A Spectroscopy* of Texts for Effective Clustering

Wenyuan Li[1], Wee-Keong Ng[1], Kok-Leong Ong[2], and Ee-Peng Lim[1]

[1] Nanyang Technological University, Centre for Advanced Information Systems
Nanyang Avenue, N4-B3C-14, Singapore 639798
liwy@pmail.ntu.edu.sg, {awkng,aseplim}@ntu.edu.sg
[2] School of Information Technology, Deakin University
Waurn Ponds, Victoria 3217, Australia
leong@deakin.edu.au

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 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. The fundamental question this paper addresses is: "How can we effectively estimate the natural number of clusters in a given *text* collection?". We propose to use spectral analysis, which analyzes the eigenvalues (not eigenvectors) of the collection, as the solution to the above. We first present the relationship between a *text* collection and its underlying spectra. We then show how the answer to this question enhances the clustering process. Finally, we conclude with empirical results and related work.

1 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 of data in recent years, an important and crucial factor to the success of knowledge discovery is to close the gap between the algorithms and the user.

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 to return. In the presence of different data characteristics and analysis contexts, it is often difficult for the user to determine the correct number of clusters in the data set [1–3]. 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

* **spectroscopy** *n.* the study of spectra or spectral analysis.

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 301–312, 2004.

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 for the user.

To strengthen the case further, certain algorithms require a good estimate of the input parameters. For example, the `EM` [4] algorithm is known to perform well in image segmentation [5] when the number of clusters and the initialization parameters are close to their true values. Yet, one reason that limits its application is the poor estimate on the number of clusters. Likewise, a poor parameter setting in `CLOPE` [6] 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.

In this paper, we provide a concrete instance of the above problem by studying the issue in the context of *text* collections, i.e., Web documents, 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, i.e., the number of clusters, becomes a challenging task. In this case, a good starting point is to initialize k to the natural number of clusters.

This gives rise to the fundamental question that this paper addresses: "How can we effectively estimate the natural number of clusters for a given *text* collection?". Our solution is to perform a spectral analysis on the similarity space of the *text* collection by analyzing the eigenvalues (not eigenvectors) that encode the answer to the above question. Using this observation, we next provide concrete examples of how the clustering process is enhanced in a user-centered fashion. Specifically, we argue that spectral analysis addresses two key issues in clustering: it provides a means to quickly assess the cluster quality; and it bootstraps the analysis by suggesting a value for k.

The outline of this paper is as follows. In the next section, we begin with some preliminaries of spectral analysis and its basic properties. Section 3 presents our contribution on the use of normalized eigenvalues to answer the question we posed in this paper. Section 4 discusses a concrete example of applying the observation to enhance the clustering process. Section 5 presents the empirical results as evidence to the viability of our proposal. Section 6 discusses the related work, and Section 7 concludes this paper.

2 Preliminaries

Most algorithms perform clustering by embedding the data in some similarity space [7], which is determined by some widely-used similarity measures, e.g., cosine similarity [8]. 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 $\widehat{v_i v_j}$ 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.

Once we obtained $\mathcal{G}(\mathbf{S})$, we can analyze its spectra as we will illustrate in the next section. However, for ease of discussion, we establish the following basic facts of spectral graph theory below. 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 1. *Let $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n$ be the eigenvalues of $\mathcal{G}(\mathbf{S})$ such that $-1 \leqslant \lambda_i \leqslant 1$, $i = 1, 2, \cdots, n$. Then, the following holds: (i) $\sum \lambda_i = 0$, and $\lambda_1 = 1$; (ii) if $\mathcal{G}(\mathbf{S})$ is connected, then $\lambda_2 < 1$; (iii) the spectra of $\mathcal{G}(\mathbf{S})$ is the union of the spectra of its disjoint subgraphs $\mathcal{G}(\mathbf{S}_i)$.*

Proof. As shown in [9, 10].

In reality, the different similarity matrices are not normalized making it difficult to analyze them directly. In other words, the eigenvalues do not usually fall within $-1 \leqslant \lambda_i \leqslant 1$. Therefore, we need to perform an additional step to get Theorem 1: we transform $\mathbf{S}$ to a weighted Laplacian $\mathbf{L} = (\ell_{ij})$, where $\ell_{ij} \in [0; 2)$ is a normalized eigenvalue obtainable by the following:

$$\ell_{ij} = \begin{cases} 1 - \frac{s_{ij}}{d_i}, & i = j \\ -\frac{s_{ij}}{\sqrt{d_i, d_j}}, & s_{ij} \neq 0 \\ 0, & otherwise \end{cases} \tag{1}$$

where $d_i = \sum_j s_{ij}$ is the degree of vertex v_i in $\mathcal{G}(\mathbf{S})$. We then derive a variant of $\mathbf{L}$ defined as follows:

$$\mathsf{L} = \mathbf{D}^{-1/2}(\mathbf{S} - \mathbf{I})\mathbf{D}^{-1/2} \tag{2}$$

where $\mathbf{D}$ is the diagonal matrix, $\text{diag}(d_i)$. From Equations 1 and 2, we can deduce $\text{eig}(\mathsf{L}) = \{1 - \lambda \,|\, \lambda \in \text{eig}(\mathbf{L})\}$, where $\text{eig}(\cdot)$ is the set of eigenvalues of $\mathbf{S}$. Notably, the eigenvalues in L maintains the same conclusions and properties of those found in $\mathbf{L}$. Thus, we now have a set of eigenvalues that can be easily analyzed. Above all, this approach does not require any clustering algorithm to find k. This is very attractive in terms of runtime and simplicity. Henceforth, the answer to our question is now mapped to a matter of knowing how to analyze $\text{eig}(\mathsf{L})$. We will describe this in the next section.

3 Clustering and Spectral Properties

For ease of exposition, we first discuss the spectra properties of a conceptually disjoint data set, whose chosen similarity measure achieves a perfect clustering. From this simple case, we extend our observations to real-world data sets, and show how the value of k can be obtained.

3.1 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 following structure:

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

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 obtain its eigenvalues in decreasing order [9], i.e.,

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

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* n *clusters. At the same time, the spectra of* $\mathcal{G}(\mathbf{S})$ *exhibit the following properties:*

$$\begin{array}{ll} \lambda_i \to 1 & (i = 1, \cdots, k \text{ and } 0 < \lambda_i \leqslant 1) \\ |\lambda_i| \to 0 & (i = k+1, \cdots, n) \end{array} \tag{5}$$

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

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

where by the well-known property of the Frobenius norm, and the p matrix norm (where $p = 2$ [10]), we have:

$$\|\mathbf{E}\|_2 \leqslant \|\mathbf{E}\|_F \tag{7}$$

and

$$|\lambda_i(\mathbf{A} + \mathbf{E}) - \lambda_i(\mathbf{A})| \leqslant \|\mathbf{E}\|_2, \quad (i = 1, \cdots, n) \tag{8}$$

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

$$\lambda_i(\mathbf{S}) \to \lambda_i(\mathbf{A}), \quad (i = 1, \cdots, n) \tag{9}$$

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

Table 1. A small text collection taken and modified from [11]. 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 **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*

Simply put, when the spectra distribution satisfies Equation (5), then **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 1. 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 1) behaves as per Equation (5).

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

3.2 Spectra Distribution in Large Data Sets

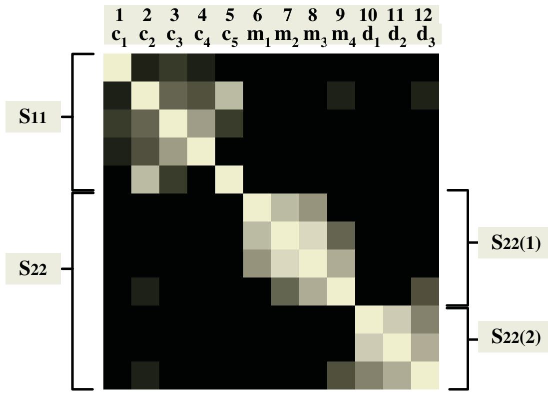
Point (iii) of Theorem 1 offers a strong conclusion between $\mathcal{G}(\mathbf{S})$ and its subgraphs. However, real-world data sets often exhibit a different characteristic. If we examine their corresponding $\mathcal{G}(\mathbf{S})$, we will see that the connections between $\mathcal{G}(\mathbf{S})$ 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 leads to the conclusion about cluster quality and k. But first, we need introduce the Cheeger constant. Let $SV \subset V$ of $\mathcal{G}(\mathbf{S})$. We define the volume of SV as:

$$\mathrm{vol}(SV) = \sum_{v \in SV} d_v \tag{10}$$

	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. 1. The spectra distribution of the collection in Table 1: (a) Spectrum (> 0) of $\mathcal{G}(\mathbf{S})$ and its subgraphs; (b) a graphical representation of **S**. Note that all grey images in this paper are not "plots" of the spectra. Rather, they are a graphical way of summarizing the results of clustering for comparison/discussion purposes.

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:

$$|E(\delta SV)| = \sum_{v_i \in SV, v_j \notin SV} \text{weight}(v_i, v_j) \tag{11}$$

and by Equations (10) and (11), we derive the Cheeger constant:

$$h(\mathcal{G}) = \min_{SV \subset V} \frac{|E(\delta SV)|}{\min(\text{vol}(SV), \text{vol}(\overline{SV}))} \tag{12}$$

which measures the optimality of the bipartition in a graph. The magnitude $|E(\delta SV)|$ measures the connectivity between SV and $\overline{SV}$ while 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 2. *Given the 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 [9], 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 2 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 2 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 3. *Given the spectra of* $\mathcal{G}(\mathbf{S})$ *as* $1 = \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, $\exists k \geqslant 2$ *such that* $\alpha_i \to 1$ *and* $\alpha_i - \alpha_{i+1} > \delta$ $(0 < \delta < 1)$ *for the sequence* $\alpha_i = \frac{\lambda_i}{\lambda_2}$, $(i \geqslant 2)$, *where* δ *is a predefined threshold to measure the first large gap between* α_i*; and* k *is the natural number of clusters in the data set.*

Proof. Since Theorem 2 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 1, we can conclude that the number of eigenvalues in $\mathcal{G}(\mathbf{S})$ (that approach 1 and have large eigengaps) give the value of k, i.e., the number of clusters.

To cite an example for the above, we revisit Table 1 and Figure 1. 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 2, 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 1(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.

4 A Motivating Example

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

Table 2. 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 [12]	10	6607

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 setting k_{max} (see Section 5.2) – 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.

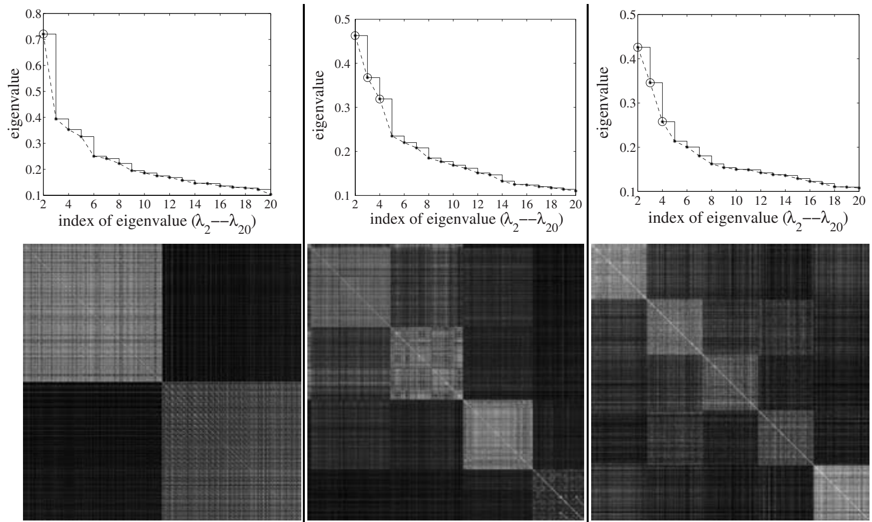
In contrast, our proposal allows the user to obtain an accurate value of k without setting k_{max}. Performance wise, this process is almost instantaneous in comparison to other methods that require a clustering algorithm. We believe this is important if the user's role is to *analyze* the data instead of *waiting* for the algorithms. Once an initial value of k is known, the user can commence clustering. Unfortunately, this isn't the end of cluster analysis.

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 2, the user is given feedback about the cluster quality. At the same time, grey images (e.g., Figure 1(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 feedback to the user via Theorem 2. 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.

5 Empirical Results

The objective of our experiments is to provide the empirical evidence on the viability of our proposal. Due to space limitation, we only report a summary of our results here. The full details can be obtained from [13].



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

Fig. 2. The spectrum graphs (since λ_1 is always 1, our analysis begins from λ_2) and the graphical representation of their clustering for all 3 collections: the first two data sets are conceptually disjoint, and the last has overlapping concepts.

Figure 2(a) shows the `web2` collection with 2 class labels, and their topic being completely disjoint: *finance* and *sport*. In this case, we observe that λ_2 has a higher value than the others. Therefore, we conclude $k = 2$ and meanwhile, the high value of λ_2 (by Theorem 2) indicates that this is a good clustering. In the second case, `classic4` has 4 conceptually disjoint topics as observed in Figure 2(b). From its spectra graph, we see that λ_2, λ_3 and λ_4 show higher values and wider gaps than other eigenvalues. Again by Theorem 3, our method obtains the correct number of clusters, i.e., $k = 4$.

The third test is the most challenging. There are 5 topics: *atheism*, *comp.sys*, *comp.windows*, *misc.forsale* and *rec.sport*. They are not conceptually disjoint because *comp.sys* and *comp.windows* belong to the broader topic of *comp* in the newsgroup. This is also graphically echoed in Figure 2(c). 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 theorem, $k = 4$. This conclusion is actually reasonable, since *comp* is more different than the other topics. If we observe the grey image in Figure 2(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 is closer to the remaining eigenvalues. Therefore, it is possible to conclude that λ_4 may not contribute to k, and hence $k = 3$ by Theorem 3. Strictly speaking, this is also an acceptable estimation. In other

Table 3. Comparison with 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 (√) 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 (√)
KL	29	17	22	27	21	22	22	21	9
Hart	6	13	4 (√)	6	10	4 (√)	1	2	1

words, the onus is on the user to judge the actual value of k, which is really problem-specific as illustrated in the Section 4.

Next, we compare the efficiency and effectiveness of our method. To date, most existing techniques require choosing an appropriate clustering algorithm, where it is iteratively ran with a predefined cluster number from 2 to k_{max}. The optimum k is then obtained by an internal indexed based on the clustering outcome. In this experiment, we compared our results against 3 widely used statistical methods [14] using 3 well-known clustering algorithms (see Table 3). From the table, our proposal remarkably outperforms all 3 methods in terms of accuracy. More exciting, our proposal is independent of any clustering algorithm, is well-suited to high dimensional data sets, has low complexity, and can be easily implemented with existing packages as shown in the following.

On the performance of our method, the complexity of transforming $\mathbf{S}$ to L is $O(h)$, where h is the number of non-zero entries in $\mathbf{S}$. Therefore, the eigenvalues can be efficiently obtained by the `Lanczos` method [10]. Since the complexity of each iteration in `Lanczos` is $O(h+n)$, the complexity of our 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 [15], PLANSO [16]) for such computation.

6 Related Works

Underlined by the fact that there is no clear definition of what is a good clustering [2, 17], the problem of estimating the number of clusters in a 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 [18], penalized likelihood estimation [19, 20], resampling [21], and finding the 'knee' of an error curve [2, 22]. However, these techniques either make a strong parametric assumption, or are computationally expensive.

Spectral analysis has a long history of wide applications in the scientific domain. In the database community, eigenvectors have applications in areas such as information retrieval (e.g., singular value decomposition [23]), collaborative filtering (e.g., reconstruction of missing data items [24]), and Web searching [25]. Meanwhile, application areas of eigenvalues include the understanding of communication networks [9] and Internet topologies [26]. 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.

7 Conclusions

In this paper, we demonstrate a concrete case of our argument on the need to close the gap between data mining algorithms and the user. We exemplified our argument by studying a well-known problem in clustering that every user faces when starting the analysis: "What value of k should we select so that the analysis converges quickly to the desired outcome?".

We answered this question, in the context of *text* collections, with spectral analysis. We show (both argumentatively and empirically) that if we are able to provide a good guess to the value of k, then we have a good starting point for analysis. Once the "ground" is known, data mining can proceed by changing the 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. This process, as part of cluster analysis, is equally important to the success of knowledge discovery. Our proposal contributes in part to this insight.

In the general context, the results shown here also demonstrate the feasibility to study techniques that bridge the algorithms and the user. We 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 algorithms, such as what we have demonstrated in this paper.

References

1. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. Technical report 2003-18, Florida Institute of Technology (2003)
2. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the gap statistic. Technical Report 208, Dept. of Statistics, Stanford University (2000)
3. Sugar, C., James, G.: Finding the number of clusters in a data set : An information theoretic approach. Journal of the American Statistical Association **98** (2003)
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of Royal Statistical Society **39** (1977) 1–38
5. Evans, F., Alder, M., deSilva, C.: Determining the number of clusters in a mixture by iterative model space refinement with application to free-swimming fish detection. In: Proc. of Digital Imaging Computing: Techniques and Applications, Sydney, Australia (2003)

6. Yang, Y., Guan, X., You, J.: CLOPE: A fast and effective clustering algorithm for transactional data. In: Proc. of KDD, Edmonton, Canada (2002) 682–687
7. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Proc. of AAAI Workshop on AI for Web Search. (2000) 58–64
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys **31** (1999) 264–323
9. Chung, F.R.K.: Spectral Graph Theory. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society (1997)
10. Golub, G., Loan, C.V.: Matrix Computations (Johns Hopkins Series in the Mathematical Sciences). 3rd edn. The Johns Hopkins University Press (1996)
11. Landauer, T., Foltz, P., Laham, D.: Introduction to latent semantic analysis. Discourse Processes **25** (1998) 259–284
12. Sinka, M.P., Corne, D.W.: A Large Benchmark Dataset for Web Document Clustering. In: Soft Computing Systems: Design, Management and Applications. IOS Press (2002) 881–890
13. Li, W., Ng, W.K., Ong, K.L., Lim, E.P.: A spectroscopy of texts for effective clustering. Technical Report TRC04/03 (`http://www.deakin.edu.au/~leong/papers/tr2`), Deakin University (2004)
14. Gordon, A.: Classification. 2nd edn. Chapman and Hall/CRC (1999)
15. LANSO: (Dept. of Computer Science and the Industrial Liason Office, Univ. of Calif., Berkeley)
16. Wu, K., Simon, H.: A parallel lanczos method for symmetric generalized eigenvalue problems. Technical Report 41284, LBNL (1997)
17. Kannan, R., Vetta, A.: On clusterings: good, bad and spectral. In: Proc. of FOCS, Redondo Beach (2000) 367–377
18. Smyth, P.: Clustering using monte carlo cross-validation. In: Proc. of KDD, Portland, Oregon, USA (1996) 126–133
19. Baxter, R., Oliver, J.: The kindest cut: minimum message length segmentation. In: Proc. Int. Workshop on Algorithmic Learning Theory. (1996) 83–90
20. Hansen, M., Yu, B.: Model selection and the principle of minimum description length. Journal of the American Statistical Association **96** (2001) 746–774
21. Roth, V., Lange, T., Braun, M., Buhmann, J.: A resampling approach to cluster validation. In: Proc. of COMPSTAT, Berlin, Germany (2002)
22. Tibshirani, R., Walther, G., Botstein, D., Brown, P.: Cluster validation by prediction strength. Technical report, Stanford University (2001)
23. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. JASIS **41** (1990) 391–407
24. Azar, Y., Fiat, A., Karlin, A., McSherry, F., Saia, J.: Spectral analysis of data. In: ACM Symposium on Theory of Computing, Greece (2001) 619–626
25. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46** (1999) 604–632
26. Vukadinovic, D., Huan, P., Erlebach, T.: A spectral analysis of the internet topology. Technical Report 118, ETH TIK-NR (2001)