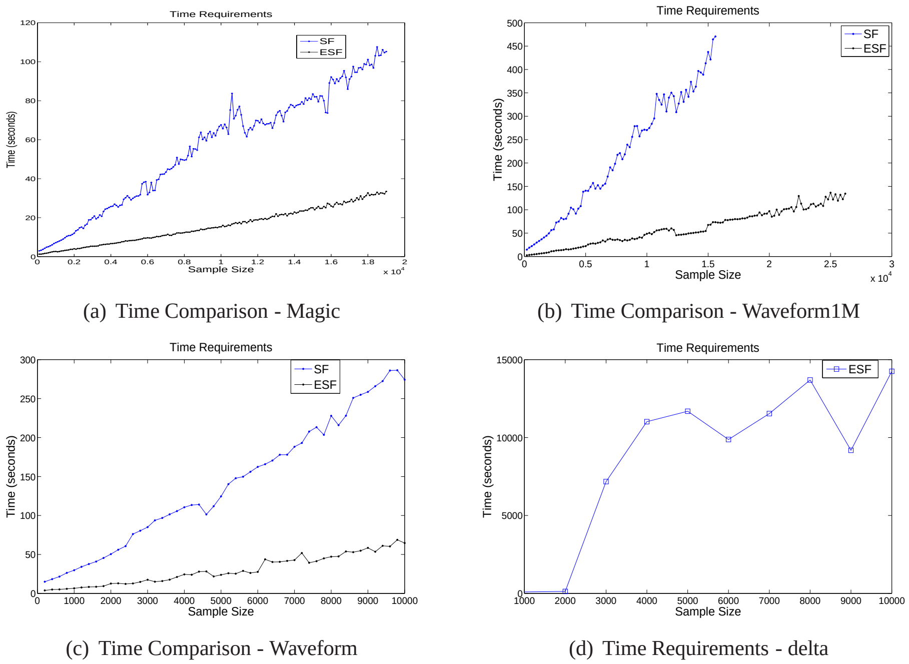
(a) *ESF* - Magic

(b) *ESF* - Waveform1M

(c) *ESF* - Waveform

(d) *ESF* - Delta

Fig. 7: Efficient Stability Factor Evolution



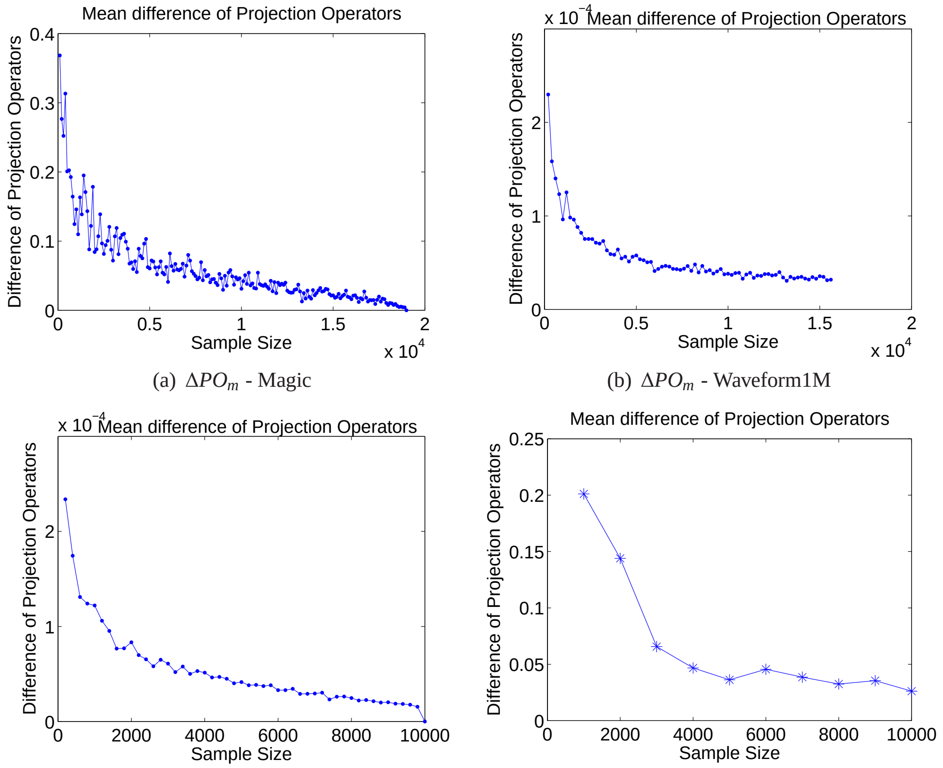
(a) Time Comparison - Magic

(b) Time Comparison - Waveform1M

(c) Time Comparison - Waveform

(d) Time Requirements - delta

Fig. 8: Time Required for one Sequential Sampling Step



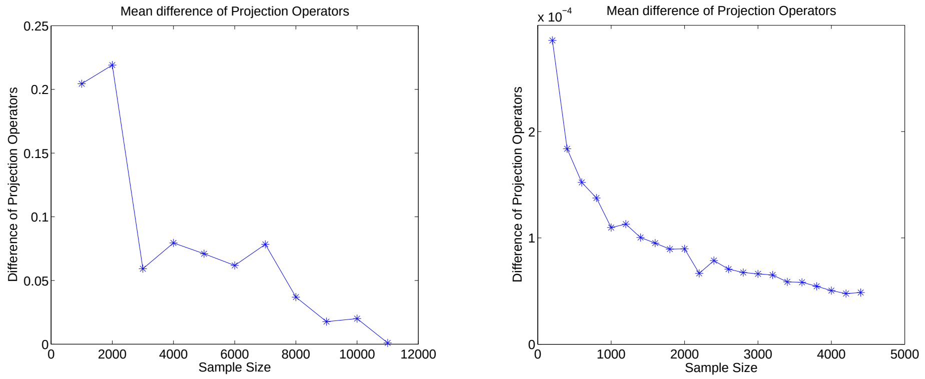
(a) ΔPO_m - Magic

(b) ΔPO_m - Waveform1M

(c) ΔPO_m - Waveform

(d) ΔPO_m - delta

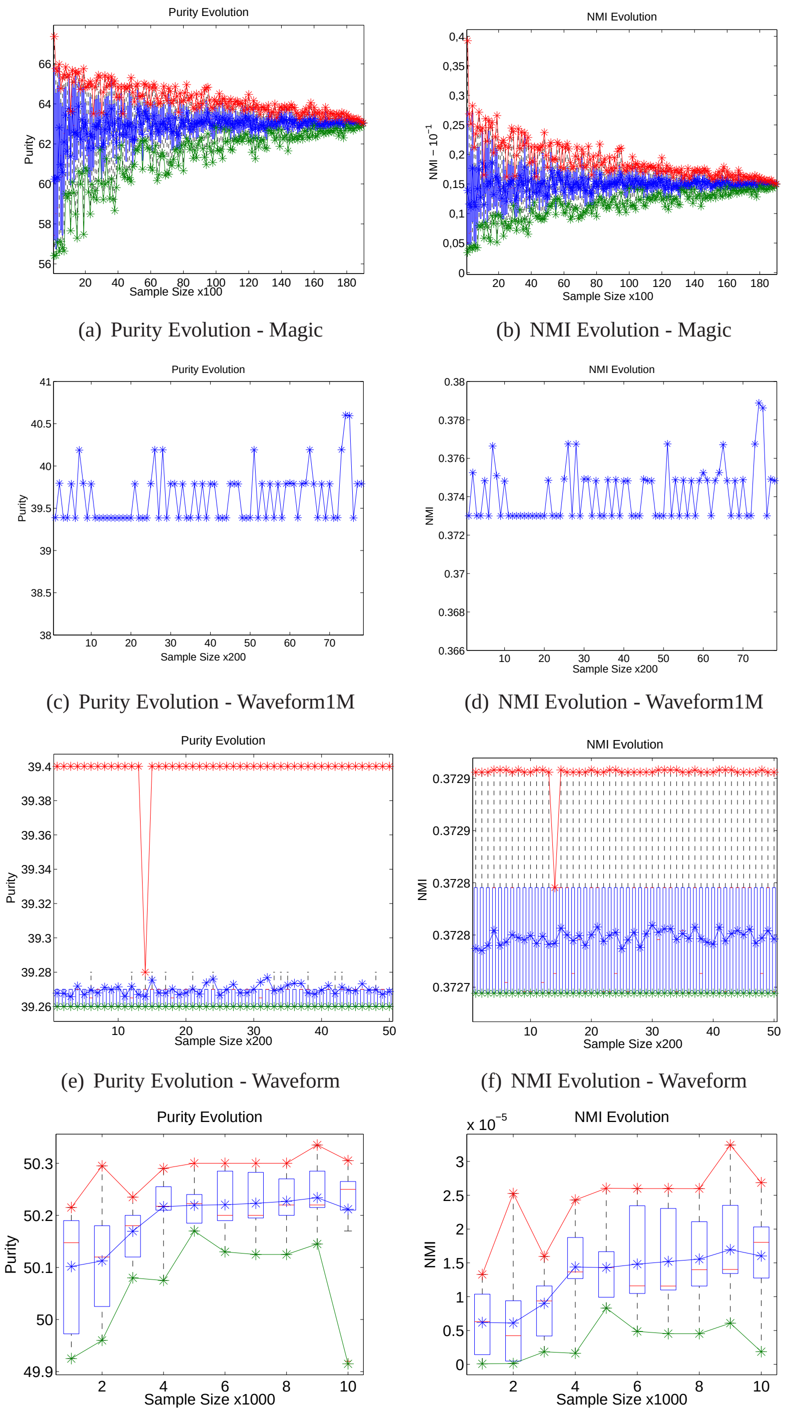
Fig. 9: Evolution of Projection Operators Difference



(a) ΔPO_m - Magic

(b) ΔPO_m - Waveform

Fig. 10: Evolution of Projection Operators Difference - Spectral Clustering



(a) Purity Evolution - Magic

(b) NMI Evolution - Magic

(c) Purity Evolution - Waveform1M

(d) NMI Evolution - Waveform1M

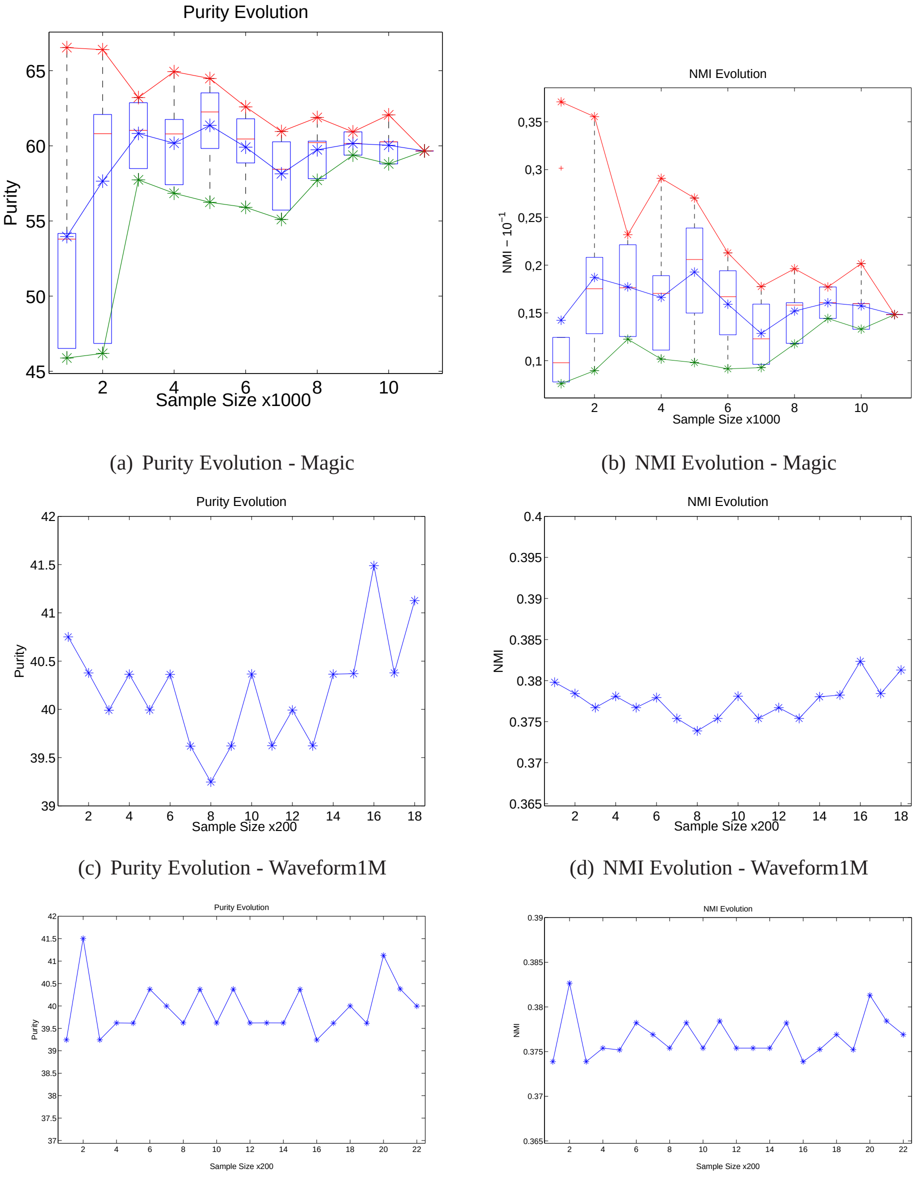
(e) Purity Evolution - Waveform

(f) NMI Evolution - Waveform

(g) Purity Evolution - delta

(h) NMI Evolution - delta

Fig. 11: Clustering Quality Evolution



(a) Purity Evolution - Magic

(b) NMI Evolution - Magic

(c) Purity Evolution - Waveform1M

(d) NMI Evolution - Waveform1M

(e) Purity Evolution - Waveform

(f) NMI Evolution - Waveform

Fig. 12: Clustering Quality Evolution - Spectral Clustering