Eigenvalue-based Incremental Spectral Clustering

Mieczysław A. Kłopotek
Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland

Bartłomiej Starosta
Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland

Sławomir T. Wierzchoń
Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland

January 9, 2025

Abstract Our previous experiments demonstrated that subsets of collections of (short) documents (with several hundred entries) share a common, normalized in some way, eigenvalue spectrum of combinatorial Laplacian. Based on this insight, we propose a method of incremental spectral clustering. The method consists of the following steps: (1) split the data into manageable subsets, (2) cluster each of the subsets, (3) merge clusters from different subsets based on the eigenvalue spectrum similarity to form clusters of the entire set. This method can be especially useful for clustering methods of complexity strongly increasing with the size of the data sample, like in case of typical spectral clustering. Experiments were performed showing that in fact the clustering and merging of subsets yield clusters close to clustering of the entire dataset. Our approach differs from other research streams in that we rely on the entire set (spectrum) of eigenvalues, whereas the other researchers concentrate on few eigenvectors related to lowest eigenvalues. Such eigenvectors are considered in the literature as of low reliability.

1 Introduction

One of intensively developing clustering techniques is the Graph Spectral Analysis, encompassing Graph Spectral Clustering (GSC). It works best for objects whose mutual relationships are described by a graph that connects them based on a similarity measure [31, 26, 33].

One important application is clustering of text documents, where the similarity of documents can be expressed in some ways, e.g. by the count of common words or in terms of more sophisticated descriptions, see e.g. [12] or [28]. In our experiments, we use the cosine similarity between document vectors in the term vector space.

The original GSC suffers from the lack of a method for assignment of new data items to the existing clusters. Hence, clustering from scratch or

training of some external classification model may be required. Clustering from scratch may be hard for large data collections. Classification by the external model may cause cluster definition drift. Due to these issues, several approaches were proposed to solve them, including [23, 4, 2, 27]. This paper can be seen as a contribution to this type of research. The mentioned approaches concentrate on transforming eigenvectors, while our method relies on eigenvalues only.

The algorithm proposed in this paper makes it possible to perform the clustering in batches. The algorithm has the following structure (details are given in section 3):

- For each batch of documents, perform the traditional spectral clustering into the predefined number of clusters.
- Compute the vector of combinatorial or normalized Laplacian eigenvalues of each cluster of each batch.
- Then, based on some dissimilarity criteria between the cluster spectra of different batches, make a decision to combine the corresponding clusters of different batches.
- The matching of clusters is based on minimizing the difference between these vectors.

We investigated the following (dis)similarity criteria:

- normalize the spectra by dividing them by the largest eigenvalue, then the dissimilarity is equal to an (approximate) integral between the class spectrum and the new data set spectrum (Combinatorial Laplacian Relative Lambda Method, `CLRL`)); see Fig 1,
- normalize the spectra by dividing by the dataset size (class or new data set), then the dissimilarity is equal to an (approximate) integral between the class spectrum and the new data set spectrum (Combinatorial Laplacian Sample Size Adjusted Lambda Method, `CLSSAL`); see Fig 2,
- normalize the spectra by dividing by the dataset size (class or new data set), then the dissimilarity is equal to the absolute difference between largest eigenvalues (Combinatorial Laplacian Sample Size Adjusted Maximum Lambda Method, `CLMXL`); see Fig 2,
- compute not the combinatorial Laplacian but rather the Normalized Laplacian (which has always by definition the largest eigenvalue not greater than 2[1], then the dissimilarity is equal to an (approximate) integral between the class spectrum and the new data set spectrum (Normalized Laplacian Method, `NLL`); see Fig 3.

The dissimilarity measures mentioned above differ due to specific properties of GSC. `NLL` is based on normalized Laplacian (see eq.(2)) while the other three measures refer to combinatorial Laplacian (see eq.(1)). This has an effect on the shape of the respective spectrograms.

Eigenvalues of normalized Laplacian are upper-bounded by the value of 2, whatever the sample size is. Thus, if one has samples of different sizes from the same population, the value range is bounded. One needs only to adjust the indexes of eigenvalues to match the spectrograms of data from the same population.

But the eigenvalues of combinatorial Laplacian can grow without any limit if the sample size increases. The `CLRL`, `CLSSAL` and `CLMXL` approaches handle the issue of matching spectrograms of data from the same population in different ways. It is necessary in all these cases to normalize the indexes of eigenvalues (into the range 0-1). The `CLRL` approach

[1] the value 2 is attained for bipartite graphs

normalizes the eigenvalues by dividing by the largest eigenvalue. CLSSAL divides them by the sample size. The effects of both on the spectrogram would be the same for samples from the same population, but the shapes of different population spectrograms will differ in different ways (e.g. in CLRL the spectrograms will meet at both ends, while in CLSSAL they will not). CLRL is more susceptible to noise at the largest eigenvalue than CLSSAL. CLMXL transforms the spectrogram in the same way as CLSSAL, but instead of using an integral to assess the differences between populations it takes the largest eigenvalue after normalization. For justifications of the used properties see [5].

Our approach differs from other research streams in that we rely on the entire set (spectrum) of eigenvalues, whereas the other researchers concentrate on few eigenvectors related to lowest eigenvalues. Such eigenvectors are considered in the literature as of slow convergence [7] and low reliability [22], related also to unreliability of smallest eigenvalues. Furthermore, the approach based on k lowest eigenvalue related eigenvectors may induce noise if the intrinsic number of clusters is lower than k [32]. It may also lead to unreliable results if these eigenvalues are close to one another [32].

As we use the entire eigenspectrum, we avoid such a problem. Papers like [14] allow to conclude that the shape of a spectrogram of eigenvalues could be computed quite reliably.

Our algorithm is proposed in Section 3. The experimental study of the effectiveness of our method is presented in Section 5. The data used in the experiments is described in Section 4. The conclusions are summarized in Section 6. Let us first provide an overview of concepts behind spectral clustering methods in Section 2.

2 Previous Work

One observes growing interest in graph spectral clustering and classification methods. While they have interesting properties with respect to the spatial form of clusters and classes [21], they face the problem of inability to operate incrementally [23, 4, 2, 27]. Let us briefly explain the reasons for this problem.

The traditional way to perform graph spectral clustering is based on the relaxation of ratio cut (RCut) and normalized cut (NCut) graph clustering methods. The k-means algorithm is applied to the rows of the matrix, the columns of which are eigenvectors associated with the k lowest eigenvalues of the corresponding graph Laplacian [21].

Formally, consider a similarity matrix S between pairs of items (e.g. documents). One can imagine a weighted graph G linking the items with weights represented by S. A(n unnormalized) or combinatorial Laplacian L of the matrix S is defined as

$$L(S) = T(S) - S, \tag{1}$$

where $T(S)$ is the diagonal matrix with $t_{jj} = \sum_{k=1}^{n} s_{jk}$ for each $j \in [n]$. A normalized Laplacian $\mathcal{L}$ of the graph represented by S is defined

$$\begin{aligned} \mathcal{L}(S) &= T(S)^{-\frac{1}{2}} L(S) T(S)^{-\frac{1}{2}} \\ &= I - T(S)^{-\frac{1}{2}} S T(S)^{-\frac{1}{2}} \end{aligned} \tag{2}$$

Recall that the RCut criterion means finding the partition matrix $P_{RCut} \in \mathbb{R}^{n \times k}$ that minimizes the formula $H'LH$ over the set of all partition matrices $H \in \mathbb{R}^{n \times k}$. This minimization problem turns out to be NP-hard. This is the reason for relaxing it by assuming that H is a column orthogonal matrix. Then the solution is simple: the columns of P_{RCut} are eigenvectors of L corresponding to the k smallest eigenvalues of L. Similarly, the columns of matrix P_{NCut}, representing NCut criterion, are eigenvectors of $\mathcal{L}$ corresponding to the k smallest eigenvalues of $\mathcal{L}$. For an explanation and further details see e.g. [21] or [33].

Various modifications are applicable, including (1) usage of the top eigenvalue eigenvectors of the matrix $T^{-1/2}ST^{-1/2}$ instead of the lowest ones [13, 30], (2) normalization of the rows of the aforementioned eigenvector sub-matrix to unit length prior to k-means clustering, (3) making use of more than k eigenvectors to cluster into k clusters [25], (4) application of a supervised learning method, instead of clustering, preferentially on a subset of the rows of the aforementioned sub-matrix, followed by employing the learned classifier to the remaining rows.

Also, there exists research on semi-supervised spectral clustering, like the semi-supervised sentiment classification of Li and Hao [18] or semi-supervised spectral detection of population stratification by Liu, Shen, and Pan in [20].

The growing interest in spectral clustering results from the ability to deal with nonlinearly separable datasets But regrettably it suffers from a critical limitation induced by its huge time and space complexity. This handicap severely restricts applicability to large-scale problems.

Because all these methods rely on the computation of eigenvectors and that eigenvectors do not exhibit the property of eigenvectors of bigger matrices being derivable from smaller matrices, there exists a problem with out-of-sample data. Such data enforce computations of the eigenvectors from scratch. This limitation prompted researchers to develop methods that can help to overcome this shortcoming.

One strategy relies on sparsifying the affinity matrix S and solving the eigen-decomposition problem by sparse eigen-solvers [21]. Another strategy is to construct sub-matrices. E.g., the method of Nyström, as applied by [8], randomly selects p representatives from the original dataset and builds an $N \times p$ affinity sub-matrix. [6] improved this method by proposing so-called landmark-based spectral clustering (LSC) method, which performs k-means on the dataset to get p cluster centers as the p representatives. Both approaches seem to suffer from the bottleneck of the number of the sub-matrices to be sampled. [11] proposed two algorithms: ultra-scalable spectral clustering (U-SPEC) and ultra-scalable ensemble clustering (U-SENC). U-SPEC relies on a fast approximation method for K-nearest representatives used in the construction of a sparse affinity sub-matrix. U-SENC integrates multiple U-SPEC into an ensemble clustering framework. These algorithms were further refined in [29] by exploring the approximate explicit feature map (aEFM) transform of low-dimensional data into a low-dimensional subspace in Hilbert space. Still another approach relies on the divide-and-conquer paradigm applied to the landmark-based methodology [17]. The path is followed in [10] where probability density estimation drives the landmark approach. The idea of dealing with processing complexity via ensemble clustering is followed up in [16].

Further research aims at enabling to incorporate out-of-sample data into existent clusters produced by graph spectral clustering. [9] uses the Nyström method solving numerically eigenfunction problems to extrapolate the complete clustering solution with only a small number of samples. A similar idea was presented in [4] by generalizing the eigenfunction approximations beyond the framework of GSC. [2] exploits the idea of approximating binary spectral clustering with weighted kernel PCA, (elaborated by [1]) extending it to multiway clustering, which makes it possible to handle out-of-sample data. This method was refined in [3]. [23] achieves out-of-sample clustering capability via modification of the target function of spectral clustering by adding linear regularisation to the target function. [27] uses modularity similarity measure-based spectral mapping algorithm, that extends the clustering model to out-of-sample data. [19] elaborated a method for out-of-sample data integration based on the methodology of reduction of the excess risk between the empiri-

cal discrete optimal solution and the population-level discrete optimal solution. [15] investigates special structures called "random dot product graph" not on the Laplacian but rather on the adjacency matrix of a graph.

All the aforementioned methods rely on extending the eigenvectors to the out-of-sample data, applying the assumption of piece-wise (approximate) constant property of eigenvectors.

The paper [5] proposes a completely different approach to the problem of this eigenvector discontinuity. Instead of relying on eigenvectors, it turns to the sole usage of eigenvalues. The paper investigates batch type classification problem. Given a collection of documents, labeled with classes, consider a new batch of documents which is known to belong to a single class, but it is unknown to which. It turns out that a comparison of the spectrum of the combinatorial Laplacian of the unlabeled batch with those of labeled batches can identify the appropriate batch with reasonable probability.

3 Our Method

The theoretical background to our assumptions is outlined in the mentioned paper [5]. We do not follow the Nyström paradigm of operating in the embedding space of the L matrix. Instead, we look at the eigenvalue spectra. This approach proved fruitful when performing spectral analysis based *classification*. Our method outperformed classical natural classification, cluster-based classification and spectral eigenvector based classification methods in ten different variants for several real datasets (with short texts) coming from diverse domains. See [5] for details.

Therefore, we decided to investigate its application in the domain of incremental graph spectral *clustering*.

Data: D - a (large) set of documents, to be processed in batches
k - the number of clusters to be obtained
Result: Γ - the clustering of D into k clusters
Split randomly D into (small) subsets $\{D_0, \ldots, D_m\}$;
For each D_i compute its spectral clustering Γ_i into k clusters;
For each cluster $C_{i,j} \in \Gamma_i$ compute the similarity matrix $S_{i,j}$;
$\Gamma := \Gamma_0$ - initial clusters (Γ_0 is the D_0 spectral custering) ;
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** k **do**
    call Algorithm 2 setting: $S := S_{i,j}$; $\mathfrak{S} := \{S_{0,1}, \ldots, S_{0,k}\}$;
    c be the identifier returned by it;
    Update $C_c \in \Gamma$ with $C_c \cup C_{i,j}$;
  end
end

Algorithm 1: The eigenvalue based clustering algorithm

Data: S - similarity matrix of the new cluster of documents
$\mathfrak{S}$ - set of similarity matrices of the clusters of documents to match with
Result: c - the assigned cluster of documents
$L := L(S)$ - Compute Laplacian;
$\mathfrak{L} := L(\mathfrak{S})$ - Compute Laplacians;
$E := spectrum(L)$ - Compute Laplacian eigenvalues;
$\mathfrak{E} := spectrum(\mathfrak{L})$ - Compute Laplacian eigenvalue for each Laplacian from $\mathfrak{L}$;
$F := specfun(E)$ - transform a spectrum into a function;
$\mathfrak{F} := specfun(\mathfrak{E})$ - transform spectra into functions;
$K \leftarrow$ number of clusters in $\mathfrak{S}$;
$c \leftarrow -1$;
$mndist \leftarrow \infty$;
for $j \leftarrow 1$ **to** K **do**
    $distance \leftarrow spectdist(F, \mathfrak{F}_j)$;
    if $distance < mndist$ **then**
        $c \leftarrow j$;
        $mndist \leftarrow distance$;
    else
        do nothing;
    end
end

Algorithm 2: The eigenvalue based class assignment algorithm

Our approach to cluster a large document set D is by breaking it into smaller batches or portions D_i that are easier to handle. Each of these batches can then be clustered using a spectral clustering method. Afterward, the clusters of each document batch can be matched by examining eigenvalue spectra. This process helps in identifying the clusters within the larger dataset D.

The Algorithm 1 presents in a compact way the described method bundle. The functions called in the sub-algorithm 2, that is $L()$, $spectrum()$, $specfun()$, $spectdist()$ are described below.

A drawback of this approach is that each cluster to be discovered must be a homogeneous group. Additionally, each cluster must be distributed proportionally over various batches. By a homogeneous group we understand a population in which each sample has the same (exactly speaking very similar) spectrogram (after normalization by a given method). In fact, our experiments reported in [5] demonstrate that this is the case for various datasets. Homogeneity does not mean that each batch must be of the same size. Rather the share of the group in each batch should be the same. If the shares of groups in different batches differ, then the spectral clustering algorithms would not perform well. More precisely, their underlying algorithm, k-means, performs poorly when the clusters differ too much in size and shape. This is not a flaw of our approach, but rather a general problem of GSC.

Nonetheless, there exist practical applications where homogeneous groups occur proportionally in batches. One example is the task of clustering products handled by big sales companies. The number of consumer products in large chains of hypermarkets may amount to hundreds of thousands and new ones occur in bundles every week. The suppliers do not care about the groups of products the chain has created. Hence, it is the job of chain employees to cluster the products based on their descriptions. While

large computers may handle spectral clustering in hundreds of thousands of dimensions, accessibility of such machines may be not common enough, so that approaches to lower the scale need to be sought.

The Algorithm 2 finds the best matching cluster $C_c \in \Gamma$ by comparing similarity matrix S $(= S_{i,j})$ against the set of matrices $\mathfrak{S}$ $(= \{S_{0,1}, \ldots, S_{0,k}\})$. Updating the cluster $C_c \in \Gamma$ with $C_{i,j}$ might slightly affect its similarity matrix, however, the change should not affect the future indices c returned by Algorithm 1 based on the original $\mathfrak{S}$, due to assumed homogenicity of the clusters.

In the Algorithm 2, being a subroutine of our main Algorithm 1, the following functions are used:

- $spectdist(F_1, F_2)$ function is the area between the two functions F_1, F_2 being its arguments for the function domains [0,1], $\int_0^1 |F_1(x) - F_2(x)|dx$, except for `CLMXL`, where $|F_1(0) - F_2(0)|$ is returned.

- The function $L(S)$ applied to the similarity matrix S is computed according to eq.(1) except for `NLL`, where $\mathfrak{L}(S)$ from eq.(2) is used instead of $L(S)$.

- The function $spectrum(L)$ applied to Laplacian L returns a vector of eigenvalues of L in non-decreasing order.

- The function $specfun(E)$ applied to the spectrum E of a Laplacian returns a function $F(x)$ defined in the domain $x \in [0,1]$ with properties depending on the type of cluster-matching method. Here, the spectrum E is understood as the vector of eigenvalues:

$$E = [\lambda_1, \ldots, \lambda_n] \tag{3}$$

  whereby $\lambda_1 = 0 \leq \cdots \leq \lambda_n$.

  - for `CLRL`:

$$F\left(\frac{n-i}{n-1}\right) = \frac{\lambda_i}{\lambda_n} \tag{4}$$

  - for `CLSSAL` and `CLMXL`:

$$F\left(\frac{n-i}{n-1}\right) = \frac{\lambda_i}{n} \tag{5}$$

  - for `NLL`:

$$F\left(\frac{n-i}{n-1}\right) = \lambda_i \tag{6}$$

  and otherwise for any $x \in \left[\frac{n-(i+1)}{n-1}, \frac{n-i}{n-1}\right]$

$$F(x) = F\left(\frac{n-(i+1)}{n-1}\right) \cdot \left(x - \frac{n-(i+1)}{n-1}\right) + F\left(\frac{n-i}{n-1}\right) \cdot \left(\frac{n-i}{n-1} - x\right).$$

  n is the number of elements in the spectrogram E; the spectrogram is the sequence of eigenvalues ordered decreasingly, with their index i running from 1 to n.

Note that our approach to distance computation between spectra (function $spectdist$) bears some resemblance to Dynamic Time Warping (DTW, [24]) distance. The difference is that we apply a linear transformation to the index axis of the spectrogram, while DTW promotes non-linear transformations.

4 Dataset

For our experiments, we used tweets provided by Twitter (a random sample of about 1% of English tweets) collected for the period from mid September 2019 till the end of May 2022.

We restricted our investigation to tweets having only one hashtag at the end of text with at

least 10 words, whereby we restrict ourselves to the hashtags: #bbnaija, #blacklivesmatter and #puredoctrinesofchrist. This dataset will be referred to as *TWT.EN*. A copy can be found in Supplementary File.

5 Experiments

We want to demonstrate via the experiments that our algorithm correctly matches the clusters stemming from different data portions.

The ideal case for such a demonstration would be: first, the base clustering algorithm splits the data portions along known labels (coming from an external labeling). Then our method matches clusters from different data portions combining the clusters with the same external label. This external label is of course not known to the algorithm.

The only external labels available for our tweets are the hashtags. So the ideal situation would be if an algorithm may split the TWT.EN data in agreement with hashtags.

We assume in our first stage of experiments (subsections 5.1 and 5.2) that in fact such an ideal algorithm exists and has split each portion exactly in agreement with the hashtag labeling. Then we check if our algorithm can correctly match "clusters" stemming from different batches (data portions).

In the second stage (subsections 5.3 and 5.4), we exploit a real spectral clustering algorithm, approximating the split by hashtag.

5.1 Differentiation of hashtags by Laplacian spectrum

In order to check the differentiation of hashtags by Laplacian eigenvalue spectrum, the dataset TWT.EN was divided randomly into three subsets (data portions) of approximately same size. The distribution of the number of documents in data portions for each hashtag is shown in Table 1.

Table 1: Hashtag distribution over data portions

Data portion	#bb naija	#black lives matter	#pure doctrines ofchrist	total
all	1857	2051	1295	5203
Portion 1	634	657	444	1735
Portion 2	616	701	418	1735
Portion 3	607	693	433	1733

For each subset (data portion) and each hashtag, the combinatorial and normalized Laplacian and their spectra of eigenvalues were calculated. The results, in normalized form, suitable for respective methods, are shown in Figs 1, 2 and 3. The figures represent the aforementioned functions $specfun()$ for CLRL (Fig. 1), CLSSAL (Fig. 2) and NLL (Fig. 3) for each of the eigenvalue spectrum of a given hashtag of a given data portion. Lines related to the same hashtag have the same color. To improve visibility, the hashtag names were replaced in the figures by the coding gr1 $\leftarrow$ #bbnaija, gr2 $\leftarrow$ #blacklivesmatter, and gr3 $\leftarrow$ #puredoctrinesofchrist. Each figure has the same structure. For example, in Fig.2, each line represents a spectrogram of one hashtag in one data portion. As there are 3 data portions and 3 hashtags, there are 9 lines in all. For example, green lines represent spectrograms of the hashtag #bbnaija. Let us consider a single line. To obtain it, the similarity matrix S for the tweets related to one hashtag and one data portion was computed and then the respective Laplacian, here according to formula (1), is computed. As a result, the eigenvalue spectrum E, as defined by the formula (3) is computed. Then, for this figure, eq (5) is applied to obtain the spectrogram F.

The spectrogram F is depicted as a line in this figure. the X axis ranges from 0 to 1 (0 is related to the highest eigenvalue, 1 to the lowest), while Y axis provides the value of F. The n eigenvalues of a hashtag dataset are depicted on the X axis in the range 0 to 1 in order to make comparable the spectrograms of hashtag datasets with diverse cardinalities. The formulas (4), (5) and (6) can be viewed as kinds of normalization, this time of the Y axis, related to the spectrum properties related to the dataset sizes, due to the observations explained in [5]. As one can see, the CLSSAL method is characterized by the best separation of the spectrograms of different hashtags. The hashtag #puredoctrinesofchrist seems to be best separated from the other ones.

5.2 Stability of hashtag spectra over various samples

In order to verify the usability of various cluster matching methods (CLRL, CLSSAL, NLL), the stability of hashtag eigenvalue spectra over various samples was investigated.

First, based on the data portion 1, our cluster-matching algorithm was "trained". It means that Laplacians were computed according to formulas (1) and (2) for each hashtag in the data portion 1. Then the spectrogram of Laplacian was computed for each data subset marked with this hashtag. The Algorithm 1 was applied then to an artificial series of 100 data portions created as random subsamples of data portions 2 and 3.

The correctness of data portion assignment to hashtags is shown in Tables 2-5 for the respective methods. Each table is a confusion matrix. Rows are labelled with the true cluster membership (true hashtag) while the columns represent clusters (hashtags) assigned by the Algorithm 1.

As visible from Table 4, the method CLSSAL provides the best results (perfect clustering). CLMXL is

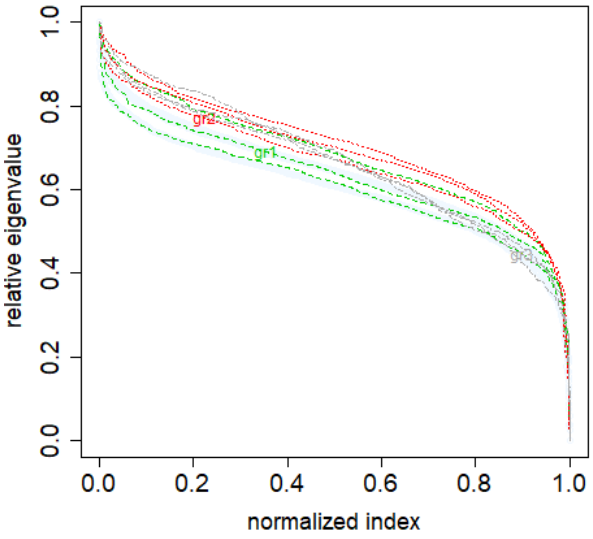

Figure 1: Spectral normalization in the Combinatorial Laplacian Relative Lambda Method (CLRL) on the TWT.EN dataset. Group labels are given in the text.

Table 2: Classification experiment for the dataset TWT.EN for classes using Combinatorial Laplacian Relative Lambda Method (CLRL)

TRUE/PRED	gr1	gr2	gr3
gr1	51	8	41
gr2	24	76	0
gr3	7	9	84

Table 3: Classification experiment for the dataset TWT.EN for classes using the Normalized Laplacian Method (NLL)

TRUE/PRED	gr1	gr2	gr3
gr1	100	0	0
gr2	99	0	1
gr3	0	0	100

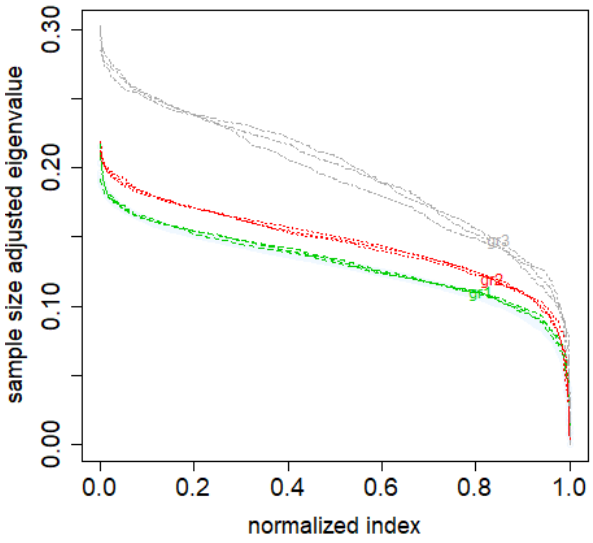

Figure 2: Spectral normalization in the Combinatorial Laplacian Sample Size Adjusted Lambda Method (CLSSAL) and Combinatorial Laplacian Sample Size Adjusted Maximum Lambda Method (CLMXL). The TWT.EN dataset.

Table 4: Classification experiment for the dataset TWT.EN for classes using the Combinatorial Laplacian Set Size Adjusted Lambda Method (CLSSAL)

TRUE/PRED	gr1	gr2	gr3
gr1	100	0	0
gr2	0	100	0
gr3	0	0	100

Table 5: Classification experiment for the dataset TWT.EN for classes using the Combinatorial Laplacian SSA Maximal Lambda Method (CLSSAL)

TRUE/PRED	gr1	gr2	gr3
gr1	33	67	0
gr2	4	96	0
gr3	0	0	100

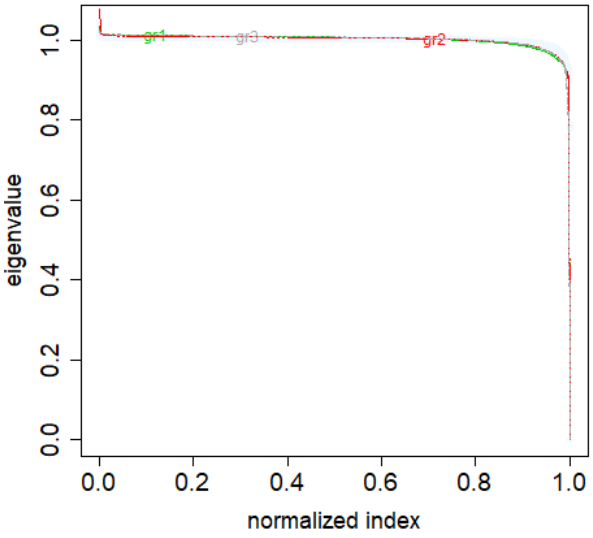

Figure 3: Spectral normalization in the Normalized Laplacian Method (NLL). The TWT.EN dataset.

Table 6: Errors and F1 values for TWT.EN datset

Method	Error %	F1 value
CLRL	29.67	69.82
CLSSAL	0	100
CLMXL	23.67	73.73
NLL	33.33	55.46

Table 7: Result of clustering data portion no 1

TRUE/PRED	pseu-1	pseu-2	pseu-3
#bbnaija	394	239	1
#black lives matter	244	412	1
#pure doctrines ofchrist	71	80	293

Table 8: Result of clustering data portion no 2

TRUE/PRED	pseu-1	pseu-2	pseu-3
#bbnaija	400	216	0
#black lives matter	280	420	1
#pure doctrines ofchrist	45	67	306

Table 9: Result of clustering data portion no 3

TRUE/PRED	pseu-1	pseu-2	pseu-3
#bbnaija	394	213	0
#black lives matter	278	415	0
#pure doctrines ofchrist	48	86	299

the second best (Table 5). On the other hand, `NLL` failed completely (Table 2). Table 6 summarizes the error rates from Tables 2-5 with the F1 measure.

This allows us to conclude that if a clustering method would approximate well the hashtag allocations for these hashtags, then incremental clustering would be possible.

5.3 Differentiation of clusters by Laplacian spectrum

As the next step, each data portion was clustered by Normalized Spectral Clustering method with unit length rows and one additional dimension (that is by a real-world spectral clustering algorithm, described e.g. in [5]).

The result of these clustering processes are visible in Tables 7, 8 and 9 for data portions 1, 2, and 3 respectively. Each table is a confusion matrix. The rows are labelled with tweet hashtags, while the columns are labelled with the cluster "names" to which the tweets were assigned. Cells count the tweets with a given hashtag assigned to a given cluster. We see that the hashtag `#puredoctrinesofchrist` falls nearly completely into a single (`pseu-3`) cluster while the other two hashtags are not separated that well (clusters `pseu-1` and `pseu-2` are quite impure).

We assigned cluster labels as follows: The cluster with the same highest share of a given hashtag gets the same cluster label in each clustering. These cluster labels were of course invisible to the cluster-matching algorithm.

For each subset (data portion) and each cluster label, the combinatorial and normalized Laplacians and their eigenvalue spectra were computed. As previously, the results, in normalized form, suitable for respective methods, are shown in Figs 4, 6 and 5.

Lines related to the same cluster label have the same color.

One can see that again the method `CLSSAL` of data normalization is a clear winner, though first two clusters are not separated well.

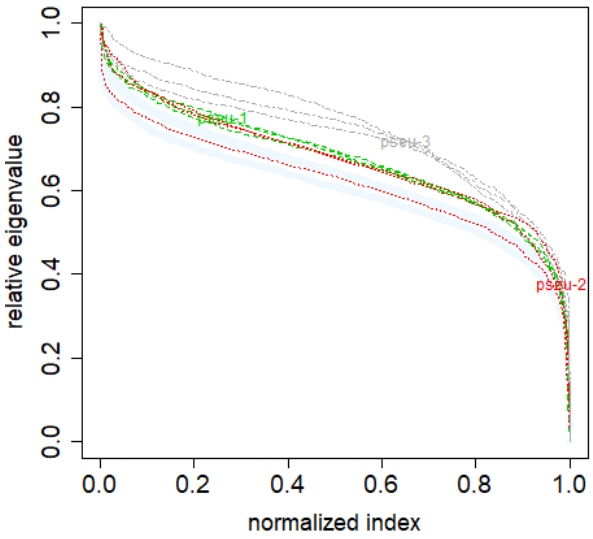

Figure 4: Spectral normalization in the Combinatorial Laplacian Relative Lambda Method (CLRL). The TWT.EN dataset

5.4 Stability of cluster spectra over various samples

To investigate the ability of our method to match clusters from various data portions appropriately, we again trained our cluster-matching algorithm based on the data portion 1, as described above, but this time not for each hashtag, but for each cluster of data portion 1 (pseu-1, pseu-2 and pseu-3). The Algorithm 1 was applied then to an artificial series of 100 data portions created as random subsamples of data portions 2 and 3.

The correctness of data portion assignment to clusters is shown in Tables 10-13 for the respective methods.

As one can see, the method CLSSAL is the best one. Due to overlapping nature of spectrograms of

Table 10: Classification experiment for the dataset TWT.EN for clusters using Combinatorial Laplacian Relative Lambda Method (CLRL)

TRUE/PRED	pseu-1	pseu-2	pseu-3
pseu-1	10	88	2
pseu-2	27	72	1
pseu-3	1	7	92

Table 11: Classification experiment for the dataset TWT.EN for clusters using Normalized Laplacian Method (NLL)

TRUE/PRED	pseu-1	pseu-2	pseu-3
pseu-1	0	0	100
pseu-2	0	0	100
pseu-3	0	0	100

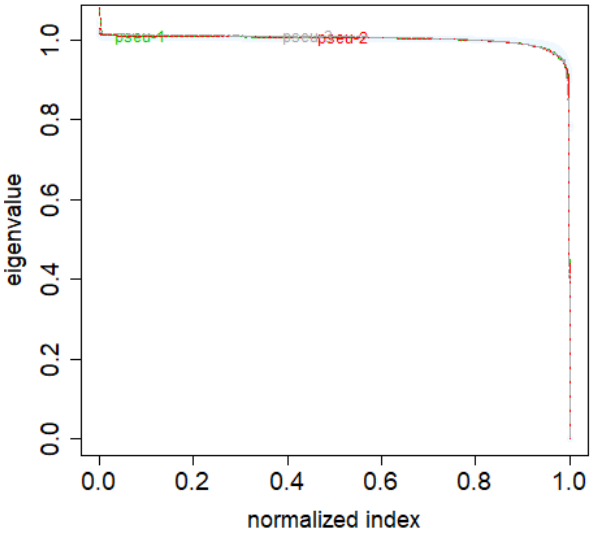

Figure 5: Spectral normalization in the Normalized Laplacian Method. The TWT.EN dataset.

Table 12: Classification experiment for the dataset TWT.EN for clusters using Combinatorial Laplacian Set Size Adjusted Lambda Method (CLSSAL)

TRUE/PRED	pseu-1	pseu-2	pseu-3
pseu-1	72	28	0
pseu-2	34	66	0
pseu-3	0	0	100

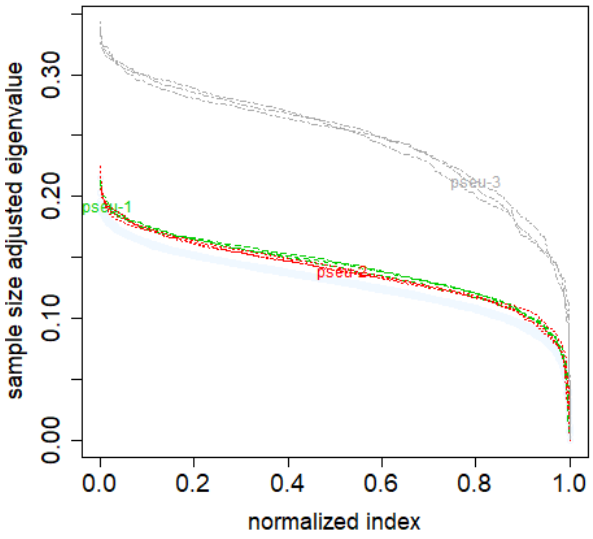

Figure 6: Spectral normalization in the Combinatorial Laplacian Sample Size Adjusted Lambda Method (CLSSAL) and Combinatorial Laplacian Sample Size Adjusted Maximum Lambda Method (CLMXL). The TWT.EN dataset.

Table 13: Classification experiment for the dataset TWT.EN for clusters using Combinatorial Laplacian SSA Maximal Lambda Method (CLMXL)

TRUE/PRED	pseu-1	pseu-2	pseu-3
pseu-1	29	71	0
pseu-2	29	71	0
pseu-3	0	0	100

Method	Error %	F1 value
CLRL	42	54.26
CLSSAL	20.67	79.31
CLMXL	33.33	65.13
NLL	66.67	16.67

Table 14: Errors and F1 values for TWT.EN datset

clusters pseu-1 and pseu-2, they are not as well matched as the cluster pseu-3. See also the error and F1 measures in Table 14. Both are best for CLSSAL, and worst for NLL.

6 Discussion and Conclusions

Graph Spectral Clustering methods, while being attractive for various reasons, suffer, among others, from the inability to integrate out-of-sample data into existent clusters. As recalled in Section 2, various approaches were tried out to overcome this shortcoming. All of them seem to concentrate on handling (extending) the (low) eigenvectors. Our research differs substantially from those approaches in that we take into account solely the eigenvalue spectrum.

The study we conducted and published in this paper demonstrates that Twitter tweets with the same hashtag are ”similar” in ”style” across all subsamples, or in other words, they have a combinatorial Laplacian spectrum. This also applies to clusters that are produced when algorithms for clustering find the hashtags. Hence, rather than clustering the entire set,

we can cluster portions, recover the total cluster, and then use the Laplacian spectrum to match the clusters from the subsets.

The fact that the subsets of collections of (brief) texts have a same normalized eigenvalue spectrum appears to be an intriguing characteristic. A given method's clusters may be used to split the clustering process into smaller data segments and then match the resulting subclusters using the described Combinatorial Laplacian Sample Size Adjusted Lambda Method if the clusters yield spectra with noticeably different characteristics. This is particularly helpful for clustering techniques whose complexity increases significantly as the sample amount of data increases.

Additional investigation would focus on comprehending the relationship between the eigenvalue spectrogram and the literary style of collections of short texts on a specific subject, as well as the explanations for why certain spectrograms are essentially the same for different topical collections.

References

[1] C. Alzate and J.A.K. Suykens. A weighted kernel pca formulation with out-of-sample extensions for spectral clustering methods. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 138–144, 2006.

[2] C. Alzate and J.A.K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 32(2):335 – 347, 2010.

[3] Carlos Alzate and Johan A. K. Suykens. Out-of-sample eigenvectors in kernel spectral clustering. In *The 2011 International Joint Conference on Neural Networks*, pages 2349–2356, 2011.

[4] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for LLE, ISOMAP, MDS, eigenmaps, and spectral clustering. In *NIPS 16*, pages 177–184, 2003.

[5] P. Borkowski, M.A. Kłopotek, B. Starosta, S.T. Wierzchoń, and M. Sydow. Eigenvalue based spectral classification. *PLOS ONE*, 18(4):e0283413, 2023. `https://doi.org/10.1371/journal.pone.0283413`.

[6] Deng Cai and Xinlei Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1669–1680, 2015.

[7] Jiangning Chen. Convergence rate of krasulina estimator. *Statistics & Probability Letters*, 155:108562, 2019.

[8] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[9] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), 2004.

[10] Xia Hong, Junbin Gao, Hong Wei, James Xiao, and Richard Mitchell. Two-step scalable spectral clustering algorithm using landmarks and probability density estimation. *Neurocomputing*, 519:173–186, 2023.

[11] Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh.

Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226, 2020.

[12] R. Janani and S. Vijayarani. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, 134:192–200, 2019.

[13] S.D. Kamvar, D. Klein, and Ch.D. Manning. Spectral learning. In *Proceedings of the 18th International Joint Conf. on Artificial intelligence. IJCAI'03*, page 561–566, 2003.

[14] Dongjin Lee, Takeo Hoshi, Tomohiro Sogabe, Yuto Miyatake, and Shao-Liang Zhang. Solution of the k-th eigenvalue problem in large-scale electronic structure calculations. *Journal of Computational Physics*, 371:618–632, 2018.

[15] Keith D. Levin, Farbod Roosta-Khorasani, Michael W. Mahoney, and Carey E. Priebe. Out-of-sample extension of graph adjacency spectral embedding. In *International Conference on Machine Learning*, 2018.

[16] Hongmin Li, Xiucai Ye, Akira Imakura, and Tetsuya Sakurai. Lsec: Large-scale spectral ensemble clustering, 2021.

[17] Hongmin Li, Xiucai Ye, Akira Imakura, and Tetsuya Sakurai. Divide-and-conquer based large-scale spectral clustering. *Neurocomputing*, 501:664–678, aug 2022.

[18] S. Li and J. Hao. Spectral clustering-based semi-supervised sentiment classification. In S. Zhou, S. Zhang, and G. Karypis, editors, *Advanced Data Mining and Applications. ADMA 2012*, volume LNAI 7713, pages 271–283. Springer-Verlag Berlin Heidelberg, 2012. https://doi.org/10.1007/978-3-642-35527-1_23.

[19] Shaojie Li, Sheng Ouyang, and Yong Liu. Understanding the generalization performance of spectral clustering algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37/7, pages 8614–8621, 2023.

[20] Binghui Liu, Xiaotong Shen, and Wei Pan. Semi-supervised spectral clustering with application to detect population stratification. *Frontiers in Genetics*, page Article 215, Oct 2013. https://doi.org/10.3389/fgene.2013.00215.

[21] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. http://dx.doi.org/10.1007/s11222-007-9033-z.

[22] Bappaditya Mandal, Xudong Jiang, How-Lung Eng, and Alex Kot. Prediction of eigenvalues and regularization of eigenfeatures for human face verification. *Pattern Recognition Letters*, 31(8):717–724, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

[23] F. Nie, Z. Zeng, I.W. Tsang, D. Xu, and Ch. Changshui. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Trans. Neur. Netw.*, 2011.

[24] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition, Chaspter 4*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[25] N. Rebagliati and A. Verri. Spectral clustering with more than K eigenvectors. *Neurocomputing*, 74(9):1391–1401, Apr 2011.

https://doi.org/10.1016/j.neurocom.2010.12.08.

[26] H. Sevi, M. Jonckheere, and A. Kalogeratos. Generalized spectral clustering for directed and undirected graphs, 2022. https://arxiv.org/abs/2203.03221.

[27] D. Shen, Xiuyi Li, and Guannan Yan. Improve the spectral clustering by integrating a new modularity similarity index and out-of-sample extension. *Modern Physics Letters B*, 34(11):2050105, 2020.

[28] Prajol Shrestha, Christine Jacquin, and Béatrice Daille. Clustering short text and its evaluation. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing. CICLing 2012*, volume 7182 of *Lecture Notes in Computer Science*, page 169–180. Springer, Berlin, Heidelberg, 2012. https://doi.org/10.1007/978-3-642-28601-8_15.

[29] Dario Sitnik and Ivica Kopriva. Clustering and classification of low-dimensional data in explicit feature map domain: intraoperative pixel-wise diagnosis of adenocarcinoma of a colon in a liver, 2022.

[30] R. Suganthi and S. Manimekalai. Spectral clustering based classification algorithm for text classification. *International Journal of Engineering Science Invention (IJESI)*, pages 36–41, 2018. http://www.ijesi.org/papers/NCIOT-2018/Volume-3/7.\%2036-41.pdf.

[31] Jingzhi Tu, Gang Mei, and Francesco Piccialli. An improved Nyström spectral graph clustering using k-core decomposition as a sampling strategy for large networks. *Journal of King Saud University - Computer and Information Sciences*, 2022.

[32] Slawomir T. Wierzchon and Mieczyslaw A. Klopotek. Spectral cluster maps versus spectral clustering. In *Computer Information Systems and Industrial Management*, volume 12133 of *LNCS*, pages 472–484. Springer, 2020.

[33] S.T. Wierzchoń and M.A. Kłopotek. *Modern Clustering Algorithms*, volume 34 of *Studies in Big Data*. Springer Verlag, 2018.