Enhancing the Effectiveness of Clustering with Spectra Analysis

Wenyuan Li, Kok-Leong Ong, and Wee-Keong Ng, *Member, IEEE*

Abstract

For many clustering algorithms such as k-means, EM and CLOPE, there is usually a requirement to set some parameters. Often, these parameters directly or indirectly control the number of clusters, i.e., k, to return. In the presence of different data characteristics and analysis contexts, it is often difficult for the user to estimate the number of clusters in the data set. This is especially true in *text* collections such as Web documents, images or biological data. In an effort to improve the effectiveness of clustering, we seek the answer to a fundamental question: *How can we effectively estimate the natural number of clusters in a given data set?* We propose an efficient method based on spectra analysis of eigenvalues (*not* eigenvectors) of the data set as the solution to the above. We first present the relationship between a data set and its underlying spectra with theoretical and experimental results. We then show how our method is capable of suggesting a range of k that is well-suited to different analysis contexts. Finally, we conclude with further empirical results to show how the answer to this fundamental question enhances the clustering process for large *text* collections and gene expression data.

Index Terms

Data Clustering, Spectra Analysis.

I. Introduction

THE bulk of data mining research is devoted to the development of techniques that solve a particular problem. Often, the focus is on the design of algorithms that outperform previous techniques either in terms of speed or accuracy. While such effort is a valuable endeavor, the overall success of knowledge discovery, i.e., the larger context of data mining, requires more than just algorithms for the data. With an exponential increase in the complexity and volume of data in recent years, an important and crucial factor to the success of knowledge discovery is to narrow the gap between the user and the available machinery.

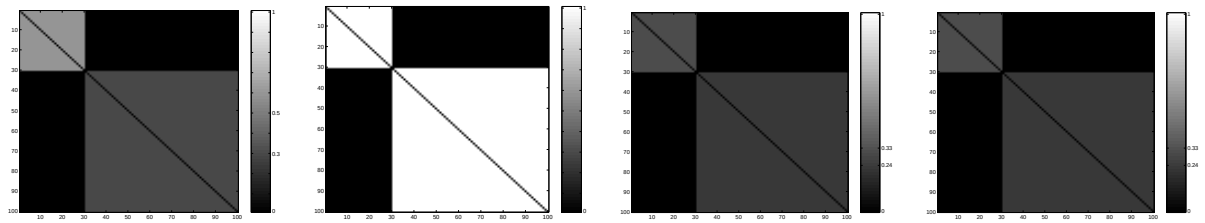
A good example to argue a case for the above is clustering. In clustering, there is usually a requirement to set some parameters. Often, these parameters directly or indirectly control the number of clusters, i.e., k, to return. In the presence of different data characteristics and analysis contexts, it is often difficult for the user to determine the correct value of k in the data set [31], [38], [36]. Therefore, setting these parameters require either detailed pre-existing knowledge of the data, or time-consuming trial and error. In the latter case, the user also needs sufficient knowledge to know what is a good clustering. Worse, if the data set is very large or has a high dimensionality, the trial and error process becomes very inefficient.

Furthermore, certain algorithms require a good estimate of the input parameters. For example, the EM [8] algorithm is known to perform well in image segmentation [14] when k and the initialization parameters are close to their true values. Yet, one reason that limits its application is the difficulty of estimating a k that is close to this true value. Likewise, a poor parameter setting (which indirectly determines k) in CLOPE [41] can dramatically increase its runtime. In all cases above, the user is likely to devote more time in parameter tuning rather than knowledge discovery. Clearly, this is undesirable.

To further strengthen the case, many analytical applications, e.g., information retrieval and pattern recognition, etc., require various forms of data preparation such as data cleaning. One of these approaches in data cleaning is to use clustering algorithms [1], [26] to remove noise (or outliers). Again, if the user is unfamiliar with the nature of the data set in question, deciding a value for k becomes a daunting task that can affect the effectiveness of data cleaning [4]. Since data preparation is such a critical step to effective analysis, getting the right value of k to start the process is important.

In this paper, we provide a concrete instance of the above problem by studying this issue in the context of complex data sets, e.g., text collections, images, biological data, etc. Such data sets are inherently large in size and have dimensionality in magnitude of hundreds to several thousands. And considering the domain specificity of the data, getting the user to set a value for k becomes a challenge. In this case, we argue that a good starting point is to initialize k to the natural number of clusters. This gives rise to the fundamental question: *How can we effectively estimate the natural number of clusters for a given data set?* In our attempt to answer this question, we made the following contributions in this paper.

- We develop an accurate method to determine k of a given data set — Our solution is to perform a spectra analysis on the similarity space of the data set by analyzing the eigenvalues (*not* eigenvectors) that encode the answer to our question. As our method does not require the user to specify a range of k to test with a clustering algorithm, the analysis is also highly efficient. We present the details in Sections II and III.
- Recognizing the subjectivity of what constitutes to the right value of k, we extend our work to estimate a range of k of a given data set — Since the outcome of clustering depends largely on the similarity measure used and the analysis



(a) (b) (c) (d)

Fig. 1. A comparison of two similarity matrices before and after the 3-normalization: (a) $\mathbf{S}_1$ with 2 disjoint clusters – cluster 1 has 30 instances and average similarity of 0.5 and cluster 2 with 70 instances with average similarity of 0.3; (b) $\mathbf{S}_2$ with the same disjoint clusters as $\mathbf{S}_1$ but both clusters have an average similarity of 1; (c) normalized $\mathbf{S}_1$; (d) normalized $\mathbf{S}_2$.

context, computing a rigid value of k may be inappropriate in some situations. We show that our method can be easily extended to handle this real-life issue by further spectra analysis. This is discussed in depth in Section IV.

- We demonstrate how our solution can be realized in the context of a larger system by presenting an algorithm to automate this preprocessing — If the ultimate goal is to narrow the gap between the user and the available machinery, our solution should be automated so that it can be implemented on real-life systems in a non-intrusive manner. Fundamental to this is an algorithm that exploits our method to determine k. We introduce this algorithm in Section V together with a motivating example in Section VI.
- We report a wide range of experimental results that empirically support the effectiveness of our method on real-life data sets — To confirm the effectiveness of our solution, we conducted a range of experiments on many real-life public data sets. In all cases, our approach performed well in terms of accuracy and speed. Sections III-C, and VII report these results.

II. Preliminaries

Most algorithms perform clustering by embedding the data in some similarity space [35] determined by some similarity measures, e.g., cosine similarity [21]. Let $\mathbf{S} = (s_{ij})_{n\times n}$ be the similarity space matrix, where $0 \leqslant s_{ij} \leqslant 1$, $s_{ii} = 1$ and $s_{ij} = s_{ji}$, i.e., $\mathbf{S}$ is symmetric. Further, let $\mathcal{G}(\mathbf{S}) = \langle V, E, \mathbf{S}\rangle$ be the graph of $\mathbf{S}$, where V is the set of n vertices and $\mathbf{E}$ is the set of weighted edges. Each vertex v_i of $\mathcal{G}(\mathbf{S})$ corresponds to the i-th column (or row) of $\mathbf{S}$, and the weight of each edge $\widehat{v_iv_j}$ corresponds to the non-diagonal entry s_{ij}. For any two vertices (v_i, v_j), a larger value of s_{ij} indicates a higher connectivity between them, and vice versa. For ease of exposition, we refer $\mathbf{S}$ as the adjacency matrix of $\mathcal{G}(\mathbf{S})$ by setting $s_{ii} = 0$ in the rest of the paper.

In cases where a single similarity measure is used, it is actually possible to analyze the spectra distribution of $\mathcal{G}(\mathbf{S})$ directly. However, there are occasions where multiple similarity measures need to be considered due to different analysis contexts. In such a situation, there is a need to first perform normalization of the different adjacency matrices to ensure comparability. As an example, Figure 1(a) and (b) shows two different similarity matrices on the same data set. While the similarity measure is different, the clustering result turns out to be the same. The only difference is that the first similarity matrix $\mathbf{S}_1$ has a totally different spectra from that of $\mathbf{S}_2$ because of the different range of similarities within each cluster. Since they have the same clustering, it is therefore necessary to normalize the adjacency matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ so that they also have a similar spectra of $\mathcal{G}(\mathbf{S}_1)$ and $\mathcal{G}(\mathbf{S}_2)$ respectively.

Definition 1 (Normalization of $\mathbf{S}$*):* Given a similarity matrix $\mathbf{S} = (s_{ij})_{n\times n}$, the normalization of $\mathbf{S}$ by p-norm (or p-normalization) is denoted as $L_p(\mathbf{S}) = (\ell_{ij})_{n\times n}$ where

$$\ell_{ij} = \frac{s_{ij}}{\sqrt{\|\vec{s}_i\|_p \|\vec{s}_j\|_p}} \tag{1}$$

and $\|\vec{s}_i\|_p = (\sum_j s_{ij}^p)^{\frac{1}{p}}$ with $p > 0$.

The p-norm presents some interesting properties. One of them is that $L_p(\mathbf{S})$ remains symmetric and its diagonal entries are set to 0. Furthermore, regardless of the value p is set to, the p-norm on $\mathbf{S}$ has the property of standardizing the adjacency matrix without affecting the underlying clustering structure in $\mathcal{G}(\mathbf{S})$. Continuing our example, Figure 1(c) and (d) shows the normalized $\mathbf{S}_1$ and $\mathbf{S}_2$ after applying a 3-normalization, i.e., $p = 3$, where a similar result can also be obtained with other values of p. Theorem 1 gives this formal guarantee.

Theorem 1 (Properties of $L_p(\mathbf{S})$*):* Let $\mathbf{S} = (s_{ij})_{n\times n}$ with $s_{ij} \geqslant 0$. Then, the p-norm of $\mathbf{S}$, i.e., $L_p(\mathbf{S}) = (\ell_{ij})_{n\times n}$ satisfies (i) $\ell_{ij} = \ell_{ji}$; (ii) $\ell_{ii} = 0$; (iii) $0 \leqslant \ell_{ij} \leqslant 1$.

Proof: Property (i) and (ii) are obvious by Eq.(1). The p-norm of a vector $\vec{x} = (x_1, \ldots, x_n)$ [16] has the property $\|\vec{x}\|_1 \geqslant \|\vec{x}\|_2 \geqslant \ldots \geqslant \|\vec{x}\|_\infty = \max_{1\leqslant i\leqslant n} |x_i|$, and therefore we have

$$\ell_{ij} = \frac{s_{ij}}{\sqrt{\|\vec{s}_i\|_p \|\vec{s}_j\|_p}} \leqslant \frac{s_{ij}}{\sqrt{\|\vec{s}_i\|_\infty \|\vec{s}_j\|_\infty}} = \frac{s_{ij}}{\sqrt{\max_{1\leqslant j\leqslant n} |s_{ij}| \max_{1\leqslant i\leqslant n} |s_{ij}|}} \leqslant 1 \tag{2}$$

■

In addition to comparability reason, $L_p(\mathbf{S})$ is also closely related to the field of spectra graph theory. In particular, when $p = 1$, the spectra of $L_p(\mathbf{S})$ reveals the number of natural clusters in $\mathcal{G}(\mathbf{S})$; and when p is increased beyond 1, further clustering information can also be discovered.

Once normalized, we can obtain the spectra of $L_p(\mathbf{S})$ by computing the eigenvalues as follows. Notably, we can utilize many mature efficient solutions in matrix computation by this formulation. Let $\mathbf{A}$ denote a symmetric $n \times n$ real matrix, λ an eigenvalue of $\mathbf{A}$ that satisfies $\mathbf{A}\vec{x} = \lambda\vec{x}$ where $\vec{x}$ is a vector. It is known [16] that all eigenvalues of $\mathbf{A}$ are real-numbers and that each of the n eigenvalue corresponds to a unique $\vec{x}$. Without loss of generality, these n eigenvalues are denoted as $\text{eig}(\mathbf{A}) = \{\lambda_1(\mathbf{A}), \lambda_2(\mathbf{A}), \ldots, \lambda_n(\mathbf{A})\}$ that we refer to as the spectra distribution. These eigenvalues are assumed to be in decreasing order, i.e., $\lambda_1(\mathbf{A}) \geqslant \lambda_2(\mathbf{A}) \geqslant \ldots \geqslant \lambda_n(\mathbf{A})$, where $\lambda_k(\mathbf{A})$ denotes the k-th largest eigenvalue of $\mathbf{A}$. Specific to our work, we are interested in $\text{eig}(L_p(\mathbf{S}))$, which is the p-spectra of $\mathcal{G}(\mathbf{S})$.

Once we establish the relationship between the structural properties of $\mathcal{G}(\mathbf{S})$ and its spectra, the answer to our fundamental question is obtained. That is, the answer to our fundamental question is now mapped to a matter of how to analyze $\text{eig}(L_p(\mathbf{S}))$. This approach is particularly attractive in comparison to existing methods of estimating k in terms of runtime and simplicity. More importantly, as we demonstrate in the subsequent sections, our proposal has better flexibility in terms of accommodating the different analytical contexts of the users, e.g., the use of a different clustering algorithm or evaluation index, on the same data set.

III. Spectra Analysis by 1-Normalization

In spectral graph theory, the weighted Laplacian matrix is commonly used [5]. For ease of analysis, let $\mathbf{L} = \mathbf{T}^{-1/2}(\mathbf{T} - \mathbf{S})\mathbf{T}^{-1/2}$ be the weighted Laplacian matrix of $\mathcal{G}(\mathbf{S})$ where $\mathbf{T}$ is the diagonal matrix, and $\text{diag}(d_i)$ and $d_i = \sum_j s_{ij}$ the degree of vertex v_i in $\mathcal{G}(\mathbf{S})$. By Definition 1, $\text{eig}(L_1(\mathbf{S})) = \{1 - \lambda \,|\, \lambda \in \text{eig}(\mathbf{L})\}$ and therefore, the 1-spectra of $\mathcal{G}(\mathbf{S})$ maintains the same conclusions and properties of those found in $\mathbf{L}$. By this, we can establish the following basic facts drawn from spectral graph theory. Among them, the last fact about $\mathcal{G}(\mathbf{S})$ is an important property that we exploit – it depicts the relationship between the spectra of the disjoint subgraphs $\mathcal{G}(\mathbf{S}_i)$ and the spectra of $\mathcal{G}(\mathbf{S})$.

Theorem 2: For a graph $\mathcal{G}(\mathbf{S})$ and its 1-spectra $\text{eig}(L_1(\mathbf{S})) = (\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, where $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$, we have: (i) $-1 \leqslant \lambda_i \leqslant 1$ for $i = 1, 2, \cdots, n$ where $\sum \lambda_i = 0$, and $\lambda_1 = 1$; (ii) if $\mathcal{G}(\mathbf{S})$ is connected, then $\lambda_2 < 1$; and (iii) the 1-spectra of $\mathcal{G}(\mathbf{S})$ is the union of the 1-spectra of its disjoint subgraphs $\mathcal{G}(\mathbf{S}_i)$.

Proof: As shown in [5], [16]. ■

We will begin with the spectra properties of a conceptually disjoint data set, whose chosen similarity measure achieves a perfect clustering. From this simple case, we then extend our observations to real-world data sets, and show how the value of k can be obtained.

Before the discussion of p-spectra analysis, a standard clustering structure in the similarity matrix $\mathbf{S}$ is denoted as follows. It shall facilitate our discussion. The similarity matrix $\mathbf{S}$ is reordered by its natural clustering structure (k clusters) and be presented as the form of the block submatrices in Eq.(3).

$$\mathbf{S} = \begin{array}{c} \left[\begin{array}{ccc} \mathbf{S}_{11} & \cdots & \mathbf{S}_{1k} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{k1} & \cdots & \mathbf{S}_{kk} \end{array}\right] \\ \begin{array}{ccc} n_1 & \cdots & n_k \end{array} \end{array} \quad \begin{array}{c} n_1 \\ \vdots \\ n_k \end{array} \tag{3}$$

with the properties: each diagonal block submatrix $\mathbf{S}_{jj}$ ($1 \leqslant j \leqslant k$) represents the intra-similarity matrix within the j-th cluster. Thus non-diagonal block submatrix $\mathbf{S}_{ij}$ shows the inter-similarity matrix between the i-th and j-th clusters. We denote n_j as the number of rows or columns of $\mathbf{S}_{jj}$, and say $l \in \mathbf{S}_{jj}$ if $\sum_{l=1}^{j-1} n_j \leqslant l \leqslant \sum_{l=1}^{j} n_j$. Therefore, if $i \in \mathbf{S}_{jj}$, then $\vec{s_i} = (s_{1i}, s_{2i}, \ldots, s_{ni})^T$, the i-th column vector of $\mathbf{S}$, belongs to the j-th cluster. And we say that $\vec{s_i}$ is in the area of $\mathbf{S}_{jj}$.

A. A Simple Case

Assume that we have a conceptually disjoint data set, whose chosen similarity measure achieves a perfect clustering. In this case, the similarity matrix $\mathbf{A}$ will have the structure given by Eq.(3) with the properties: all entries in each diagonal block matrix $\mathbf{A}_{ii}$ of $\mathbf{A}$ are 1; and all entries in each non-diagonal block matrix $\mathbf{A}_{ij}$ in $\mathbf{A}$ are 0. From this similarity matrix, we can get the corresponding graph $\mathcal{G}(\mathbf{A})$ and obtain its 1-spectra in decreasing order [5], i.e.,

$$\lambda_i(L_1(\mathbf{A})) = \begin{cases} 1, & 1 \leqslant i \leqslant k \\ 0, & k < i \leqslant n \end{cases} \tag{4}$$

Lemma 1: Given a similarity matrix $\mathbf{S}$ as defined in Equation (3), where $n_1 + \cdots + n_k = n$; where each diagonal entry $\mathbf{S}_{ii}$ satisfies $0 < n_i - \|\mathbf{S}_{ii}\|_F < \delta(\delta \to 0)$; and where each non-diagonal entry $\mathbf{S}_{ij}$ satisfies $\|\mathbf{S}_{ij}\|_F \to 0$ ($\|\cdot\|_F$ is the Frobenius norm), then $\mathbf{S}$ achieves a perfect clustering of k clusters. At the same time, the 1-spectra of $\mathcal{G}(\mathbf{S})$ exhibits the following properties:

TABLE I

A SMALL TEXT COLLECTION TAKEN AND MODIFIED FROM [27] FOR OUR EXAMPLE. IT CONTAINS THE TITLES OF 12 TECHNICAL MEMORANDA: 5 ABOUT HUMAN-COMPUTER INTERACTION; 4 ABOUT MATHEMATICAL GRAPH THEORY; AND 3 ABOUT CLUSTERING. THE TOPICS ARE CONCEPTUALLY DISJOINT WITH TWO ASSUMPTIONS: (I) THE ITALICIZED TERMS ARE THE SELECTED FEATURE SET; AND (II) THE COSINE SIMILARITY MEASURE IS USED TO COMPUTE $\mathbf{S}$.

c1	*Human* machine *interface* for ABC *computer* applications
c2	A *survey* of *user* opinion of *computer system response time*
c3	The *EPS user interface* management *system*
c4	*System* and *human system* engineering testing of *EPS*
c5	Relation of *user* perceived *response time* to error measurement
m1	The generation of random, binary, ordered *trees*
m2	The intersection *graph* of paths in *trees*
m3	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4	*Graph minors*: A *survey*
d1	Linguistic features and *clustering* algorithms for topical *document clustering*
d2	A comparison of *document clustering techniques*
d3	*Survey* of *clustering* Data Mining *Techniques*

$$\begin{array}{r} \lambda_i(L_1(\mathbf{S})) \to 1 \quad (i = 1, \cdots, k \text{ and } 0 < \lambda_i \leqslant 1) \\ |\lambda_i(L_1(\mathbf{S}))| \to 0 \quad (i = k+1, \cdots, n) \end{array} \tag{5}$$

Proof: Let $\mathbf{E} = L_1(\mathbf{S}) - L_1(\mathbf{A})$ and $\mathbf{E}' = \mathbf{S} - \mathbf{A}$, where $\mathbf{A}$ is as defined in Eq.(3). From definitions of $\mathbf{A}$ and $\mathbf{S}$, we obtain the following:

$$\left.\begin{array}{rr} 0 < n_i - \|\mathbf{S}_{ii}\|_F < \delta(\delta \to 0), & \|\mathbf{A}_{ii}\|_F = n_i \\ \|\mathbf{S}_{ij}\|_F \to 0, & \|\mathbf{A}_{ij}\|_F = 0 \end{array}\right\} \Rightarrow \|\mathbf{E}'\|_F \to 0 \tag{6}$$

When $\|\mathbf{E}'\|_F \to 0$, this also means that $L_1(\mathbf{S})$ is much closer to $L_1(\mathbf{A})$ and thus we have:

$$\|\mathbf{E}\|_F \to 0 \tag{7}$$

By the property of the Frobenius norm, and the p matrix norm (where $p = 2$ [16]), we have:

$$\|\mathbf{E}\|_2 \leqslant \|\mathbf{E}\|_F \tag{8}$$

and

$$|\lambda_i(L_1(\mathbf{A}) + \mathbf{E}) - \lambda_i(L_1(\mathbf{A}))| \leqslant \|\mathbf{E}\|_2, \quad (i = 1, \cdots, n) \tag{9}$$

where $\|\cdot\|_2$ is the $p = 2$ matrix norm. Eq.(8) states that the Frobenius norm of a matrix is always greater than or equal to the p matrix norm at $p = 2$, and Eq.(9) defines the distance between the eigenvalues in $L_1(\mathbf{A})$ and its perturbation matrix $L_1(\mathbf{S})$. In addition, the sensitivity of the eigenvalues in $L_1(\mathbf{A})$ to its perturbation is given by $\|\mathbf{E}\|_2$. Hence, from Equations (6), (7) (8), and (9), we can conclude that:

$$\lambda_i(L_1(\mathbf{S})) \to \lambda_i(L_1(\mathbf{A})), \quad (i = 1, \cdots, n) \tag{10}$$

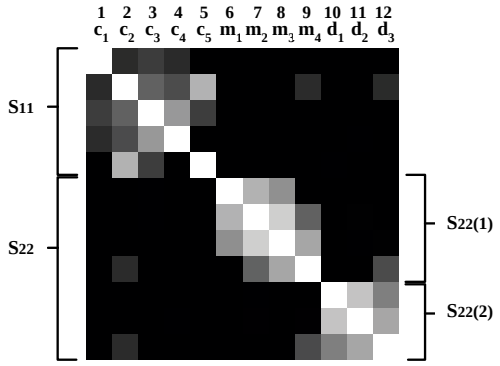
which when we combine with Eq.(4), we arrive at Lemma 1. ■

Simply put, when the 1-spectra distribution satisfies Eq.(5), then $\mathbf{S}$ shows a good clustering, i.e., the intra-similarity approaches 1, and the inter-similarity approaches 0. As an example, suppose we have a collection with 3 clusters as depicted in Table I. The 3 topics are setup to be conceptually disjoint, and the similarity measure as well as the feature set are selected such that the outcome produces 3 distinct clusters. In this ideal condition, the spectra distribution (as shown in Figure 2) behaves as per Eq.(5).

Of course, real-world data sets that exhibit perfect clustering are extremely rare. This is especially the case for data whose dimensionality is large but the data itself is sparse, e.g., *text* collections. In this case, most similarity measures do not rate two data points as distinctively similar, or different. If we perform a spectra analysis on the data set, we will end up with a 1-spectra of $\mathcal{G}(\mathbf{S})$ that is very different from our example in Figure 2. As we will see next, this 1-spectra distribution is much more complex.

	λ_1	λ_2	λ_3	λ_4	λ_5
$\mathcal{G}(\mathbf{S})$	1	0.92	0.88	0.25	0.06
$\mathcal{G}(\mathbf{S}_{11})$	1	0.25	—	—	—
$\mathcal{G}(\mathbf{S}_{22})$	1	0.91	0.07	—	—
$\mathcal{G}(\mathbf{S}_{22}(1))$	1	0.04	—	—	—
$\mathcal{G}(\mathbf{S}_{22}(2))$	1	—	—	—	—

(a)



(b)

Fig. 2. The spectra distribution of the collection in Table I: (a) Spectrum (> 0) of $\mathcal{G}(\mathbf{S})$ and its subgraphs; (b) a graphical representation of $\mathbf{S}$. Note that all grey images in this paper are not "plots" of the spectra – They are a graphical way of summarizing the results of clustering for analysis purposes.

B. 1-*Spectra Distribution in Large Data Sets*

Point (iii) of Theorem 2 offers a strong conclusion between $\mathcal{G}(\mathbf{S})$ and its subgraphs. However, real-world data sets often exhibit a different characteristic. If we examine their corresponding $\mathcal{G}(\mathbf{S})$, we will see that the connections between $\mathcal{G}(\mathbf{S})$ and its subgraphs are weak, i.e., Lemma 1 no longer holds.

Fortunately, we can still judge the cluster quality and estimate the number of natural clusters with spectral analysis. In this section, we present the proofs that lead to the conclusion about cluster quality and k. But first, we need to introduce the Cheeger constant. Let $SV \subset V$ of $\mathcal{G}(\mathbf{S})$. We define the volume of SV as:

$$\text{vol}(SV) = \sum_{v \in SV} d_v \tag{11}$$

where d_v is the sum of all weighted edges containing vertex v. Further, let $E(\delta SV)$ be the set of edges, where each edge has one of its vertices in SV but not the other, i.e., $\overline{SV}$. Then, its volume is given by:

$$|E(\delta SV)| = \sum_{v_i \in SV, v_j \notin SV} \text{weight}(v_i, v_j) \tag{12}$$

and by Equations (11) and (12), we derive the Cheeger constant:

$$h(\mathcal{G}) = \min_{SV \subset V} \frac{|E(\delta SV)|}{\min(\text{vol}(SV), \text{vol}(\overline{SV}))} \tag{13}$$

which measures the optimality of the bipartition in a graph. The magnitude $|E(\delta SV)|$ measures the connectivity between SV and $\overline{SV}$ while $\text{vol}(SV)$ measures the density of SV against V.

Since SV enumerates all subsets of V, $h(\mathcal{G})$ is a good measure that finds the best bipartition, i.e., $\langle SV, \overline{SV} \rangle$. Perhaps, more interesting is the observation that no other bipartition gives a better clustering than the bipartition determined by $h(\mathcal{G})$. Therefore, $h(\mathcal{G})$ can be used as an indicator of cluster quality, i.e., the lower its value, the better the clustering.

Theorem 3: Given the 1-spectra of $\mathcal{G}(\mathbf{S})$ as $1 = \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, if $\lambda_2 \to 1$, then there exists a good bipartition for $\mathcal{G}(\mathbf{S})$, i.e., a good cluster quality.

Proof: From [5], we have the Cheeger inequality: $\frac{(1-\lambda_2)}{2} \leqslant h(\mathcal{G}) < \sqrt{2(1-\lambda_2)}$ that gives the bound of $h(\mathcal{G})$. By this inequality, if $\lambda_2 \to 1$, then $h(\mathcal{G}) \to 0$. And since $h(\mathcal{G}) \to 0$ implies a good clustering, we have the above. ■

For a given similarity measure, Theorem 3 allows us to get a "feel" of the clustering quality without actually running the clustering algorithm. This saves computing resources and reduces the amount of time the user waits to get a response. By minimizing this "waiting time" during initial analysis, we promote interactivity between the user and the clustering algorithm. In such a system, Theorem 3 can also be used to help judge the suitability of each supported similarity measure. Once the measure is decided, the theorem to be presented next, provides the user a starting value of k.

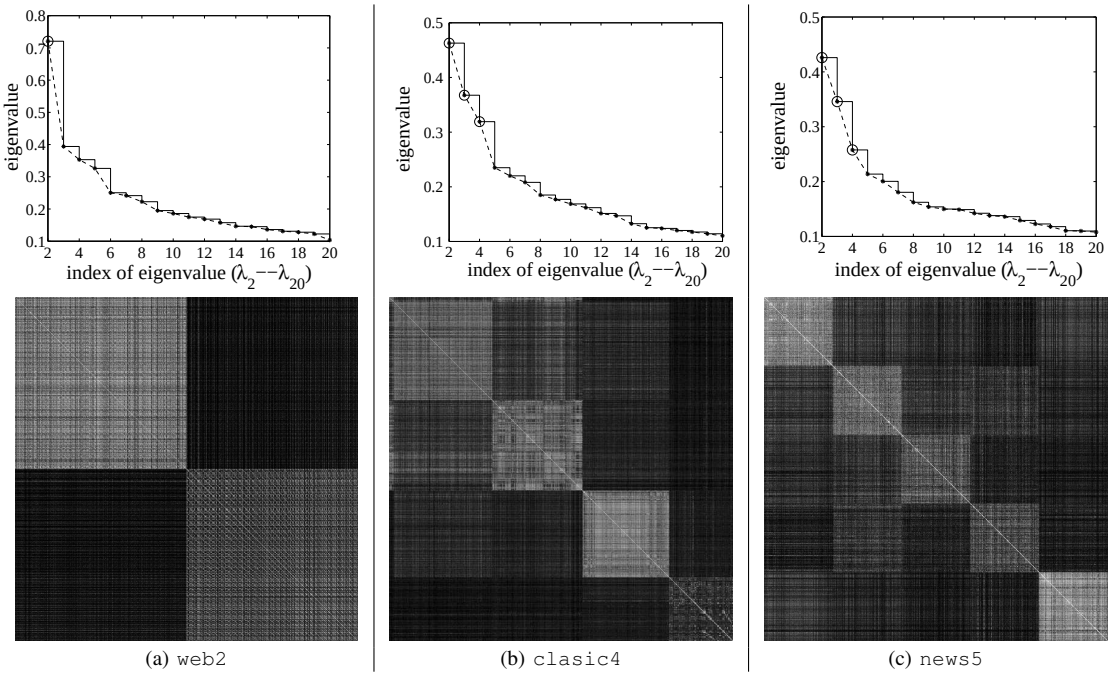
Theorem 4: Given the 1-spectra of $\mathcal{G}(\mathbf{S})$ as $1 = \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, $\exists k \geqslant 1$ such that when $i \leqslant k$, there are $\alpha_i \to 1$ and $\alpha_i - \alpha_{i+1} > \delta$ $(0 < \delta < 1)$ for the sequence $\alpha_i = \frac{\lambda_{i+1}}{\lambda_2}$, $(i \geqslant 1)$, where δ is a predefined threshold to measure the first large gap between α_i; and $k+1$ is the natural number of clusters in the data set.

Proof: Since Theorem 3 applies to both $\mathcal{G}(\mathbf{S})$ and its subgraphs $\mathcal{G}(\mathbf{S}_{ii})$, then we can estimate the cluster quality of the bipartition in $\mathcal{G}(\mathbf{S}_{ii})$ (as well as its subgraphs). Combine with Point (iii) of Theorem 2, we can conclude that the number of eigenvalues in $\mathcal{G}(\mathbf{S})$ (that approach 1 and have large eigengaps) gives the value of $k+1$, i.e., the number of clusters. ■

TABLE II

THE *text* COLLECTIONS USED IN OUR EXPERIMENTS TO ESTIMATE k: WE SELECTED 4 CLASSES OF `classic` WITH EACH CLASS CONTAINING 1,000 DOCUMENTS; 5 `newsgroups` WITH EACH NEWSGROUP CONTAINING 500 DOCUMENTS; 2 CATEGORIES OF THE `webset` WITH EACH CATEGORY CONTAINING 600 DOCUMENTS.

Collections	Source	# Classes	# Documents
`classic`	ADI/CACM/CISI/CRAN/MED	5	5559
`newsgroup`	UseNet news postings	17	7473
`webset`	Categories in Yahoo [33]	10	6607



(a) `web2` (b) `clasic4` (c) `news5`

Fig. 3. The spectrum graphs and the graphical representation of their clustering for all 3 collections: the first two data sets are conceptually disjoint, and the last is conceptually overlapping.

By Theorem 4, we shall use the α value sequence for analysis in subsequent experiments except we clearly mention to use eigenvalues. To cite an example for the above, we revisit Table I and Figure 2. By the Cheeger constant of $\mathcal{G}(\mathbf{S})$, $SV = \{c_1, c_2, c_3, c_4, c_5\}$ and $\overline{SV} = \{m_1, m_2, m_3, m_4, d_1, d_2, d_3\}$ produces the best bipartition. Thus, $\mathbf{S}_{11}$ represents the inter-similarities in SV and $\mathbf{S}_{22}$ represents inter-similarities in $\overline{SV}$. From Theorem 3, we can assess the cluster quality of $\mathcal{G}(\mathbf{S})$'s bipartition by λ_2. Also, we can recursively consider the bipartitions of the bipartitions of $\mathcal{G}(\mathbf{S})$, i.e., $\mathcal{G}(\mathbf{S}_{11})$ and $\mathcal{G}(\mathbf{S}_{22})$. Again, the Cheeger constant of $\mathcal{G}(\mathbf{S}_{22})$ shows that $\mathcal{G}(\mathbf{S}_{22}(1))$ and $\mathcal{G}(\mathbf{S}_{22}(2))$ are the best bipartition in the subgraph $\mathcal{G}(\mathbf{S}_{22})$. Likewise, the λ_2 of $\mathcal{G}(\mathbf{S}_{11})$, $\mathcal{G}(\mathbf{S}_{22})$, $\mathcal{G}(\mathbf{S}_{22}(1))$, and $\mathcal{G}(\mathbf{S}_{22})$ all satisfy this observation.

In fact, this recursive bisection of $\mathcal{G}(\mathbf{S})$ is a form of clustering using the Cheeger constant – the spectra of $\mathcal{G}(\mathbf{S}_{22})$ contains the eigenvalues of $\mathcal{G}(\mathbf{S}_{22}(1))$ and $\mathcal{G}(\mathbf{S}_{22}(2))$, and $\mathcal{G}(\mathbf{S})$ contains the eigenvalues of $\mathcal{G}(\mathbf{S}_{11})$ and $\mathcal{G}(\mathbf{S}_{22})$ respectively (despite with some small "fluctuations"). As shown in Figure 2(a), λ_2 of $\mathcal{G}(\mathbf{S})$ gives the cluster quality of the bipartition $\mathcal{G}(\mathbf{S}_{11})$ and $\mathcal{G}(\mathbf{S}_{22})$ in $\mathcal{G}(\mathbf{S})$; and λ_3 of $\mathcal{G}(\mathbf{S})$, which corresponds to λ_2 of $\mathcal{G}(\mathbf{S}_{22})$, gives the cluster quality indicator for the bipartition $\mathcal{G}(\mathbf{S}_{22}(1))$ and $\mathcal{G}(\mathbf{S}_{22}(2))$ in $\mathcal{G}(\mathbf{S}_{22})$, and so on. Therefore, if there exist k distinct and dense diagonal squares (i.e., $\mathbf{S}_{ii}$ where $1 \leqslant i \leqslant k$) in the matrix, then λ_i of $\mathcal{G}(\mathbf{S})$ will be the cluster quality indicator for the i-th bipartition ($2 \leqslant i \leqslant k$), and the largest k eigenvalues of $\mathcal{G}(\mathbf{S})$ give the estimated number of clusters in the data.

C. Empirical Results

We digress for a moment to a set of experiments to provide some evidence on the viability of the 1-spectra of $\mathcal{G}(\mathbf{S})$ to estimate k. The text data sets used are of high dimensionality and sufficiently large to reflect the real-world situation. The details of these data sets are given in Table II.

In practice, we can estimate k by using Theorem 4. However, for the purpose of illustration, we will walk through the analysis by using eigenvalues, not α sequence. Since λ_1 is always 1, our analysis begins from λ_2. In Figure 3, we have marked out the large eigenvalues whose gap is larger than the rest. This gap can be best identified by the big stair steps among the eigenvalues. According to Theorem 4, the number of such eigenvalues (including λ_1) gives the number of clusters. We can

TABLE III

COMPARISON OF OUR APPROACH AGAINST 3 WELL-KNOWN INDEXES: THE CALINSKI AND HARABASZ (CH) INDEX; KRZANOWSKI AND LAI (KL) INDEX; AND HARTIGAN (HART) INDEX WITH 3 WELL-KNOWN CLUSTERING ALGORITHMS: BISECTING k-MEANS, GRAPH-BASED, AND HIERARCHICAL – A ($\surd$) INDICATES A CORRECT ESTIMATION.

	Bisecting k-means			Graph-based			Hierarchical		
	web2	classic4	news5	web2	classic4	news5	web2	classic4	news5
CH	5	2	3	3	3	3	7	2	5 ($\surd$)
KL	29	17	22	27	21	22	22	21	9
Hart	6	13	4 ($\surd$)	6	10	4 ($\surd$)	1	2	1

verify this by analyzing their corresponding grey images in the same figure.

Figure 3(a) shows the web2 collection with just 2 class labels and their topic being completely disjoint: *finance* and *sport*. In this case, notice that λ_2 has a higher value than the others. Since the remaining eigenvalues fall along a smooth curve, this phenomenon conforms to Theorem 3 and 4. In this case, we therefore conclude $k = 2$. At the same time, the high value of λ_2 indicates that this is a good clustering by the similarity measure used.

In the second case, classic4 has 4 topics from scientific abstract from different research domains: *computing algorithms*, *information retrieval*, *aerodynamics* and *medicine*. They are conceptually disjoint, which can be observed from Figure 3(b) where there are 4 distinctive diagonal squares. From its spectra graph, we observe that λ_2, λ_3 and λ_4 show higher values and wider gaps than other eigenvalues. Again by the same theorem, our method obtains the correct number of clusters, i.e., $k = 4$.

The third collection is the most challenging. There are 5 topics: *atheism*, *comp.sys*, *comp.windows*, *misc.forsale* and *rec.sport*. Unlike the previous two collections, the topics are not disjoint. In this case, both *comp.sys* and *comp.windows* belong to the broader topic of *comp* in the newsgroup. Therefore, the graphical representation in Figure 3(c) do not show a set of distinctive squares along its diagonal. When we apply our analysis, only λ_2, λ_3, and λ_4 have a higher value and a wider gap than the others. So by our theorems, $k = 4$. This conclusion is actually reasonable since *comp* is more different from the other topics. If we observe the grey image in Figure 3(c), we see that the second and third squares appear to "meshed" together – an indication of similarity between *comp.sys* and *comp.windows*.

Furthermore, *comp.sys*, *comp.windows* and *misc.forsale* can also be viewed as one topic. This is because *misc.forsale* has many postings on buying and selling of computer parts. Again, this can be observed in the grey image. On the spectra graph, λ_4 is much lower than λ_2, λ_3, and closer to the remaining eigenvalues. Therefore, λ_4 may not be counted as the number of clusters. Thus, it is possible to conclude $k = 3$ by Theorem 4. Strictly speaking, this is also an acceptable estimation. Thus, the onus is on the user to judge the actual value of k, which is really problem-specific as illustrated in the Section VI.

There are many methods to estimate the number of clusters in a data set. To date, most require choosing an appropriate clustering algorithm, e.g., k-means, where it is ran multiple times with predefined cluster numbers from 2 to k_{max}. The optimum k is then obtained by an internal indexed based on the clustering outcome. The key difference between these methods is in the index used. In this experiment, we compared our results to 3 widely used statistical methods [17] on 3 well-known clustering algorithms (see Table III).

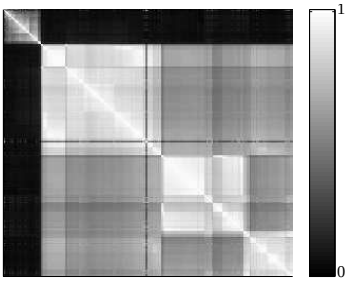
From the table, we found that all 3 indexes managed to get only 1 out of the 3 estimations right. Although the Hart index correctly estimated 4 clusters in news5, it fails to handle the conceptually disjoint web2 and classic4. Worse, most of the estimation are way off-track. For example, the KL index predicted $k = 29$ on web2! Furthermore, to the best of our knowledge, many of these methods report results on data sets with low dimensionality. Hence, our experiment reveals how sensitive these indexes are to the choice of clustering algorithms and the dimensionality of the data set.

In comparison, our approach (based on the 1-spectra of $\mathcal{G}(\mathbf{S})$) is independent of any clustering algorithm, is well-suited to high dimensional data sets, has low complexity (see Section VII), and can be easily implemented with existing packages (see Section VII). Clearly, our proposal remarkably outperforms all 3 methods in terms of speed and accuracy.

IV. p-NORMALIZATION ($p > 1$) SPECTRA ANALYSIS

Although the 1-spectra of $\mathcal{G}(\mathbf{S})$ gives a good estimate of k for a given data set and similarity measure, this information may be too restrictive for users with different analytical requirements. Due to subjectivity or domain expertise, it is possible that some overlapping clusters are viewed as one big clusters on one instance but not on another. In this case, providing a single value for k may be inappropriate.

As an example, Figure 4 shows a text collection with 4 topics. At the high-level, there are two conceptually disjoint clusters, i.e., $\langle\{T_1\}, \{T_2, T_3, T_4\}\rangle$, and among the four topics within the second cluster, each topic can be viewed as being overlapping or completely disjoint. On some instances, the user may choose $k = 2$ as the "right" cluster numbers when looking at the data from a high-level concept. On another occasion, the user may perceive $k = 4$ (i.e., $\langle\{T_1\}, \{T_2\}, \{T_3\}, \{T_4\}\rangle$) as the "right" value when the data is dealt with at a low-level concept.



(a)

Topic	Topic content
T_1	rec.motorcycles
T_2	comp.sys.mac.hardware
T_3	comp.os.ms-windows.misc
T_4	comp.windows.x

(b)

Fig. 4. A text collection of 275 new posters selected from `newsgroup` with four topics that overlaps at high-level concepts: (a) gray scale image of similarity matrix $\mathbf{S}$; (b) the topic of each cluster.

Given this situation, we now have a variation of our fundamental question: *In light of different analytical contexts, how can we suggest an appropriate value of k?* Our solution in this case is to provide the user with two possible values of k that is based on two different perspectives of the data set. To our knowledge, previous works rarely address this issue.

A. Differential Levels Between Clusters

Since the cause of ambiguity in the value of k lies in the overlapping of concepts, we need a method to characterize the degree of overlapping between clusters. To do so, we first define two measures that are used to compute this. Assume a similarity matrix $\mathbf{S}$ with the structure given in Eq.(3), we define the *signal* and *noise* level of an i-th column vector $\vec{s}_i$ below.

Definition 2 (Signal and noise level of $\vec{s}_i$): Given a similarity matrix $\mathbf{S}$ with structure similar to Eq.(3) where the i-th column vector $\vec{s}_i = (s_{1i}, \ldots, s_{ni})^T$ is in the area of $\mathbf{S}_{jj}$ $(1 \leqslant j \leqslant k)$, then all s_{li} $(l \in \mathbf{S}_{jj})$ are *signals* that contribute to the structure $\mathbf{S}_{jj}$ while all s_{li} $(l \notin \mathbf{S}_{jj})$ are noise that interfere with the structure $\mathbf{S}_{jj}$. We denote the signal level of $\vec{s}_i$ as $\mathrm{sig}(\vec{s}_i)$ and the noise of $\vec{s}_i$ as $\mathrm{noi}(\vec{s}_i)$ which are respectively defined in Eq.(14) and therefore, $\|\vec{s}_i\|_p = (\mathrm{sig}^p(\vec{s}_i) + \mathrm{noi}^p(\vec{s}_i))^{\frac{1}{p}}$ and typically $\mathrm{sig}(\vec{s}_i) \gg \mathrm{noi}(\vec{s}_i)$.

$$\mathrm{sig}(\vec{s}_i) = (\sum_{l \in \mathbf{S}_{jj}} s_{li}^p)^{\frac{1}{p}}, \quad \mathrm{noi}(\vec{s}_i) = (\sum_{l \notin \mathbf{S}_{jj}} s_{li}^p)^{\frac{1}{p}} \tag{14}$$

Lemma 2 (Properties of $\mathrm{sig}(\vec{s}_i)$): Let n_j be the size of signals (i.e., the number of rows or columns of $\mathbf{S}_{jj}$) of $\mathbf{S}_{jj}$, we define $\overline{\mathrm{sig}(\vec{s}_i)} = \frac{1}{n_j}\sum_{l \in \mathbf{S}_{jj}} s_{li}$ as the average of signals in $\mathbf{S}_{jj}$, and $\max(\mathrm{sig}(\vec{s}_i)) = \max_{l \in \mathbf{S}_{jj}}\{s_{li}\}$ as the maximum of signals in $\mathbf{S}_{jj}$. We have the following properties: (i) $\|\vec{s}_i\|_p \approx \mathrm{sig}(\vec{s}_i)$; (ii) when $p \to 1$, $\|\vec{s}_i\|_p \to \overline{\mathrm{sig}(\vec{s}_i)} n_j$; (iii) when $p \to \infty$, $\|\vec{s}_i\|_p \to \max(\mathrm{sig}(\vec{s}_i))$.

Proof: Because $\|\vec{s}_i\|_p = (\mathrm{sig}^p(\vec{s}_i) + \mathrm{noi}^p(\vec{s}_i))^{\frac{1}{p}}$ and $\mathrm{sig}(\vec{s}_i) \gg \mathrm{noi}(\vec{s}_i)$, we can easily get (i). As $\mathrm{sig}(\vec{s}_i) = \overline{\mathrm{sig}(\vec{s}_i)} n_j$ when $p = 1$, we get (ii) according to (i). According to the properties of p-norm of vectors as shown in proof of Theorem 1, when $p \to \infty$, $\mathrm{sig}(\vec{s}_i) \to \max(\mathrm{sig}(\vec{s}_i))$ and we get (iii) according to (i). ■

Notice that the signal level of the i-th column vector actually describes the cluster $\mathbf{S}_{jj}$ where $\vec{s}_i$ resides. This conclusion can be drawn by the observation where nearly all column vectors in the same cluster exhibit a similar signal level. Hence, extending Definition 2 gives us the signal level of a cluster.

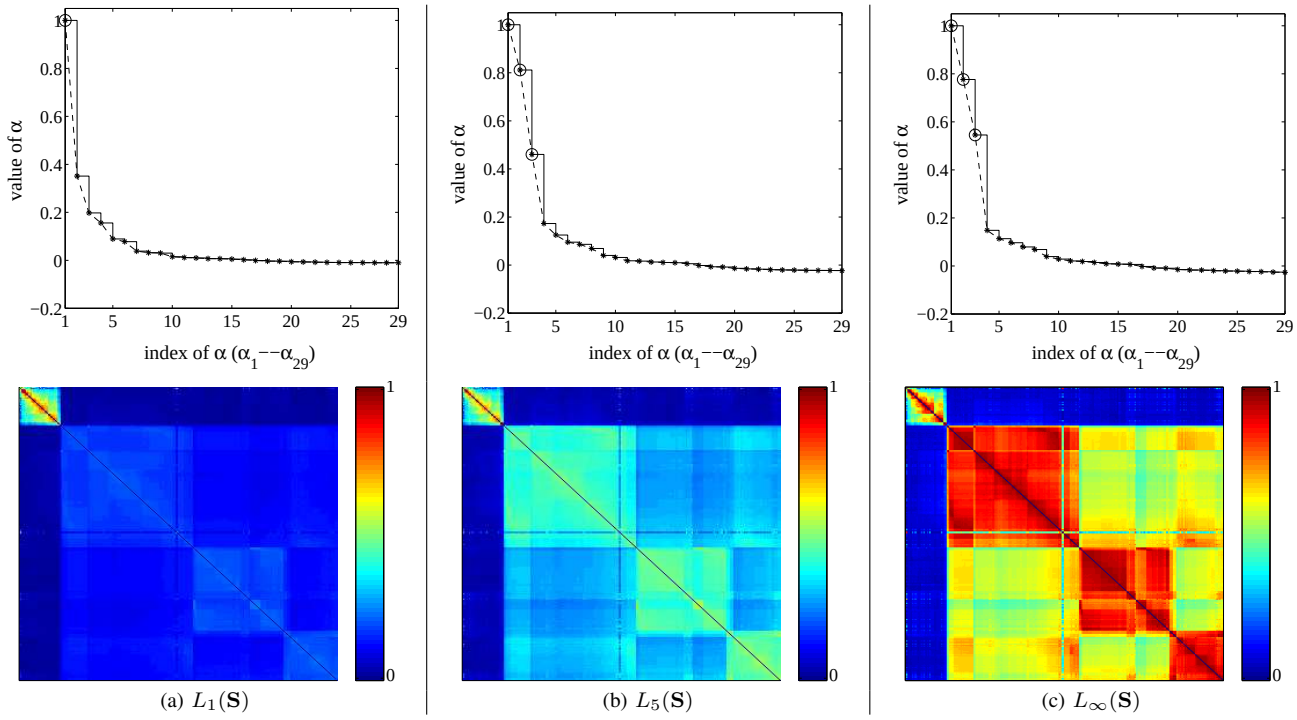
Definition 3 (Signal level of a cluster): Given a similarity matrix $\mathbf{S}$ with a structure similar to Eq.(3), the signal level of a cluster $\mathbf{S}_{jj}$ $(1 \leqslant j \leqslant k)$ is given as

$$\mathrm{sig}(\mathbf{S}_{jj}) = \frac{1}{n_j}\sum_{i \in \mathbf{S}_{jj}} \mathrm{sig}(\vec{s}_i) \tag{15}$$

The definition of $\mathrm{sig}(\mathbf{S}_{jj})$ is straightforward. We simply take the mean of the signal levels of all the row vectors in $\mathbf{S}_{jj}$. This is possible because the signal level of a cluster can be approximated by its column vectors. Based on this concept, we derive the following lemma that tells us how L_p normalization makes $\mathbf{S}$ meaningful in subsequent spectra analysis.

Theorem 5 (Differential level of L_p normalization): Let a and c be entries from $\mathbf{S}$ with structure similar to Eq.(3). We let a and c represent two possible cases – whether the entry is a diagonal or non-diagonal sub-matrix. Further, suppose a is in $\mathbf{S}_{11}$ and c is in $\mathbf{S}_{12}$. Without loss of generality, we permutate $\mathbf{S}$ to make the sub-matrices containing a and c in the place of $\mathbf{S}_{11}$ and $\mathbf{S}_{12}$. Then, we select one of the entries in $\mathbf{S}_{22}$ which we denote as b. After p-normalization of $\mathbf{S}$, a', b' and c' in $L_p(\mathbf{S})$ corresponds to a, b and c in $\mathbf{S}$ respectively. We therefore have the following conclusions: (i) if $\mathrm{sig}(\mathbf{S}_{11}) \approx \mathrm{sig}(\mathbf{S}_{22})$, then $a' : b' : c' \approx a : b : c$; (ii) if $\mathrm{sig}(\mathbf{S}_{11}) \ll \mathrm{sig}(\mathbf{S}_{22})$, then $\frac{a'}{b'} \gg \frac{a}{b}$, $\frac{c'}{a'} \ll \frac{c}{a}$ and $\frac{c'}{b'} \gg \frac{c}{b}$.

Proof: According to Definition 1,



(a) $L_1(\mathbf{S})$ (b) $L_5(\mathbf{S})$ (c) $L_\infty(\mathbf{S})$

Fig. 5. The color images of $L_p(\mathbf{S})$ and the α sequences (see Theorem 4, $\alpha_i = \frac{\lambda_{i+1}}{\lambda_2}$) induced from their p-spectra respectively.

$$a' = \frac{a}{\sqrt{\|\vec{s}_{a1}\|_p}\sqrt{\|\vec{s}_{a2}\|_p}}, b' = \frac{a}{\sqrt{\|\vec{s}_{b1}\|_p}\sqrt{\|\vec{s}_{b2}\|_p}}, c' = \frac{c}{\sqrt{\|\vec{s}_{c1}\|_p}\sqrt{\|\vec{s}_{c2}\|_p}} \tag{16}$$

where $\vec{s}_{a1}$, $\vec{s}_{b1}$ and $\vec{s}_{c1}$ represent the row vectors containing the entries a, b and c respectively; likewise for representation for column vectors $\vec{s}_{a2}$, $\vec{s}_{b2}$ and $\vec{s}_{c2}$. By Lemma 2(i) and the fact that we can approximate $\text{sig}(\vec{s}_i)$ by $\text{sig}(\mathbf{S}_{jj})$, we have

$$a' = \frac{a}{\sqrt{\text{sig}(\mathbf{S}_{11})}\sqrt{\text{sig}(\mathbf{S}_{11})}}, b' = \frac{b}{\sqrt{\text{sig}(\mathbf{S}_{22})}\sqrt{\text{sig}(\mathbf{S}_{22})}}, c' = \frac{c}{\sqrt{\text{sig}(\mathbf{S}_{11})}\sqrt{\text{sig}(\mathbf{S}_{22})}} \tag{17}$$

and

$$\frac{a'}{b'} = \frac{a}{b}\cdot\frac{\text{sig}(\mathbf{S}_{22})}{\text{sig}(\mathbf{S}_{11})}, \frac{c'}{b'} = \frac{c}{b}\sqrt{\frac{\text{sig}(\mathbf{S}_{22})}{\text{sig}(\mathbf{S}_{11})}}, \frac{c'}{a'} = \frac{c}{a}\sqrt{\frac{\text{sig}(\mathbf{S}_{11})}{\text{sig}(\mathbf{S}_{22})}} \tag{18}$$

and hence, we can easily get conclusions when $\text{sig}(\mathbf{S}_{11}) \approx \text{sig}(\mathbf{S}_{22})$ and $\text{sig}(\mathbf{S}_{11}) \ll \text{sig}(\mathbf{S}_{22})$. ■

Corollary 1 (Differential level of L_1): In the same context as Theorem 5, when $p \to 1$: (i) if $n_1 \approx n_2$, then $a' : b' : c' \approx a : b : c$; (ii) if $n_1 \ll n_2$, then $\frac{a'}{b'} \gg \frac{a}{b}$, $\frac{c'}{a'} \ll \frac{c}{a}$ and $\frac{c'}{b'} \gg \frac{c}{b}$; where n_1 and n_2 are the size of $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ respectively.

Corollary 2 (Differential level of L_∞): Again from Theorem 5, we have the following when $p \to \infty$: (i) if $\overline{\max(\text{sig}(\mathbf{S}_{11})} \approx \overline{\max(\text{sig}(\mathbf{S}_{22})}$, then $a' : b' : c' \approx a : b : c$; (ii) if $\overline{\max(\text{sig}(\mathbf{S}_{11})} \ll \overline{\max(\text{sig}(\mathbf{S}_{22})}$, then $\frac{a'}{b'} \gg \frac{a}{b}$, $\frac{c'}{a'} \ll \frac{c}{a}$ and $\frac{c'}{b'} \gg \frac{c}{b}$; where $\overline{\max(\text{sig}(\mathbf{S}_{11})} = \frac{1}{n_1}\sum_{i\in\mathbf{S}_{11}} \max(\text{sig}(\vec{s}_i))$ is the average of all maximums of column vectors in $\mathbf{S_{11}}$, and likewise for $\overline{\max(\text{sig}(\mathbf{S}_{22})}$.

From Corollary 1 and 2, we can now quantify the degree of overlapping in the clusters by applying the L_1 and L_∞ operations. To better understand this, we revisit our example in Figure 4. Let us first look at the results of $L_1(\mathbf{S})$ shown in Figure 5(a). Here, we have opted to display the figures in color to better illustrate the differential levels between clusters. If we observe $\mathbf{S}$ in Figure 4, the last 3 topics (i.e., T_2, T_3 and T_4) are highly overlapping and being rather disjoint against T_1. Therefore, it is possible to view them as 1 unique cluster T_{234}. This point can also be observed by the differentiation level between T_1 and T_{234} in $L_1(\mathbf{S})$, which reflects the differentiation property of L_1. Since the size of T_1 is much less than T_{234}, we have Point 2 of Corollary 1 where nearly all entries of T_1 (which are close to 1) in $L_1(\mathbf{S})$ are greater than those of T_{234} – Theorem 1. This implies that the differentiation level of T_1 and T_{234} (shown by their color in Figure 5(a)) has "dramatically" changed leading to a signal for T_1 being greater than T_{234} after applying the L_1 operation to $\mathbf{S}$.

To see the differentiation level between T_2, T_3 and T_4, we recursively apply Corollary 1 to T_{234}. Since they have similar sizes, they satisfy Point 1 of Corollary 1. Thus, all entries in and between T_2, T_3 and T_4 of $L_1(\mathbf{S})$ maintain a similar level of

Algorithm 1 (DkA): Detects the number of distinct large eigenvalues

Input : $\lambda_2, \ldots, \lambda_{l+1}$, the sequence with ℓ elements, i.e., the p-spectra
Output : k, the number of distinct largest eigenvalues

generate α sequence $\alpha_i = \frac{\lambda_{i+1}}{\lambda_2}$ where $(1 \leqslant i \leqslant \ell)$
select $\alpha_{\mathcal{O}} = \alpha_1$ and $\alpha_{\mathcal{N}} = \alpha_l$ from sequence α to represent the outliers and normal sets respectively
repeat
  for all remaining α_i **do**
    assign α_i to the nearest representative, i.e., make $\alpha_i \in \mathcal{O}$ or $\alpha_i \in \mathcal{N}$
  end for
  select a non-representative α_{nr} at random
  compute the total costs $S_{\mathcal{O}}$ of swapping $\alpha_{\mathcal{O}}$ with α_{nr} and $S_{\mathcal{N}}$ of swapping $\alpha_{\mathcal{N}}$ with α_{nr}
  if $\min(S_{\mathcal{O}}, S_{\mathcal{N}}) < 0$ **then**
    swap the representative whose $S = \min(S_{\mathcal{O}}, S_{\mathcal{N}})$ with α_{nr} to form the new $\alpha_{\mathcal{O}}$ or $\alpha_{\mathcal{N}}$
  end if
until no further changes in $\alpha_{\mathcal{O}}$ and $\alpha_{\mathcal{N}}$
return $k = |\mathcal{O}| + 1$, where $|\mathcal{O}|$ is the number of elements in the set $\mathcal{O}$

differentiation as they are in $\mathbf{S}$. When rendering the differentiation levels, this small level of differentiation results in a similar color tone for most entries in T_{234} as Figure 5(a) depicts along with its α value curve of $L_1(\mathbf{S})$, which indicates 2 clusters. Therefore, the L_1 operation magnifies the differentiation levels between disjoint clusters, while "downplaying" those clusters that overlaps by obscuring their signal and noise levels, as the total size of overlapping clusters is often larger than the size of single disjoint cluster. Therefore, the 1-spectra has the ability to find the natural number of clusters that the user can use to have an insight on the number of disjoint clusters in the data. In this example, examining the collections in T_2, T_3 and T_4 will reveal that they belong to the newsgroup "comp" which shares many of the keywords that are not in T_1. Depending on the purpose of analysis, $k = 2$ can be the "right answer".

So that the users can count the number of overlapping clusters overlooked by the 1-spectra, we have the p-spectra ($p \gg 1$). Using the same example but with the p-normalization ($p > 1$), we will show how Corollary 2 works. The color images of $L_5(\mathbf{S})$ and $L_\infty(\mathbf{S})$ are shown in Figure 5(b) and (c). Recall that in L_p operation, the cluster signal is determined by the average of all maximums of its column vectors. Therefore, $L_\infty(\mathbf{S})$ reveals the differentiation level between clusters by the maximum entry of each cluster. In this way, we can achieve a higher differentiation level to reveal smaller but conceptually more cohesive clusters. As Figure 5(c) shows, the differentiation level of T_2, T_3 and T_4 under $L_\infty(\mathbf{S})$ is more distinctive than under $L_1(\mathbf{S})$ in Figure 5(a).

For completeness, Figure 5(b) shows the effect of applying a $L_5(\mathbf{S})$ on the same data. The result of $L_5(\mathbf{S})$ gives a higher differentiation level to T_2, T_3 and T_4 as compared to $L_1(\mathbf{S})$ but less than that of a $L_\infty(\mathbf{S})$ operation. This gradual change from $p = 1$ to $p = \infty$ validates the effectiveness of 1-spectra and ∞-spectra for our purpose of estimating k under different analysis context. We will report more empirical results in Section VII to support this argument.

V. Automatic Detection of k

Until now, we have shown how to determine k by counting the number of large eigenvalues after applying $L_p(\mathbf{S})$. In practice, our approach should operate automatically in the background so that the user can focus on the actual analysis. Therefore, it is important that our proposal can be automated such that k is provided to the user in a non-intrusive fashion. Fundamental to this is an algorithm that can decide the value of k once the analysis is carried out.

Recall from Theorem 4, the first k largest α values (or the first k+1 largest eigenvalues) have two properties. First, they form a large gap from the rest of the eigenvalues which are almost equal in their magnitude. Second, the number of large eigenvalues is always smaller than the number of remaining small eigenvalues since it is practically possible to assume that the number of clusters is always less than the number of distinct data points. From these two properties, we can actually perceive the first k large α values as "outliers" from the rest of the α values in the p-spectra sequence. Then, a simple strategy can be done to identify these "outliers". We first choose two representatives: $\alpha_{\mathcal{O}}$ to represent the "outliers" and $\alpha_{\mathcal{N}}$ to represent the remaining eigenvalues. We then assign the remaining α values to either $\alpha_{\mathcal{O}}$ or $\alpha_{\mathcal{N}}$ depending on the α value's distance to a representative. We repeat this process until $\alpha_{\mathcal{O}}$ and $\alpha_{\mathcal{N}}$ reaches equilibrium, as shown in Algorithm 1.

In practice, this algorithm is efficient because ℓ is far less than the number of rows or columns of the matrix n. This is because most α values in the tail of the descending spectrum curve are redundant to the detection of k. Therefore, the number of clusters would be less than 10% of the data points in real-life and hence, setting $\ell < 0.1n$ would be sufficient for detection.

VI. A Motivating Example

In this section, we discuss an example of how the theoretical observations discussed earlier work to close the gap between the user and the machinery. For illustration, we assume that the user is given some unknown collection.

TABLE IV

THE TEXT COLLECTIONS USED TO COMPUTE THE p-SPECTRA.

Index	Source	# Classes	# Doc.
a	TREC (tr12)	8	313
b	Reuters-21578 (re0)	13	1504
c	San Jose Mercury, TREC (hitech)	6	2301
d	CLUTO (mm)	2	2521

TABLE V

THE SAME EXPERIMENT CONDUCTED IN TABLE III ON ADDITIONAL DATA SETS. THE SUBSCRIPTS OF EACH DATA SET INDICATES THE POSSIBLE CLASS NUMBERS, E.G., mm CAN HAVE 2, 4 OR 5 CLUSTERS.

	Bisecting k-means			Graph-based			Hierarchical		
	CH	KL	Hart	CH	KL	Hart	CH	KL	Hart
tr12 $_{8}$	5	28	1	10	23	2	4	28	1
re0 $_{5,6,13}$	2	13 ($\surd$)	8	23	11	2	6 ($\surd$)	18	2
hitech $_{2,3,6}$	2 ($\surd$)	23	22	2 ($\surd$)	18	12	3 ($\surd$)	28	1
mm $_{2,4,5}$	5 ($\surd$)	20	19	6	28	10	3	25	1

If the user does not have pre-existing knowledge of the data, there is a likelihood of not knowing where to start. In particular, all clustering algorithms directly or indirectly require the parameter k. Without spectral analysis, the user is either left guessing what value of k to start with; or expend time and effort to find k using one of the existing estimation algorithm. In the case of the latter, the user has to be careful in selecting a clustering algorithm and in setting k_{max} (see Sub-Section III-C) – if it's set too high, the estimation algorithm takes a long time to complete; if it's set too low, the user risks missing the actual value of k.

In contrast, our proposal allows the user to obtain an accurate value of k without setting k_{max}. And in situations where the data is sufficiently complex, our proposal can also suggest alternate values of k depending on the user's judgement and the data in question – if the data has some highly overlapping clusters, our proposal can provide a macro-view and micro-view of the data by the 1-spectra and ∞-spectra. In either case, the performance of our approach is almost instantaneous when compared to methods that require a clustering algorithm. We believe this is important if the user's role is to analyze the data instead of waiting for the machinery. Once an initial value of k is decided, the user can commence clustering. Unfortunately, this isn't the end of clustering in real-life.

Upon obtaining the outcome, the user usually faces another question: *What is the quality of this clustering?* In our opinion, there is no knowledge discovery when there is no means to judge the outcome. As a result, it is also at this stage where interactivity becomes important. On this issue, some works propose the use of constraints. However, it is difficult to formulate an effective constraint if the answer to the above is unknown. This is where spectral analysis plays a part. By Theorem 3, the user is given feedback about the cluster quality. At the same time, grey images (e.g., Figure 2(b)) can also be constructed to help the user gauge the outcome.

Depending on the feedback, the user may then wish to adjust k, or use another similarity measure. In either case, the user is likely to make a better decision with this assistance. Once the new parameters are decided, another run of the clustering algorithm begins. Our proposal would then kick in at the end of each run to provide the necessary feedback to the user via Theorem 3. This interaction exists because different clustering objectives can be formulated on the same data set. At some point, the user may group overlapping concepts in one class. Other times, the user may prefer to separate them. In this aspect, our approach is non-intrusive and works in tandem with the user's intentions.

VII. ADDITIONAL EMPIRICAL RESULTS

So far, we have provided some initial empirical results to support our proposal. In this section, we report further results on more data sets from different sources. These data sets are more complex and varied in domain to demonstrate the effectiveness of our approach. Their characteristics are summarized in Table IV and VI.

A. Text Data

Our first test is to use the text collections (listed in Table IV) generated by the CLUTO toolkit [24]. This text collection is more complex than the one used in Table II in terms of the cluster size and the hierarchical structures embedded among clusters. Furthermore, this collection is interesting in the sense that the number of classes provided is actually different from the natural and actual number of clusters – a reflection of the subjectivity of k in the real-world.

Figure 6(a_1) and (a_∞) shows the spectra analysis on the tr12 collection. For this collection, there are 8 predefined classes and that tallies with the results of our analysis using both L_1 and L_∞ operations (note that the 8th class at the lower right of

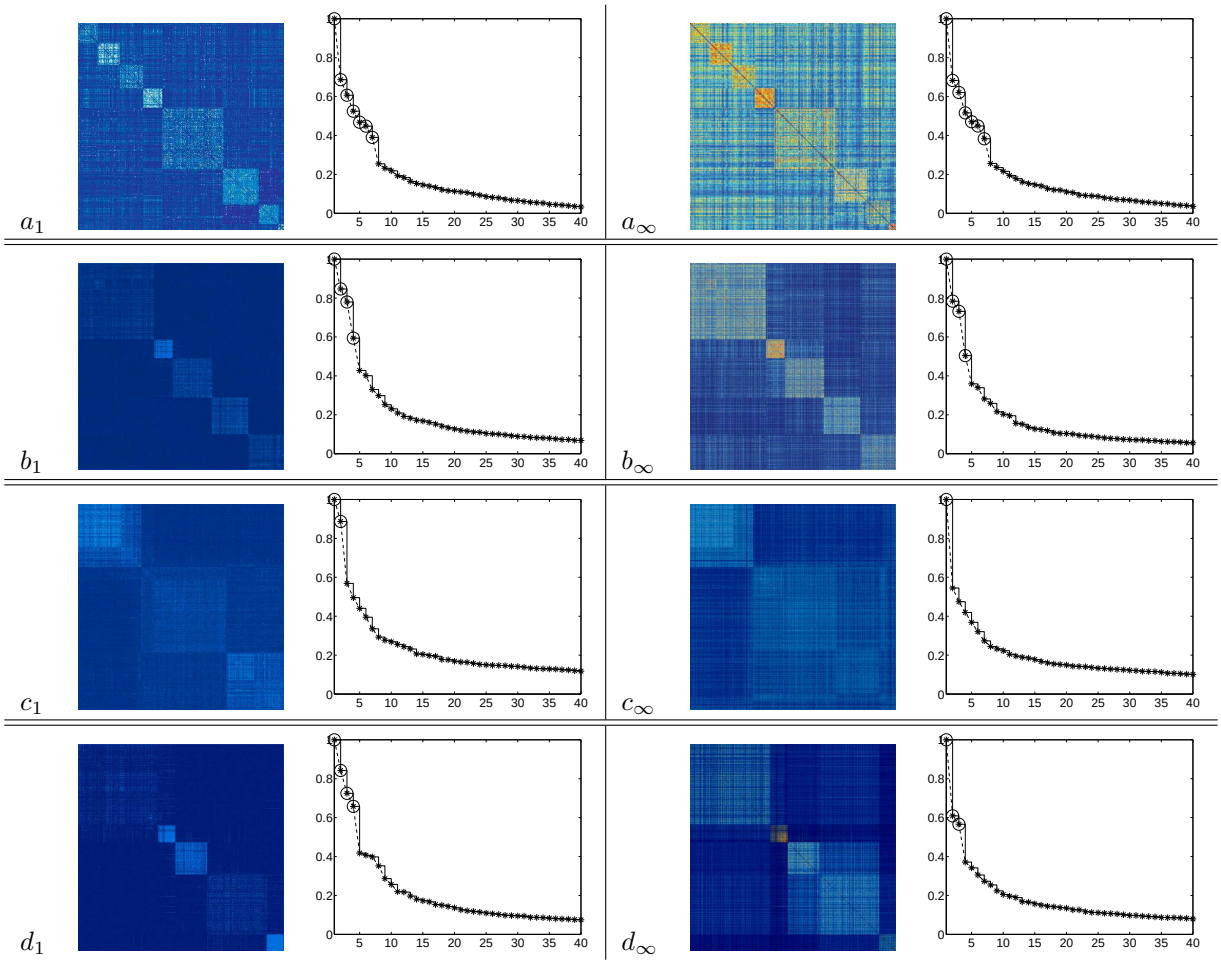


Fig. 6. The color Images of $L_p(\mathbf{S})$ and the α sequences (refer to Theorem 4, $\alpha_i = \frac{\lambda_{i+1}}{\lambda_2}$) induced from their p-spectra, respectively. The left depicts the color image and α sequences of $L_1(\mathbf{S})$, and the right depicts the color image and α sequences of $L_\infty(\mathbf{S})$: (a) `tr12`; (b) `re0`; (c) `hitech`; (d) `mm`.

the image is very small). In this case, there is no ambiguity in the number of clusters since there is no obvious hierarchical structure present in the image of $\mathbf{S}$, despite some noise being detected. Therefore, we can conclude that `tr12` is likely to be conceptually cohesive even without examining the documents. For situations like this, the analysis would be relatively straightforward.

The `re0` collection is more interesting than `tr12`. Although `re0` has 13 predefined classes, most of the clusters are small with some having less than 20 documents while a few classes (*money*, *trade* and *interest*) made up 76.2% of documents in `re0`, i.e., the remaining 10 classes contain 23.8% of the documents. This is made more challenging when the 10 classes are highly related. Hence, our analysis only shows 2 clusters for these 10 classes giving us a total of 5 clusters for the similarity matrix used. Furthermore, our analysis also reveals no obvious hierarchical structures in the collection. Unlike our study of `tr12`, `re0` goes on to show the possibility where the analysis may differ from the predefined class labels. Of course, the crucial point here is to appreciate that both are valid because of different analysis contexts, and that this should only serve as a reference to the actual analysis that the user is conducting. Figure 6(b_1) and 6(b_∞) shows the image and α sequences of the 1-spectra and ∞-spectra in which we derive the above conclusions.

There are 6 predefined classes in the `hitech` collection namely, *computer*, *electronics*, *medical*, *health*, *research*, and *technology*. The results of the L_1 operation shows 3 clusters as shown in Figure 6(c_1) while in Figure 6(c_∞), we have the results of the L_∞ spectra showing the second clusters to contain 3 highly correlated topics, i.e., *computer*, *electronics* and *technology*. For simplicity, we collectively refer these 3 topics as "cet" and can, as a matter of fact, be viewed as a single cluster. Likewise, topics like *health* and *medical* are also highly correlated and therefore, can be viewed as another cluster which we shall refer to as "mh". Thus, if we now look at the images in Figure 6(c_∞), we can see a clear hierarchical structure consisting of the class "mh" and *research*. If we look at them closely, we can conclude that they are 2 distinct clusters when taking the 1-normalization on $\mathbf{S}$, and when ∞-normalization on $\mathbf{S}$ is applied, $L_\infty(\mathbf{S})$ concludes a single cluster. Thus, our method signals the existence of a clear hierarchical structure within this collection. Given that the other classes do not exhibit an obvious hierarchical structure, the possible number of clusters can be 2 or 3 depending on the analysis requirements. Again,

TABLE VI

THE GENE EXPRESSION DATA SETS USED TO COMPUTE THE p-SPECTRA.

Name	Source	# Genes	# Conditions	# Classes
Leukemia	Study of cancer classification [12]	7129	72	2
Yeast	Study of budding yeast saccharomyces cerevisiae [10]	64	79	4
Serum	Study of the response of human fibroblast to serum [13]	517	12	6
NCI60	National Cancer Institute's (NCI) anti-cancer drug screen [11]	1161	60	4
Monocytes	Peripheral blood monocytes [20]	2329	139	18

TABLE VII

SIMILARITY MEASURES USED IN THE GENE EXPRESSION DATA.

Name	Formula	Note
Pearson correlation coefficient	$\frac{1}{2}\ \frac{(\vec{x}-\bar{\vec{x}})'(\vec{y}-\bar{\vec{y}})}{\|\vec{x}-\bar{\vec{x}}\|\|\vec{y}-\bar{\vec{y}}\|}+1$	The average feature value of $\vec{x}$ over all dimensions is denoted by $\bar{\vec{x}}$. To get the similarity values to within the range of $[0,1]$, we apply the translation operator to each similarity value before halving them.
Extension of Euclidean distance	$\exp(-\frac{\|\vec{x}-\vec{y}\|^2}{\sigma})$	To get the similarity values to within the range of $[0,1]$, we apply exponential function to the scaled Euclidean distance. The scale is controlled by a positive number σ.

this result differs from that of what was given in the predefined class labels.

The last text collection (mm) that we looked at contains 2 predefined classes: *music* and *movie*. The interesting result here is that we obtained 4 distinct clusters from our analysis instead of the predefined 2. When we did further checks, we found a clear hierarchical structure embedded in the image of $\mathbf{S}$ (enclosed in white squares in the figure). In this case, we demonstrate how our method can be used to reveal a hierarchical structure that is otherwise "hidden" by predefined classes given under a different context.

We shall conclude by checking the accuracy of our method on the above data sets against other measures similar to what we did in Section III-C. From Table V, we can see that although tr12 contains only 313 documents, i.e., a small data set, none of the indices tested can correctly estimate the cluster number. We attribute the rationale of this behavior to the high level of noise among clusters as we can see from the image given in Figure 6. On the other data sets, while some of the indices did successfully estimate the right number of classes, we found that their estimation are largely dependent on the specific clustering algorithm used. In all cases, none of the indices worked correctly in all 3 clustering algorithms on all data sets. The best result in this case is the CH index on hitech, which managed to estimate k to be either 2 or 3 – a result that agrees with our method and analysis. Interestingly, the CH index estimates $k = 3$ when the hierarchical clustering algorithm was used and $k = 2$ on the other partition-based algorithms. This reinforces our earlier observation of a hierarchical structure in hitech that our method reveals effectively.

B. Gene Expression Data

In this set of experiments, we selected 5 gene expression data sets to demonstrate the effectiveness of our approach on different *text* collections. For ease of comparison, we preprocess all the data sets by standardizing the object vectors with a mean of 0 and a standard deviation of 1. We also used similarity measures to compute the similarities between the genes or conditions. The details of the data sets are given in Table VII. In most cases, we found that the Pearson correlation coefficient is sufficient to develop the similarity matrix for effective clustering. If higher accuracy is desired, our experiments show that an extension of the Euclidean distance can be used as the similarity measure. Of course, this conclusion comes easily since we can substitute different similarity measures within the spectra analysis.

1) Leukemia Data: The Leukemia data set [12] comes from a study of gene expression of two types of acute Leukemias: Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). The data comprises 47 samples of ALL (of which 38 are B-Cell ALL and 9 are T-Cell ALL) and 25 samples of AML. There are no missing values and the condition vectors are standardized as described. From the above, we expect a good clustering algorithm to output 2 (ALL, AML) or 3 (ALL B-Cell, ALL T-Cell, AML) clusters.

We first tested the 72 samples by considering all 7129 genes and the Pearson correlation coefficient to compute the 72×72 similarity matrix. The 1 and ∞-spectra on this matrix is given in Figure 7(a). Interestingly, the λ sequences in both cases contain only 1 large eigenvalue while the others are nearly 0, i.e., all 72 samples belong to a big cluster. Clearly, this analysis appears "strange". On checking further, we realized that our analysis fails because we did not consider the domain knowledge that should be used to preprocess the data as Gloub et. al. [12] suggested: (i) thresholding, i.e., floor of 100 and ceiling of 16,000; (ii) filtering genes by excluding genes with $max/min \leqslant 5$ or $(max - min) \leqslant 500$ where max and min respectively refers to the maximum and minimum intensities for a particular gene across the 72 samples; and (iii) base-10 logarithmic transformation.

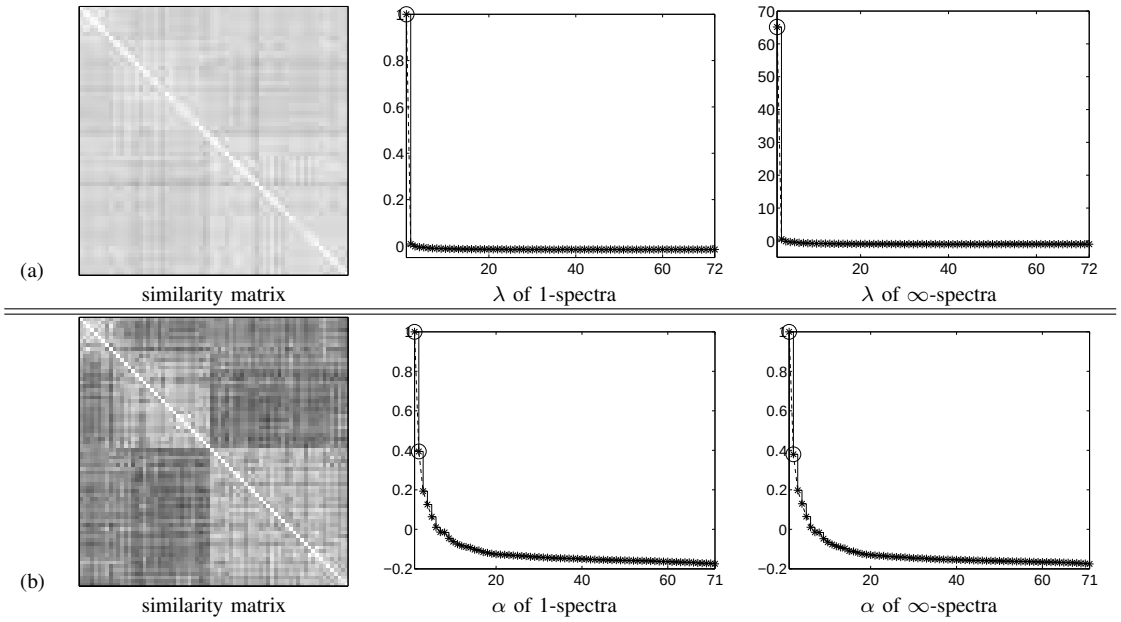


Fig. 7. The **Leukemia** data set analyzed with the Pearson correlation coefficient using the same similarity measure but different feature sets: (a) the complete gene set (7129 genes) and the eigenvalue curves of 1-spectra and ∞-spectra; (b) 100 selected genes and its α sequence of 1-spectra and ∞-spectra.

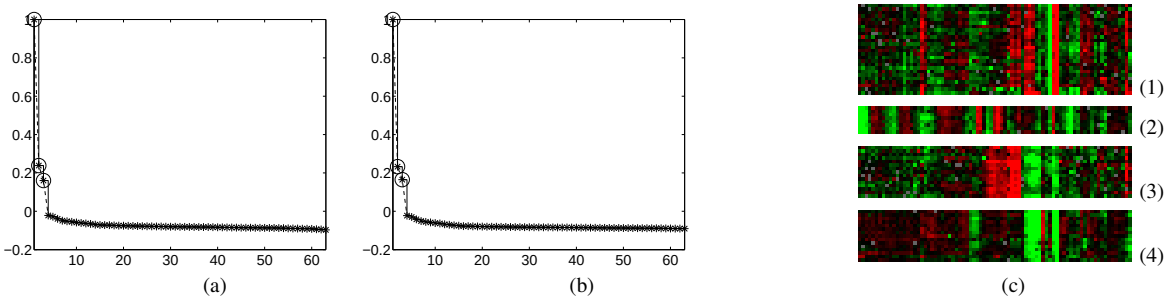


Fig. 8. The **budding yeast** gene expression data set: (a) α sequence of 1-spectra; (b) α sequence of ∞-spectra; (c) the grey scale plot where (1) is protein degradation, (2) is chromatin structure, (3) is protein synthesis and (4) is glycolysis.

After preprocessing, the data is summarized by a 72×3571 matrix. We then select the top 100 gene features having the largest standard deviation. We then standardize their mean and deviation and compute the similarity matrix as described earlier to obtain the α sequences of the 1 and ∞-spectra from this 72×100 matrix. The results, shown in Figure 7(b), now gives the correct answer that we expected. Meanwhile, the large gap between the first and second α values in Figure 7(b) suggests that it is also possible to have 2 clusters.

2) Budding Yeast Data: This data set is obtained from a study of gene expression in the *Saccharomyces cerevisiae* budding yeast during a diauxic shift [10] where each gene has 79 arrays. Each cell in the gene expression matrix represents the measured Cy5/Cy3 fluorescence ratio of the corresponding target element on the appropriate array. For our experiment, we selected 4 clusters (64 genes in total) sharing similar expression patterns, and are annotated along the same biological pathway. The 4 clusters are *protein degradation* (cluster C), *chromatin structure* (cluster H), *protein synthesis* (cluster F), and *glycolysis* (cluster E).

For simplicity of experimental setup, but without loss of accuracy, we do not impute the cells using complex and CPU-intensive techniques. Instead, we simply filled the cells without data in the gene expression matrix with zeros. Again, we used the Pearson correlation coefficient as the measure of similarity. From these preprocessing, we obtained the corresponding spectra in Figure 8.

By Theorem 4, the spectra graph suggests $k = 4$ in this particular data set, which corresponds to our experimental setup. At the same time, the image in Figure 8(b) suggests that it is also possible to have 2 clusters, i.e., $k = 2$. This possibility is correct since we specially selected the 4 clusters such that it is possible to put *protein synthesis* and *glycolysis* in one class and the other 2 into another.

3) Serum Data: This data set comprises the transcriptional response of human fibroblasts to serum. The details of this data set are given in [13], [22]. We selected 517 genes whose expressions vary in response to serum concentration in human

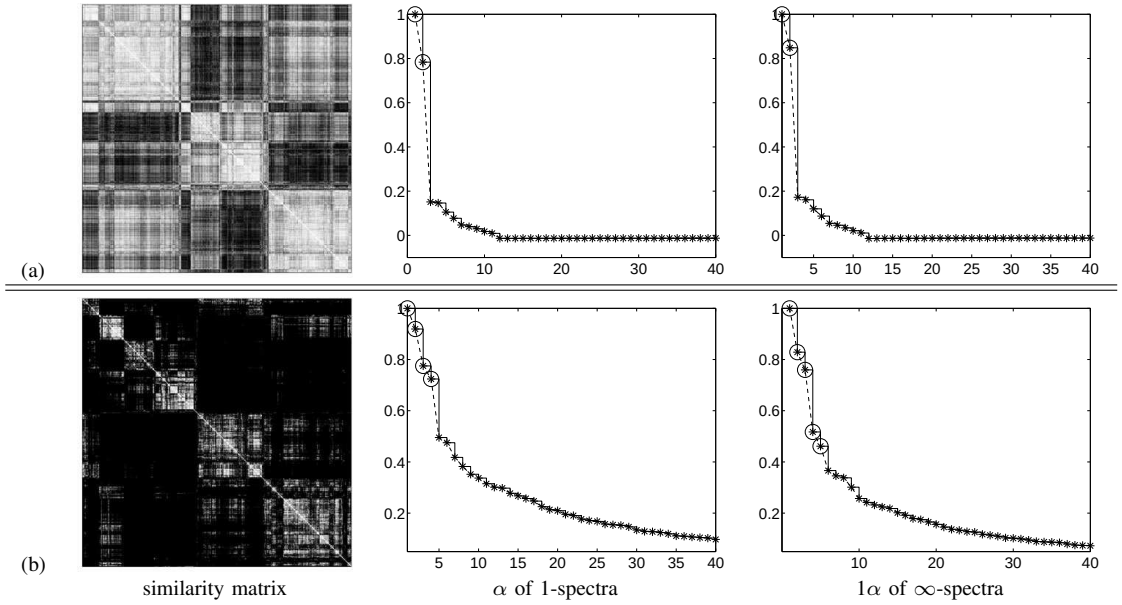


Fig. 9. The **serum** gene expression data set analyzed using different similarity measures: (a) Pearson correlation coefficient; (b) Euclidean distance with $\sigma = 3$. The gray image for the similarity matrix is reordered by the true class labels.

fibroblasts. Based on [22], 6 clusters were found and class labels for each gene were assigned[1]. There are no missing values and therefore, we simply standardize the mean and deviation. The similarity matrix is built using the Pearson correlation coefficient, and the results are shown in Figure 9(a).

Interestingly, the analysis reveals only 3 clusters, and there is no obvious hierarchical structures present. Given that there should be 6 clusters, we suspect that the Pearson correlation coefficient may be the cause. We therefore replaced it with a variant of the Euclidean distance. Performing the analysis with this measure, we have the results in Figure 9(b). This time round, we see that the number of clusters reported is 5, and the ∞-spectra suggests the presence of a hierarchical structure. This experiment therefore conveyed an important point: without sufficient domain knowledge, the user may have to use a number of similarity measure to compare the results before furthering other cluster analysis, or conclusions.

4) NCI60 Data: The cDNA microarrays was used to examine the variation in gene expression among the 64 cell lines from the National Cancer Institute's (NCI) anti-cancer drug screen [11]. This time, we selected a subset of 1,161 cDNAs measured across 64 cell lines similar to that described and used in [11]. For convenience, we reproduced the part of this information in Figure 10. As with the previous tests, we performed the same standardization and used the Pearson correlation coefficient as the similarity measure. The 1 and ∞-spectra of this data set are shown in Figure 10(b) and (c). Once again, the estimation from our analysis agrees with that given by Figure 10(a) and its discussion in the original paper.

5) Peripheral Blood Monocytes Data: Our last data set contains the hybridization fingerprints of 2329 cDNAs obtained from 139 oligonucleotide probes. This data set is part of a library of some 100,000 cDNAs prepared from purified peripheral blood monocytes that was used in [20]. The true clustering of these 2329 cDNAs is known from back hybridization experiments. – it contains 18 gene clusters varying in sizes from 709 to 1 as shown in [20]. The hybridization matrix (2329×139) and the class labels for each cDNA in the true clustering are available at `http://www.cs.tau.ac.il/~rshamir/expander`.

There are 2 characteristics in this data set that makes its clustering structure more complex than all of the previous gene expressions. First, there is a wide variation in the cluster size. In increasing order, the 18 clusters has sizes 1, 2, 10, 12, 14, 32, 39, 43, 67, 86, 91, 108, 146, 187, 213, 284, 285, and 709 which is also reflected in Figure 11. Second, there are many singletons (data that do not belong to any compact clusters). For example, the size of the first 2 clusters do not really constitute to a cluster per se and in all likelihood, may be seen as singletons. As a matter of fact, the analysis in [20], concludes the first 8 clusters as invalid clusters.

We verified this point by observing the absence of white blocks in the top-left corner of the 3 similarity matrices shown in Figure 11. Table VIII contains the excerpts of results in [20] to show further details of the clustering structure of this data set. Now, if we consider only the true clusters T_i, then there is only 10 clusters. If we consider the number of clusters C_i found in [20], then there are 13 clusters. And if we ignore the 2 small clusters C_{13} and C_{16} having only 6 data points, then we have only 11 clusters (or 12 depending on the preprocessing and similarity matrix's response to C_{13} and C_{16}). In all cases, we note that the above are possible values of k because of the data characteristics that is further complicated by the different outcomes in the preprocessing and similarity measure used.

[1] `http://cheed.nus.edu.sg/~chergs/ismb2004`

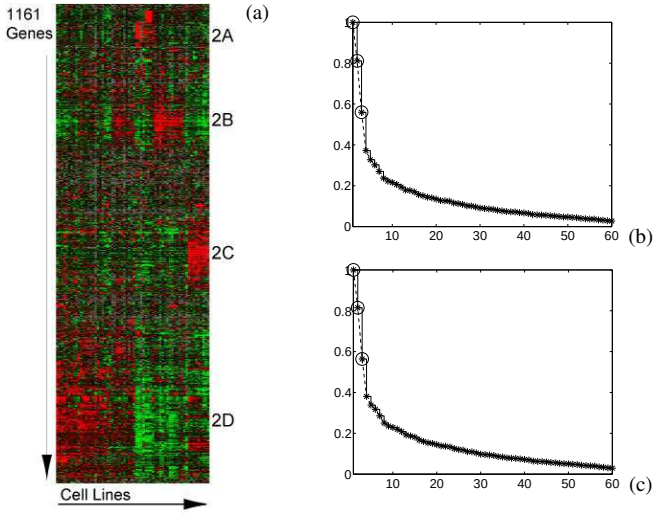


Fig. 10. The **NCI60** data set: (a) the grey scale plot at `genome-www.stanford.edu/nci60/figures.shtml`; (b) α sequence of 1-spectra; and (c) α sequence of ∞-spectra.

TABLE VIII

THIS TABLE IS EXCERPTED FROM TABLE 2 IN [20] WHERE ONLY THE ENTRIES IN THE ORIGINAL TABLE REPRESENTING PURE OR ALMOST PURE CLUSTERS (THE BOLDFACE NUMBERS) ARE SELECTED. IN THIS TABLE, T_i REPRESENTS THE TRUE CLUSTERS AND C_i REPRESENTS THE CLUSTERS FOUND BY [20].

	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}
C_2						162				
C_3					62					
C_4										563
C_5									199	
C_6		83								
C_7								224		
C_8				97						
C_9	42									
C_{10}							170			
C_{11}			61							
C_{13}	6									
C_{14}								26		
C_{16}								6		

In this experiment, we used the Pearson correlation coefficient and a variant of the Euclidean distance (with $\sigma = 50$) to compute 2 similarity matrices from the standardized input matrix (for cDNAs). We then conduct the analysis using these matrices to obtain the results in Figure 11(a) and (b). Next, we preprocessed the data according to [20] as follows: the hybridization matrix H has rows corresponding to the cDNA clones and columns corresponding to the probes. Each entry H_{ij} is the intensity level of the hybridization of clone i with probe j. The matrix H is used to construct the 2329×2329 similarity matrix S where $S_{ij} = \sum_k H_{ik}H_{jk}$. In S, each entry has a value in the range of 3.42 to 139. From S, the adjacency matrix $A_{2329\times 2329}$ is generated by setting $A_{ij} = 1$ if $S_{ij} \geqslant 110$ and $A_{ij} = 0$ otherwise. The resultant adjacency matrix therefore takes into account the domain expertise that in turn, implies that using A to estimate k is likely to give the best result. This is confirmed in Figure 11(c).

From Figure 11(a), we see that the Pearson correlation coefficient generates a high level of noise. Nevertheless, we can still detect 9 distinct clusters using our analysis, where the last 2 clusters T_{17} and T_{18} are blurred and can only be viewed as one class. The optimality of our approach is demonstrated when we incorporate multiple similarity measures in our analysis as shown in Figure 11(a) and (b). In this case, the value of k estimated ranges from 10 to 12 which agrees with our analysis in Table VIII. Some readers may find that the largest α values are not obvious in Figure 11. If we inspect the image rendered, we can see that some of the clusters are not "well-formed" and that leads to obscured α sequences. Even in this situation, our proposed detection algorithm manages to detect the largest α sequences correctly.

6) Comparison with Other Methods: We conclude our experiments by showing the effectiveness of our approach on the data sets in VI like we did in Section III-C and Section VII-A. We have also increased the number of validation indices to 5 (added Davies-Bouldin index [6] and Dunn index [9]), and the number of clustering algorithms to 4 (added CLICK [32] and MCLUST [15]). The results are shown in Table IX.

In this table, the Leukemia data set compromises of the 100 selected genes obtained by the preprocessing discussed in [12], and all of the data points have a mean of 0 and a variance of 1. Not surprising, most indices managed to successfully estimate the number of clusters in the Leukemia and Budding Yeast data sets because of the small number of data points (72 and 64 respectively), and their low dimensionality. Yet, there are also indices that failed to give a reasonable estimate, e.g., the KL

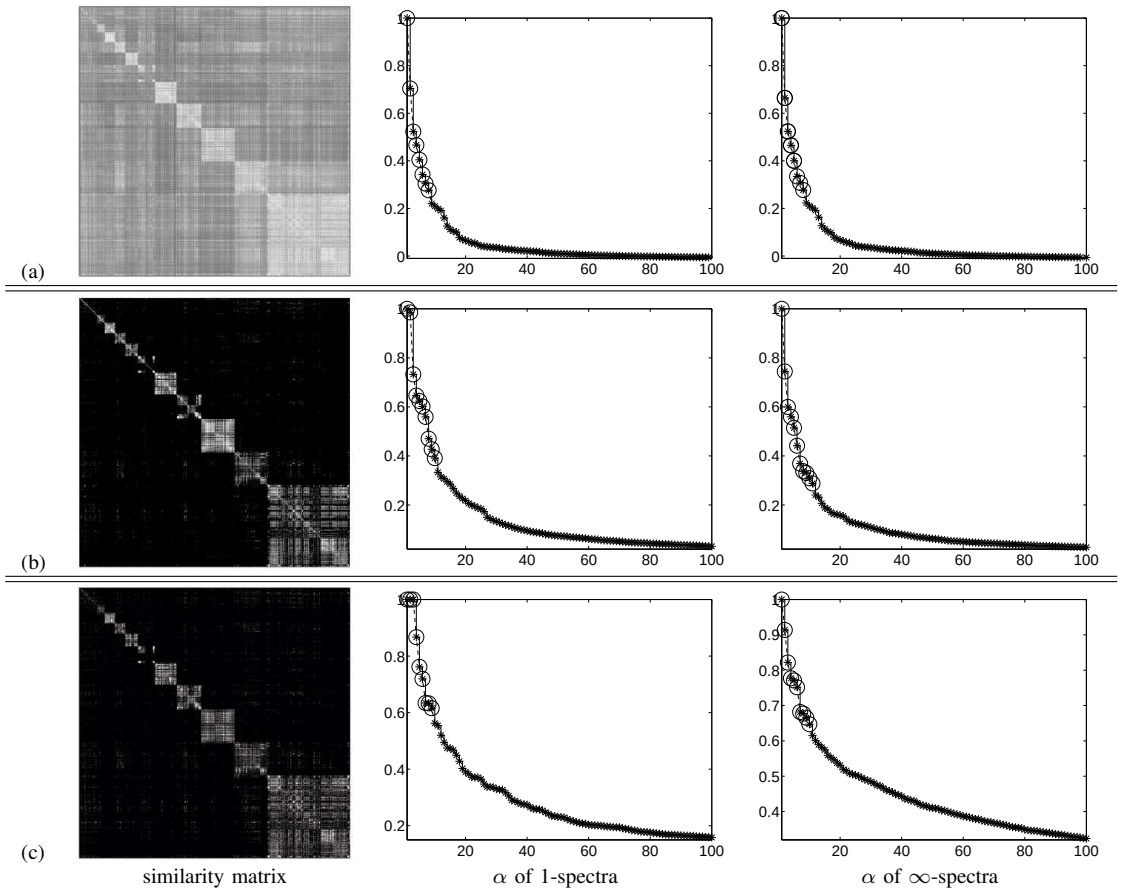


Fig. 11. The **monocytes** cDNA data set analyzed with different feature sets and similarity measures: (a) Pearson correlation coefficient resulting in 8 distinct α sequences in the 1-spectra and ∞-spectra respectively; (b) Extension of Euclidean distance with $\sigma = 50$ resulting in 10 distinct α sequences in the 1-spectra and 11 distinct α sequences in the ∞-spectra; (c) Processing techniques described in [20] followed resulting in 9 distinct α sequences in the 1-spectra and 10 distinct α sequences in the ∞-spectra.

TABLE IX

THE SAME EXPERIMENT CONDUCTED IN TABLE III AND TABLE V ON ADDITIONAL DATA SETS, INDICES AND ALGORITHMS. THE ADDITIONAL INDEXES ARE THE DAVIES-BOULDIN (DB) INDEX AND DUNN (DUNN) INDEX, AND THE ADDITIONAL ALGORITHMS ARE CLICK AND CLUST WITH k-MEANS (KM) AND HIERARCHICAL (HI). IN CLICK, THE NUMBER IN THE BRACKET REPRESENTS THE UNCLUSTERED SINGLETONS.

	CH		KL		Hart		DB		Dunn		CLICK	CLUST
	km	hi	km	hi	km	hi	km	hi	km	hi		
Leukemia $_{2,3}$	2 ($\surd$)	2 ($\surd$)	27	25	2 ($\surd$)	2 ($\surd$)	3 ($\surd$)	7	2 ($\surd$)	2 ($\surd$)	1 (0)	4
Yeast $_{4}$	4 ($\surd$)	4 ($\surd$)	25	1	4 ($\surd$)	4 ($\surd$)	6	4 ($\surd$)	2	2	3 (3)	1
Serum $_{6}$	3	3	30	2	10	10	7	8	3	3	4 (10)	1
NCI60 $_{4}$	2	2	28	22	7	9	2	2	2	2	6 (521)	1
Monocytes $_{9-16}$	3	3	28	18	18	14 ($\surd$)	7	3	2	2	16 (60) ($\surd$)	1

index. Worse, all the indices failed to give a correct estimate when we scaled out test to larger and higher dimensionality data sets. For example, none of the indices managed to give a correct value of k for the Serum and NCI60. For the Monocytes data set, only the Hart index managed to get the acceptable value of k when the hierarchical algorithm is use.

Finally, we also note that some of the indices are not stable, i.e., there is a large difference in the estimate when used with different clustering algorithms. For example, both KL and DB proved to be highly sensitive in our tests. In contrast, our approach is independent of clustering algorithms and do not fluctuate as much despite being dependent on the preprocessing and similarity measures used. Finally, the additional algorithms introduced also proved to be ineffective in estimating k with CLICK managing one right estimate in Monocytes, while CLUST failed in all cases.

C. Performance

We end this session by discussing the performance of our method. Suppose the symmetric matrix $\mathbf{S}$ has n rows and columns, and h non-zero entries, then the complexity of performing a p-normalization from $\mathbf{S}$ to $L_p(\mathbf{S})$ would be $O(h+n)$. The study of how to effectively compute the largest eigenvalues has been well-developed within the framework of symmetric

generalized eigen problems that arouse from structural analysis in physics and computational chemistry. Thus, a series of mature mathematical tools have been developed for our purpose.

In the case where the symmetric matrix is spare, i.e., h and n^2 do not have the same magnitude, we can use the `Lanczos` method to compute the eigenvalues in $k \ll n$ iterations allowing the process to converge quickly [16]. Since the complexity of each iteration is $O(h+n)$, the final complexity of this method is therefore $O(k(h+n))$. If we ignore the very small k in real computations, the complexity of our method becomes $O(h+n)$. There are fast `Lanczos` packages (e.g., LANSO [28], PLANSO [40]) for such computation.

If the symmetric matrix is dense, i.e., h and n^2 have the same magnitude, then the ARPACK package can be used [29]. This is a collection of `Fortran77` subroutines that is capable of computing a few (k) eigenvalues using $2nk+O(k^2)$ storage (and no auxiliary storage [29] required). In addition, the ARPACK can allow user-specified settings such as finding the largest eigenvalues by its magnitude. This is done through the definition of matrix-vector multiplication routines that can be tuned to compute eigenvalues in sparse matrices efficiently. Since our matrices are symmetric, the method used in ARPACK actually reduces to a variant of the `Lanczos` process known as Implicitly Restarted `Lanczos` Method (IRLM).

VIII. Related Work

Underlined by the fact that there is no clear definition of what is a good clustering [38], [23], the problem of estimating the number of clusters in a large data set is arguably a difficult one. Over the years, several approaches to this problem have been suggested. Among them, the more well-known ones include cross-validation [34], penalized likelihood estimation [3], [19], resampling [30], finding the 'knee' of an error curve [38], [37], Gap statistics [38], validity indices under the framework of cluster validation [18], and criteria such as the Baysian Information Criterion (BIC) for model selection capability and likelihood-based clustering algorithms, e.g., MCLUST [15].

The problem with these techniques is that they either make a strong parametric assumption, or they are computationally expensive. For example, both cross-validation and penalized likelihood estimation require some form of input parameters, e.g., the number of cross validations or the MML. On the other hand, techniques such as resampling and finding the 'knee' of an error curve are examples of CPU-intensive methods. In resampling, the natural number of clusters are discovered by repeated clustering of samples drawn from the original data set; while in the case of finding the 'knee' of an error curve, each potential value of k requires a run of the clustering algorithm. In cluster validity indices and Gap statistics, runs of the algorithm on the target data (and in the case of Gap statistics, reference data as well) with a range of k values are needed to provide the necessary information for analysis. Furthermore, the model selection criteria such as BIC must work along with certain likelihood-based clustering algorithms, e.g., MCLUST, by computing BIC values for each model.

Unfortunately, these methods become undesirable when the objective of estimating k is to facilitate productive cluster analysis. What is really needed is a "roadmap" containing the necessary information for the user to perform the knowledge discovery. In our case, this "roadmap" is the result of analyzing the spectra of the *text* collection. More importantly, this result is easy to obtain, i.e., no parameters needed[1] nor computationally intensive. Furthermore, the spectra contains other information besides k – it also provides a means to estimate the cluster quality of the data set in question.

Spectral analysis has a long history of wide applications in the scientific domain. Usually, the data under study is represented in a matrix, where the eigenvectors or eigenvalues are derived to understand certain properties of the data. In the database community, eigenvectors have applications in areas such as information retrieval (e.g., singular value decomposition [7]), collaborative filtering (e.g., reconstruction of missing data items [2]), and Web searching [25]. Meanwhile, eigenvalues are used in analyzing graphs that are abstraction of real-world objects and their relationships. Application examples include the understanding of communication networks [5] and Internet topologies [39].

The application of spectral analysis in data mining only became popular in the recent years [2]. To the best of our knowledge, we have yet to come across works that use eigenvalues to assist cluster analysis. Most proposals that use spectral techniques for clustering focused on the use of eigenvectors, not eigenvalues. And for those that use eigenvalues, their applications are limited to graph-based data sets (e.g., network topologies), not *text* collections. Our work is therefore novel in two ways: we present an efficient and effective solution to close the gap between the clustering algorithms and the user; and we apply the results of eigenvalue analysis to data sets from various sources.

IX. Conclusions

In this paper, we demonstrate a concrete case of our argument on the need to close the gap between the user and the available machinery. We support our argument by studying a well-known issue in clustering which is faced by every user during analysis: *What value of k should we select so that analysis converges quickly to the desired outcome?*

We answered this question with spectra analysis within the context of *text* collections. We show, both argumentatively and empirically, that if we are able to provide a good estimate to the value of k (or values of k if p-norm is used over 1-norm), then we will have a good starting point for analysis. Once this "ground" is known, data mining can proceed by changing the

[1]From the user's perspective, δ is unknown.

value of k incrementally from the starting point. This is often better than the trial and error approach. In addition, we also show that our proposal can be used to estimate the quality of clustering. As part of the analytical processing, this is equality important to the success of knowledge discovery. Our proposal contributes in part to this insight.

In general, the results shown here demonstrates the feasibility to study techniques that bridge the user with the available machinery – in this case, the clustering algorithm. We strongly believe that this endeavor will play a pivotal role to the advancement of knowledge discovery. In particular, as data takes a paradigm shift into continuous and unbounded form, the user will no longer be able to devote time in tuning parameters. Rather, their time should be spent on interacting with the available machinery such as what we have demonstrated in this paper.

REFERENCES

[1] M. H. Arshad and P. K. Chan. Identifying Outliers via Clustering for Anomaly Detection. Technical Report TR CS-2003-19, Florida Institute of Technology, 2003.

[2] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral Analysis of Data. In *Proc. STOC*, pages 619–626, Crete, Greece, July 2001.

[3] R. Baxter and J. Oliver. The Kindest Cut: Minimum Message Length Segmentation. In *Proc. Int. Workshop on Algorithmic Learning Theory*, pages 83–90, 1996.

[4] Z. Chen, A. Fu, and J. Tang. On Complementarity of Cluster and Outlier Detection Schemes. In *Proc. 5th DaWaK*, Prague, Czech Republic, September 2003.

[5] F. R. K. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[6] D. L. Davies and D. W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1(2):224–227, 1979.

[7] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391–407, 1990.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via The EM Algorithm. *J. of Royal Statistical Society*, 39:1–38, 1977.

[9] J. Dunn. Well Separated Clusters and Optimal Fuzzy Partitions. *J. of Cybernetics*, 4:95–104, 1974.

[10] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster Analysis and Display of Genome-wide Expression Patterns. *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868, December 1998.

[11] D. T. Ross et. al. Systematic Variation in Gene Expression Patterns in Human Cancer Cell Lines. *Nature*, 24(3):227–235, March 2000.

[12] T. R. Golub et. al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286:(5439):531–537, October 1999.

[13] V. R. Iyer et. al. The Transcriptional Program in the Response of Human Fibroblast to Serum. *Science*, 283:83–87, 1999.

[14] F. Evans, M. Alder, and C. deSilva. Determining the Number of Clusters in a Mixture by Iterative Model Space Refinement with Application to Free-swimming Fish Detection. In *Proc. Digital Imaging Computing: Techniques and Applications*, Sydney, Australia, December 2003.

[15] C. Fraley and A. E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, 1998.

[16] G. Golub and C. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, November 1996.

[17] A. Gordon. *Classification*. Chapman and Hall/CRC, 2nd edition, 1999.

[18] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *J. of Intelligent Information Systems*, 17(2-3):107–145, December 2001.

[19] M. Hansen and B. Yu. Model Selection and the Principle of Minimum Description Length. *J. of the American Statistical Association*, 96(454):746–774, 2001.

[20] E. Hartuv, A. O. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir. An Algorithm for Clustering cDNA Fingerprints. *Genomics*, 66(3):249–256, June 2000.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(13):264–323, 1999.

[22] S. Jonnalagadda and R. Srinivasan. An Information Theory Approach for Validating Clusters in Microarray Data. In *Proc. Intelligent Systems for Molecular Biology (ISMB)*, Glasgow, Scottish, August 2004.

[23] R. Kannan and A. Vetta. On Clusterings: Good, Bad and Spectral. In *Proc. FOCS*, pages 367–377, Redondo Beach, 2000.

[24] G. Karypis. CLUTO: A Clustering Toolkit. Technical Report #02-017, University of Minnesota, 2002.

[25] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *J. of the ACM*, 46(5):604–632, 1999.

[26] E. M. Knorr and R. T. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proc. 24th VLDB*, pages 392–403, New York, USA, August 1998.

[27] T. Landauer, P. Foltz, and D. Laham. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.

[28] LANSO. Dept. of Computer Science and the Industrial Liason Office, Univ. of Calif., Berkeley.

[29] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, PA, 1998. The software and this manual are available at URL `http://www.caam.rice.edu/software/ARPACK`.

[30] V. Roth, T. Lange, M. Braun, and J. Buhmann. A Resampling Approach to Cluster Validation. In *Proc. COMPSTAT*, Berlin, Germany, August 2002.

[31] S. Salvador and P. Chan. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. Technical report 2003-18, Florida Institute of Technology, 2003.

[32] R. Sharan and R. Shamir. CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 307–316, California, August 2000.

[33] M. P. Sinka and D. W. Corne. *Soft Computing Systems: Design, Management and Applications*, chapter A Large Benchmark Dataset for Web Document Clustering, pages 881–890. IOS Press, 2002.

[34] P. Smyth. Clustering Using Monte Carlo Cross-Validation. In *Proc. SIGKDD*, pages 126–133, Oregon, USA, 1996.

[35] A. Strehl, J. Ghosh, and R. Mooney. Impact of Similarity Measures on Web-page Clustering. In *Proc. AAAI Workshop on AI for Web Search*, pages 58–64, July 2000.

[36] C. Sugar and G. James. Finding the Number of Clusters in a Data Set: An Information Theoretic Approach. *J. of the American Statistical Association*, 98, 2003.

[37] R. Tibshirani, G. Walther, D. Botstein, and P. Brown. Cluster Validation by Prediction Strength. Technical report, Stanford University, 2001.

[38] R. Tibshirani, G. Walther, and T. Hastie. Estimating the Number of Clusters in a Dataset via the Gap Statistic. Technical Report 208, Dept. of Statistics, Stanford University, 2000.

[39] D. Vukadinovic, P. Huan, and T. Erlebach. A Spectral Analysis of the Internet Topology. Technical Report 118, ETH TIK-NR, 2001.

[40] K. Wu and H. Simon. A Parallel Lanczos Method for Symmetric Generalized Eigenvalue Problems. Technical Report 41284, LBNL, 1997.

[41] Y. Yang, X. Guan, and J. You. CLOPE: A fast and effective clustering algorithm for transactional data. In *Proc. ACM SIGKDD*, pages 682–687, Edmonton, Canada, July 2002.