Spectral Clustering Based on k-Nearest Neighbor Graph

Conference Paper · September 2012

DOI: 10.1007/978-3-642-33260-9_22

Małgorzata Lucińska
Politechnika Świętokrzyska
9 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Slawomir T. Wierzchon
Polish Academy of Sciences
162 PUBLICATIONS **815** CITATIONS
SEE PROFILE

Spectral clustering based on k-nearest neighbor graph

Małgorzata Lucińska[1] and Sławomir T. Wierzchoń[2,3]

[1] Kielce University of Technology, Kielce, Poland
[2] Institute of Computer Science Polish Academy of Sciences, Warsaw, Poland
[3] University of Gdańsk, Gdańsk, Poland

Abstract. Finding clusters in data is a challenging task when the clusters differ widely in shapes, sizes, and densities. We present a novel spectral algorithm `Speclus` with a similarity measure based on modified mutual nearest neighbor graph. The resulting affinity matrix reflex the true structure of data. Its eigenvectors, that do not change their sign, are used for clustering data. The algorithm requires only one parameter – a number of nearest neighbors, which can be quite easily established. Its performance on both artificial and real data sets is competitive to other solutions.

Keywords: spectral clustering, nearest neighbor graph, signless Laplacian

1 Introduction

Clustering is a common unsupervised learning technique; its aim is to divide objects into groups, such that members of the same group are more similar each to another (according to some similarity measure) than any two members from two different groups. Different applications of clustering in practical problems are reviewed e.g. in [6]. Although many clustering methods have been proposed in the recent decades, see e.g. [5] or [15], there is no universal one that can deal with any clustering problem, since the real world clusters may be of arbitrary complicated shapes, varied densities and unbalanced sizes.

Spectral clustering techniques [13] belong to popular and efficient clustering methods. They allow to find clusters even of very irregular shapes, contrary to other algorithms, like k-means algorithm [7]. Spectral techniques use eigenvalues and eigenvectors of a suitably chosen matrix to partition the data. The matrix is the affinity matrix (or a matrix derived from it) built on the basis of pairwise similarity of objects to be grouped. The structure of the matrix plays a significant role in correct cluster separation. If it is clearly block diagonal, its eigenvectors will relate back to the structural properties of the set of the objects, [9].

One of the key tasks in spectral clustering is the choice of similarity measure. Most spectral algorithms adopt a Gaussian kernel function defined as:

$$S(i,j) = \exp\Big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\Big) \tag{1}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the Euclidean distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel parameter σ influences the structure of an affinity matrix and generally it is difficult to find its optimal value. Some authors propose a global value of σ for the whole data set e.g. [11] and [12] while the others suggest using a local parameter e.g. [16]. However both the solutions fail to reveal the properties of real world data sets [14]. Another open issue of key importance in spectral clustering is that of choosing a proper number of groups. Usually this number is a user defined parameter [11], but sometimes it is estimated – with varying success rate [12] – in a heuristically motivated way.

In this paper we present a spectral clustering algorithm `Speclus` that can simultaneously address both of the above mentioned challenges for a variety of data sets. It adopts the idea derived by Shi *et al.* from their analysis of the relationship between a probability distribution and spectrum of the corresponding distribution-dependent convolution operator, [12]. Their DaSpec (i.e. Data Spectroscopic) algorithm estimates the group number by finding eigenvectors with no sign change and assigns labels to each point based on these eigenvectors. In our algorithm the similarity between pairs of points is deduced from their neighborhoods. The use of similarity based on nearest neighbors approach removes, at least partially, problems with cluster varying densities and the unreliability of distance measure. Resulting adjacency matrix reflects true relationships between data points. Also the σ parameter is replaced by the number of neighbors parameter, which can be chosen more simply since it is an integer and takes a small number of values. Apart from only one parameter another advantage of the presented approach is that it incorporates a variety of recent and established ideas in a complete algorithm which is competitive to current solutions.

In section 2 the notation and related work is presented, the next section explains the main concepts used in the `Speclus` algorithm, which is presented in details in section 4. Then, in section 5, we compare performance of our algorithm with other solutions. Finally, in section 6, the main conclusions are drawn.

2 Notation and related work

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be the set of data points to be clustered. For each pair of points i, j an adjacency $a_{ij} \in \{0, 1\}$ is attached (see Section 3 for details). The value $a_{ij} = 1$ implies the existence of undirected edge $i \sim j$ in the graph G spanned over the set of vertices $\mathbf{X}$. Let $A = [a_{ij}]$ be the adjacency matrix. Let $d_i = \sum_j a_{ij}$ denote the degree of node i and let D be the diagonal matrix with d_i's on its diagonal. A clustering $\mathcal{C} = (C_1, C_2, ..., C_l)$ is a partition of $\mathbf{X}$ into l nonempty and mutually disjoint subsets. In the graph-theoretic language the clustering represents a multiway cut in G [2].

In the `Speclus` algorithm a signless Laplacian $M = D + A$, introduced by Cvetković [1], is used. Cvetković proves that the spectrum (i.e. the set of eigenvalues) of M can better distinguish different graphs than spectra of other commonly used graph matrices. Graphs with the same spectrum of an associated matrix B are called cospectral graphs with respect to B, or B-cospectral graphs. A graph

H cospectral with a graph F, but not isomorphic to F, is called a cospectral mate of H. Let $\mathcal{G}$ be a finite set of graphs, and let $\mathcal{G}'$ be the set of graphs in $\mathcal{G}$ which have a cospectral mate in $\mathcal{G}$ with respect to M. The ratio $|\mathcal{G}'|/|\mathcal{G}|$ is called the spectral uncertainty of (graphs from) $\mathcal{G}$ with respect to B. Cvetković compares spectral uncertainties with respect to the adjacency matrix, the Laplacian ($L = D - A$), and the signless Laplacian of sets of all graphs on n vertices for $n \leq 11$. Spectral uncertainties in case of the signless Laplacian are smaller than for the other matrices. This indicates that the signless Laplacian seems to be very convenient for use in studying graph properties.

As already mentioned the `Speclus` algorithm utilizes the idea proposed by Shi *et al.* in [12]. They study the spectral properties of an adjacency matrix A and its connection to the data generating distribution P. The authors investigate the case when the distribution P is a mixture of several dense components and each mixing component has enough separation from the others. In such a case A and L are (close to) block-diagonal matrices. Eigenvectors of such block-diagonal matrices keep the same structure. For example, the few top (i.e. corresponding to highest eigenvalues) eigenvectors of L can be shown to be constant on each cluster, assuming infinite separation between clusters. This property allows to distinguish the clusters by looking for data points corresponding to the same or similar values of the eigenvectors. Shi *et al.* develop in [12] theoretical results based on a radial similarity function with a sufficiently fast tail decay. They prove that each of the top eigenvectors of A corresponds exactly to one of the separable mixture components. The eigenvectors of each component decay quickly to zero at the tail of its distribution if there is a good separation of components. At a given location $\mathbf{x}_i$ in the high density area of a particular component, which is at the tails of other components, the eigenvectors from all other components should be close to zero.

Also Elon [3] attempts to characterize eigenvectors of the Laplacian on regular graphs. He suggests that the distribution of eigenvectors, except the first one, follows approximately a Gaussian distribution. There are also proofs that in general, top eigenvalues have associated eigenvectors which vary little between adjacent vertices. The two facts confirm the assumption that each cluster is reflected by at least one eigenvector with large components associated with the cluster vertices and almost zero values in the other case.

Another concept incorporated in the `Speclus` algorithm comes from Newman. It concerns a quality function called modularity, which is used for assessing a graph cut [10].

Another concept incorporated in the `Speclus` algorithm is so-called modularity, i.e. a quality function introduced by Newman [10] for assessing a graph cut. According to its inventor a good division of a graph into partitions is not merely one in which there are few edges between groups; it is one in which there are fewer than expected edges between groups. The modularity Q is, up to a multiplicative constant, the number of edges falling within groups minus the expected number in an equivalent graph with edges placed at random, or in functional form

$$Q = \frac{1}{2m} \sum_{ij} \left[a_{ij} - \frac{d_i d_j}{2m} \right] \delta(g_i, g_j) \tag{2}$$

where $\delta(r, s) = 1$ if $r = s$ and 0 otherwise, and m is the number of edges in the graph. Newman suggests that a division on a graph makes sense if $Q > 0.3$.

3 The main concepts

The novel concept of the `Speclus` algorithm is the similarity measure based on nearest neighbors approach. Specifically the k mutual nearest neighbor graph is constructed with points as the vertices and edges as similarities. First for each of the points k symmetric nearest neighbors are found with Euclidean distance as the distance metric. Then for each two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ the connecting edge v_{ij} is created if vertex $\mathbf{x}_i$ belongs to k-nearest neighbors of vertex $\mathbf{x}_j$ and vice versa. Afterwards vertices with a small number of edges (less than half of an average number of edges connected to one point in the graph) are identified. Each of such vertices with low degree is additionally connected to a few nearest neighbors of vertices in its closest proximity. By "closest proximity" we understand approximately the first $k/2$ neighboring vertices and half of its neighbors create additional connections, but only in case their degree is less than $k/2$. The resulting graph is similar to a mutual nearest neighbor graph described in [8]. The difference lies in additional edges, which are created between vertices with low degrees. For each pair of nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ in such constructed graph the value a_{ij} is set to one if and only if there is an edge joining the two vertices. Otherwise a_{ij} equals 0. Also all diagonal elements of the affinity matrix A are zero.

Such an approach guarantees a sparse affinity matrix, capturing the core structure of the data and achieved simply with only one parameter k. It can also handle data containing clusters of differing densities. To illustrate this statement let us consider two neighboring clusters: a dense cluster A and a sparse cluster B, as Figure 1 shows. The point P does not belong to the mutual nearest neighbors of the point A, as the last one has many other neighbors closer to it than P. In such a case lacking neighbors of the point P will be supplemented by the nearest neighbors of points Q and R.

In order to estimate the number of groups and divide data into clusters the `Speclus` algorithm utilizes structure of the top eigenvectors of signless Laplacian. According to works [3] and [12], and our extensive numerical observations, top eigenvectors of sparse matrices, related to points creating disjoint subsets, reflect the structure of the data set. Figure 2 shows an ideal example, when three clusters are completely separated and each of them can be presented in the form of the regular graph of the same degree. Top eigenvectors of signless Laplacian show clearly its structure. Each cluster is represented by an eigenvector, which assumes relatively high values (of one sign) for points belonging to the cluster and zero values for points from other clusters. An additional regularity can also be seen – if a point is close to a cluster center its value in the corresponding

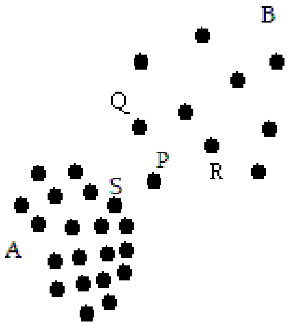

Fig. 1. Choice of mutual nearest neighbors in case of two clusters with different densities

eigenvector is high. The points, which lay at the border of a cluster have relatively small values of the appropriate eigenvector.

In real situations, when subsets are close to each other, overlap or have different densities, the picture of data structure given by the top eigenvectors can be a little confusing. Shi *et al.* notice that smaller or less compact groups may not be identified using just the very top part of the spectrum. More eigenvectors need to be investigated to see these clusters. On the other hand, information in the top few eigenvectors may also be redundant for clustering, as some of these eigenvectors may represent the same group. In the `Speclus` algorithm the problems are solved with the help of modularity function. If two eigenvectors indicate two different divisions of the set, the modularity is calculated in order to choose a better cut in terms of modularity maximization.

4 The `Speclus` algorithm

The steps of the `Speclus` algorithm are as follows:

The `Speclus` algorithm

```
Input: Data X, number of nearest neighbors k
Output: C clustering of X
Algorithm:
1. Compute, in the following order
    k-nearest neighbors for each x
    mutual nearest neighbors for each x
    additional neighbors in case degree of x < half of
    average degree in X
2. Create affinity matrix A and signless Laplacian M=D+A
3. Compute top w eigenvectors of M
4. Find eigenvectors with no sign change (up to standard
    deviation of its values)
```

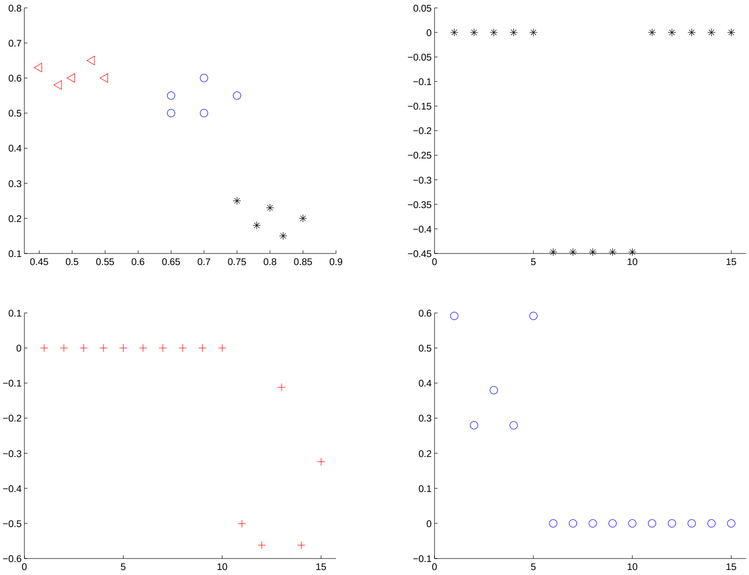

Fig. 2. Perfectly separated clusters (top right) and their eigenvectors (top left and bottom).

```
5. Determine overlapping eigenvectors (related to the same
    cluster)
6. Calculate modularity corresponding to the graph cut for
    each overlapping eigenvector and choose the best one
7. Create a set of eigenvectors A, each representing one cluster
8. Assign each point x  to one eigenvector from the set A,
    having the biggest entry for x
```

The algorithm builds a graph, with points as vertices and similarities between points as edges. It starts by retrieving k nearest neighbors for each point and afterwards generates mutual nearest neighbors. If a degree of a vertex is smaller than half of the average degree of vertices in the graph, edges to nearest neighbors of the vertices from the closest proximity are added, as it is described in section 3. After determining the affinity matrix and signless Laplacian the eigenvectors and eigenvalues of the last one are calculated. The number of eigenvectors computed, $w = 20$ is estimated as twice the maximum expected number of clusters, that guarantees representation of each cluster by at least one eigenvector. Next, eigenvectors with no sign change are extracted. We assume that an eigenvector does not change a sign if all its positive entries are smaller than its standard deviation or absolute values of its all negative entries do not exceed the standard deviation. If the clusters are not perfectly separated or have varying densities one cluster may be represented by a few eigenvectors. Such overlapping eigenvectors are recognized with a help of a point with the biggest entry for each eigenvector (eigenvector maximum). As it is mentioned in the section 3 of this paper, points located in the center of a cluster have big entries in appropriate eigenvectors. A point corresponding to the maximum of one eigenvector should have small entries in the other eigenvectors, unless they represent the same cluster. After establishing the maximum of an eigenvector $\mathbf{v}$ we compare its values in the other eigenvectors. If the appropriate entry in an eigenvector $\mathbf{w}$ is bigger than a small value ϵ, e.g. $\epsilon = 0.001$, it means that the two eigenvectors overlap. Such pairs of eigenvectors create a set B, while the eigenvectors, which do not overlap belong to a set A. First the data set is divided into clusters on the basis of the set A. If a point $\mathbf{x}$ has the biggest entry in the eigenvector $\mathbf{v}$ it receives a label v, etc. Afterwards similar divisions are made with a use of each overlapping eigenvector pair from the set B. Let us assume that eigenvectors $\mathbf{v1}$ and $\mathbf{v2}$ overlap with each other. A point, which has an entry in $\mathbf{v1}$ bigger than ϵ is labeled as a set $C1$, if the entry in $\mathbf{v2}$ is bigger the label corresponds to a set $C2$. For each of the two divisions a modularity function is calculated. The eigenvector, which leads to better division in terms of modularity function is added to the set A. Eigenvectors from this set are used for the final labeling of the data. The number of eigenvectors included in the set A indicates the number of groups. Each eigenvector represents one cluster. Each point is labeled according to the eigenvector with the highest entry for the point.

Computational complexity of the proposed algorithm is relatively small. First of all the affinity matrix is very sparse as we use the concept of mutual neighbors. Second the number of needed eigenvectors is relatively small, if we consider clusters of reasonable size only, i.e. if we require that the minimal cluster size exceeds 1 percent of the size of the whole data set. Moreover, in case of a signless Laplacian we seek for top eigenvectors, which are easier to find than eigenvectors corresponding to smallest eigenvalues. In such situation solving the eigen problem even for large data set is not very time consuming. The other steps of the algorithm take time $O(n)$ each. So the solution is scalable.

5 Experimental results

We have compared the performance of the `Speclus` algorithm (implemented in MATLAB) to three other methods: the Ng *et al.* algorithm [11], the Fischer *et al.* algorithm [4], and the DaSpec algorithm. The first one is a standard spectral algorithm, which uses normalized Laplacian $L = D^{1/2}SD^{1/2}$ and k-means algorithm for final clustering. The second one aims at amplifying the block structure of affinity matrix by context-dependent affinity and conductivity methods. The DaSpec algorithm uses the same properties of eigenvectors as the `Speclus` algorithm and similarly does not need a cluster number to be given in advance. The first two algorithms need a number of clusters as an input parameter.

In the case of the three algorithms the σ parameter should be carefully established. Ng *et al.* and Shi *et al.* have proposed heuristics to calculate the value of the σ parameter. For many data sets neither of the formulas can guarantee correct classification. In order to compare the best achievements of all the algorithms the values of the σ parameter were chosen manually, as described by Fischer *et al.* For each data set they systematically scanned a wide range of σ's and ran the clustering algorithms. We use their results in case of the first two algorithms.

All the algorithms are evaluated on a number of benchmark data sets identical as in [4]. Six of the sets are artificial and three are real-world problems. They cover a wide range of difficulties, which can be met during data segmentation. The first data set 2R3D.2 is obtained by dispersing points around two interlocked rings in 3D. The dispersion is Gaussian with standard deviation equal 0.2. The 2RG data set consists of two rather high density rings and a Gaussian cluster with very low density. The 2S set is created by two S-shaped clusters with varying densities within each group. Sets 4G and 5G have four and five Gaussian clusters each of different density in 3D and 4D respectively. 2Spie is a standard set used for evaluation of spectral clustering algorithms consisting of two spirals with double point density. The last three sets are common benchmark sets with real-world data: the iris, the wine and the breast cancer. The first one consists of three clusters, two of which are hardly separated. The wine is a very sparse set with only 178 points in 13 dimensions.

Table 1. Comparision of Ng et al., Fischer et al. DaSpec, and `Speclus` algorithms. n denotes number of points, l – number of clusters, and D – data dimension.

Data	n	l	D	Ng *et al.*	Fischer *et al.*	DaSpec	`Speclus`
2R3D.2	600	2	3	4	93	195	11
2RG	290	3	2	101	0	180	0
2S	220	2	2	0	0	70	0
4G	200	4	3	18	1	41	2
5G	250	5	4	33	11	53	11
2Spi	386	2	2	0	193	191	0
Iris	150	3	4	14	7	35	14
Wine	178	3	13	3	65	89	9
BC	683	2	9	22	20	239	21

As can be seen from Table 1 the `Speclus` algorithm is the most flexible one and performs well independently on data set structure. Although both the `Speclus` and the DaSpec algorithms use the same concept of eigenvector properties the second one often fails on real-world data or clusters with different densities. For sets presented in Table 1 it usually is not able to detect all the clusters. The dramatic differences in the performance between the two algorithms can be explained as a result of the use of signless Laplacian and special similarity measure in the `Speclus` algorithm. The signless Laplacian spectrum pictures better data structure than spectrum of any other graph matrix. The similarity measure based on the mutual neighbors concept caused the affinity matrix to be more clearly block diagonal. Whether density of points varies meaningly with or within clusters, our method of constructing affinity matrix gives good results. Even in case of very sparse data or high Gaussian noise the labeling of data by `Speclus` is correct. If data are sparse, adding edges between vertices with few connections and neighbors of vertices from their closest proximity avoids lost of graph connectivity and isolation of some points. On the other hand creating connections between graph vertices on the basis of mutual nearest neighbors eliminates influence of any noises in the data.

We also compare the performance of the `Speclus` algorithm with the DaSpec algorithm on the basis of sets suggested by Shi *et al.* in [12], that contain non-Gaussian groups and various levels of noise. We use the set $DS1$, that consists of three well-separable groups in $\mathbb{R}^2$. The first group of data is generated by adding independent Gaussian noise $N((0,0)^T, 0.15^2\mathbb{I})$ to 200 uniform samples from three fourth of a ring with radius 3. The second group includes 100 data points sampled from a bivariate Gaussian $N((3,-3)^T, 0.5^2\mathbb{I})$ and the last group has only five data points sampled from a bivariate Gaussian $N((0,0)^T, 0.3^2\mathbb{I})$. Here $\mathbb{I}$ stands for the unit matrix. Given DS1, three more data sets (DS2, DS3, and DS4) are created by gradually adding independent Gaussian noise (with standard deviations 0.3, 0.6, 0.9 respectively). The results obtained for the four data sets with the `Speclus` algorithm are shown in the left column and with the DaSpec algorithm in the right column of Figure 3. It is clear that the degree of separation decreases from top to bottom. The divisions resulting form our algorithm are more correct than in the case of the other algorithm. However, neither of them is able to separate the five points inside the part of the ring. But even for the highest level of noise the `Speclus` algorithm finds the right number of groups.

At last we show how performance of the `Speclus` algorithm changes if we use ordinary Laplacian $L = D - A$ instead of signless Laplacian $M = D + A$. In Figure 4 there are eigenvectors of Laplacian L and Laplacian M, which are used for partitioning of the set 2RG. The data structure is perfectly illustrated by signless Laplacian eigenvectors, indicating three separate clusters. Ordinary Laplacian eigenvectors indicate only two clusters, whereas the third one does not have any clear representation. This result constitutes an experimental proof, that signless Laplacian is more suitable for partitioning sets with varying densities than Laplacian L.

6 Conclusions and future work

We have presented a new spectral clustering algorithm, which uses signless Laplacian eigenvectors and a novel affinity matrix.

The matrix is created on the basis of a mutual nearest neighbor graph with additional edges connecting points from sparse density areas. Experiments confirm that a good similarity measure is crucial to the performance of spectral clus-

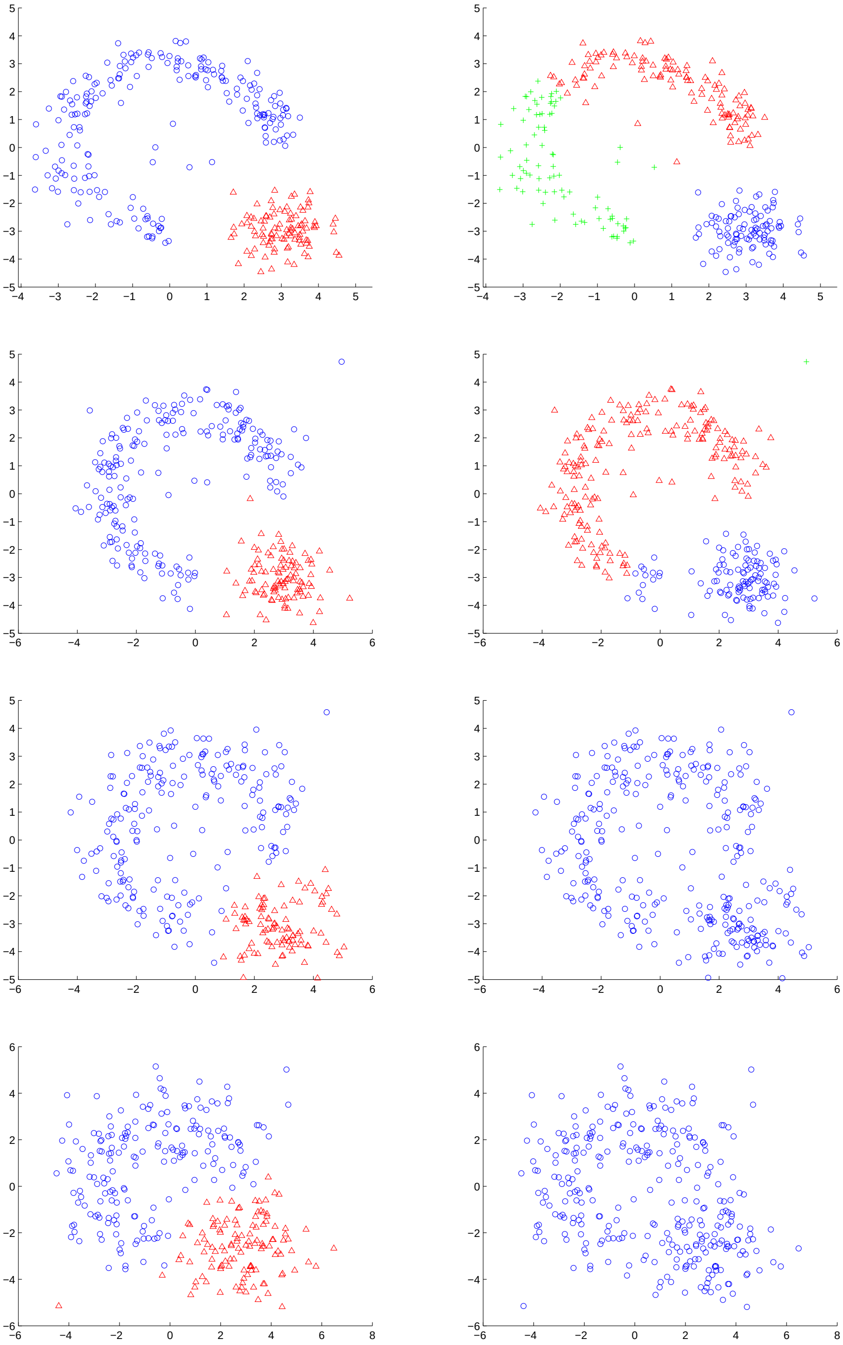

Fig. 3. Performance of `Speclus` (left) and DaSpec (right) for artificial data sets DS1, DS2, DS3, and DS4.

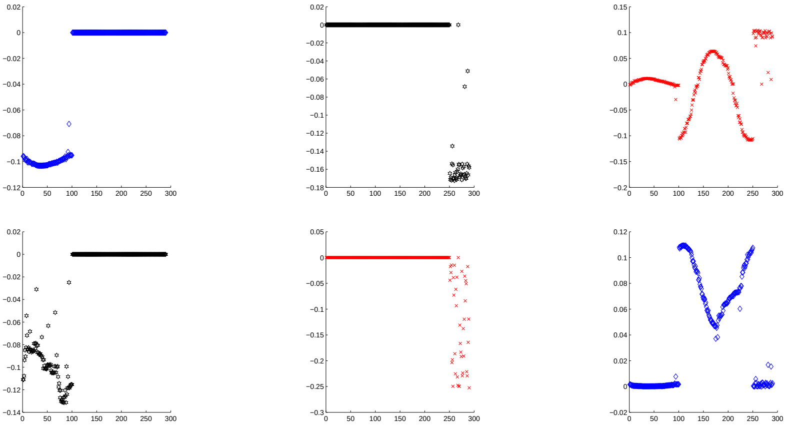

Fig. 4. Comparison of ordinary Laplacian (top) and signless Laplacian (bottom) eigenvectors for 2RG dataset.

tering algorithms. Our solution correctly separates different types of clusters with varying densities with and within groups, being simultaneously noise-resistant. It has only one parameter, which is quite easy to establish. The `Speclus` algorithm does not require a group number as an input parameter and estimates it correctly using eigenvectors structure and a modularity function.

These observations show that our algorithm is a good candidate to apply it to image segmentation, that will be our next task.

References

1. Cvetković D.: Signless Laplacians and line graphs. Bull. Acad. Serbe Sci. Arts, Cl. Sci. Math. Natur., Sci. Math. 131, No. 30, pp. 85–92 (2005)
2. Deepak, V., and Meila, M.: Comparison of Spectral Clustering Methods. UW TR CSE-03-05-01 (2003)
3. Elon, Y.: Eigenvectors of the discrete Laplacian on regular graphs a statistical approach. J. Phys. A: Math. Theor.41 (2008)
4. Fischer, I., and Poland, J.: Amplifying the Block Matrix Structure for Spectral Clustering. Technical Report No. IDSIA-03-05, Telecommunications Lab (2005)
5. Jain, A. Murty, M., and Flynn, P. Data clustering: A review. ACM Computing Surveys, 31, pp. 264–323 (1999)
6. Jain, A.: Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31, pp. 651–666 (2010)
7. MacQueen, L.: Some methods for classification and analysis of multivariate observations. In: LeCam, L. and Neyman, J. (eds.) 5th Berkeley Symposium on Mathematical Statistics and Probabilitz, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
8. Maier, M., Hein, M., and von Luxburg, U.: Cluster identification in nearest-neighbor graphs. In: Proc. of the 18th International Conference on Algorithmic Learning Theory, ALT'07, Springer, Berlin, Germany, pp. 196–210 (2007)

9. Meila, M., and Shi, J.: A random walks view of spectral segmentation. In: Proc. of 10th International Workshop on Artificial Intelligence and Statistics (AISTATS), pp. 8–11 (2001)
10. Newman, M.E.J.: Detecting community structure in networks. European Physics. J. B 38, pp. 321-330 (2004)
11. Ng, A., Jordan, M., and Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14, pp. 849–856 (2001)
12. Shi, T., Belkin, M., and Yu, B.: Data spectroscopy: eigenspace of convolution operators and clustering. The Annals of Statistics, vol. 37 6B, pp. 3960–3984. (2009)
13. von Luxburg, U.: A tutorial on spectral clustering. J. Statistics and Computing, 17(4), pp. 395– 416 (2007)
14. Xia, T., Cao, J., Zhang, Y., and Li, J.: On defining affinity graph for spectral clustering through ranking on manifolds. Neurocomputing 72 (1315), pp. 3203-3211 (2008)
15. Xu, and R., Wunsch II, D.: Survey on clustering algorithms. IEEE Trans. on Neural Networks, 16(3), pp. 645–678 (2005)
16. Zelnik-Manor, L., and Perona, P.: Self-tuning spectral clustering. In: Proc. of NIPS'04, pp. 1601–1608 (2004)