Spectral Clustering Based on Analysis of Eigenvector Properties

Malgorzata Lucińska, Slawomir Wierzchoń

Spectral Clustering Based on Analysis of Eigenvector Properties

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

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

Abstract. In this paper we propose a new method for choosing the number of clusters and the most appropriate eigenvectors, that allow to obtain the optimal clustering. To accomplish the task we suggest to examine carefully properties of adjacency matrix eigenvectors: their weak localization as well as the sign of their values. The algorithm has only one parameter — the number of mutual neighbors. We compare our method to several clustering solutions using different types of datasets. The experiments demonstrate that our method outperforms in most cases many other clustering algorithms.

Keywords: spectral clustering, nearest neighbor graph

1 Introduction

Clustering is one of the most important research topics in both machine learning and data mining communities. It means an unsupervised classification of observed data into different subsets (clusters) such that the objects in each subset are similar while objects in different subsets are dissimilar one to another. Clustering has been applied in many research areas, like image segmentation [11], machine learning, and bioinformatics [13], to name a few.

A fundamental, and largely unsolved, problem in cluster analysis is the determination of the number of groups in a dataset. Numerous approaches to this problem have been suggested over the years (consult [2] for further details). The most common procedure is to use the number of clusters as a parameter of the clustering method and to select it from a maximum reliability criteria. The second approach uses statistical procedures (for example the sampling with respect to a reference distribution). Unfortunately, many of the methods require strong parametric assumptions, or to be computation-intensive, or both. Usually they include clustering algorithms as a preprocessing step.

Spectral clustering techniques [8], [16] belong to the most popular and efficient clustering methods. They use eigenvalues and eigenvectors of a suitably chosen matrix to partition the data. The matrix is an adjacency matrix (or a matrix derived from it) built on the basis of pairwise similarity of objects to be grouped. If it is clearly block diagonal, its eigenvalues and eigenvectors will relate back to the structural properties of the set [11]. In such a case the number

of clusters is usually given by the value k, that maximizes the eigengap (difference between successive eigenvalues). Then the k principal eigenvectors are used for clustering the original data. However, an adjacency matrix generated from real-world data is virtually never block-diagonal, regardless of a similarity measure. In such situations an open issue of key importance in spectral clustering is choosing not only the proper number of groups but also the right eigenvectors, that reveal the structure of the data.

The `SpecLoc2` algorithm, proposed in this paper, provides a solution for both the problems. We have developed an alternative approach to choosing the number of clusters and the most appropriate eigenvectors, which allow to obtain the optimal grouping. Our method is based only on spectral analysis of the adjacency matrix of the data points to be clustered. We have exploited carefully properties of adjacency matrix eigenvectors. The proposed algorithm constitutes an extension of the `SpecLoc` algorithm, our previous work [9]. The `SpecLoc` algorithm utilizes absolute values of weakly localized eigenvectors, which correspond to different clusters and reveal the structure of the data. Weak localization is characterized by slow decay of the component values away from its main existence subregion [3].

In the presented algorithm we use not only weak localization of eigenvectors, but also we take into consideration the sign of their values. The new method is more general than the original `SpecLoc`, and can be applied to much wider range of clustering problems, than the previous one. There is no need to search for parameters resulting in weak localization of eigenvectors. Practically all the spectra (within the area of spectral methods usefulness) allow to employ the new way of establishment of the cluster number.

We present an automated technique, which does not use any additional clustering processing, and verify our approach using well known real-world datasets. The performance of the `SpecLoc2` algorithm is competitive to other solutions that require the number of clusters to be given as a parameter.

In section 2 the notation and related terms are presented. The next section describes some important properties of graph eigenvectors. Then, in section 4, we have presented the policy of selecting eigenvectors that reveal the structure of dataset. The main concepts used in the `SpecLoc2` algorithm are explained in section 5. Section 6 includes the description of experiments and results obtained with the use of the `SpecLoc2` algorithm. Finally, in section 7, the main conclusions are drawn.

2 Notation and definitions

The set of data points to be clustered will be denoted by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$. For each pair of points i, j a similarity $s_{ij} \in [0, 1]$ is attached. The value $s_{ij} > 0$ implies the existence of the undirected edge $i \sim j$ in the graph G spanned over the set of vertices $\mathbf{X}$. The matrix $S = [s_{ij}]$ plays a role of the adjacency matrix for G.

Let $d_i = \sum s_{ij}$ denote the degree of node i and let D be the diagonal matrix with d_i's on its diagonal. In an undirected graph, the degree of a node is given by the number of its adjacent edges. It can also be defined as the sum of the weights of its adjacent edges.

The Laplacian matrix associated with graph G is the $n \times n$ matrix $L = D - S$. The normalized Laplacian, is defined as: $L_{sym} = D^{-1/2} L D^{-1/2}$. Its complement, $\mathbb{I} - L_{sym}$, is used in the NJW algorithm [12], which serves as a comparison with our solution in section 6.

The right eigenvector associated to the second smallest eigenvalue of the Laplacian matrix is called the Fiedler vector [2]. It carries significant structural information regarding the connectivity of the graph and forms the basis of spectral graph partitioning heuristics, see, e.g. [16] for a review. As the Fiedler vector has both positive and negative values, the signs of the values are used to partition the graph into two components: one associated with positive and the other with non-positive values. The original theorem proposed by Fiedler is presented in the next section.

3 Properties of graph eigenvectors

Fiedler has proved in [2] that if G is a connected graph and y is the eigenvector corresponding to the second eigenvalue of the Laplacian matrix L then one of the following two cases occurs:

- There is a single block B_0 in G which contains vertices with both positive and negative values of y. Each other block has either vertices with only positive, or only negative, or only zero y values.
- No block of G contains vertices with both positive and negative y values. Each block contains either vertices with only positive, or only negative, or only zero y values.

The eigenvector corresponding to the first (smallest) eigenvalue has only nonpositive or only nonnegative values. This is the result of the fact that the sum of each row of the Laplacian equals zero. Thus, multiplying L by a constant vector x, we state that $Lx = 0 = 0 \cdot x$.

If we consider e.g. three infinitely far apart clusters, the adjacency matrix is block diagonal and consists of three blocks. Its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros) [12].

In [3] Filoche *et al.* study the behavior of eigenfunctions for a complex domain Ω with a bottleneck separating it into two subregions. In any partially separated subregion, an eigenfunction of Ω has only two possible choices: (1) either its amplitude is very small throughout this subregion, or (2) this eigenfunction mimics one of the subregion own eigenfunctions. The subregions are disjoint for a few dominant eigenfunctions. However, for the next eigenfunctions, initially disjoint subdomains begin to merge to form larger subregions. After reaching the critical point, completely new fully delocalized modes can appear.

Combining Fiedler's theorem with Filoche's observations one can generalize this theory to eigenvectors of graphs consisting of partially separated subgraphs. In the sequel by the graph eigenvectors we will understand eigenvectors of the graph adjacency matrix. In Figure 1 on the left, four principal eigenvectors of the adjacency matrix of a graph corresponding to two close and well separated sets of points are depicted. On the right we can see the two principal eigenvectors for the second subset only. Entries of the first and the second eigenvector (the left picture) are large only on one subset. They include the first eigenvectors of the two subsets, whereas the third eigenvector includes the second eigenvector of the second subset. The fourth eigenvector, with large entries for both subsets, emerges as a new mode consisting of eigenvectors of both subgraphs. The first three eigenvectors are weakly localized on one of the subset, contrary to the fourth one.

In a real situation, where the subgraphs are weakly separated, the picture is distorted and eigenvectors of the subgraphs mix with one another in the spectrum of the whole graph. For the clustering purposes the most appropriate are these eigenvectors that include the first or the second principal eigenvectors of its subgraphs. They mimic local eigenvectors of subgraphs of the whole graph and have one of the following form:

- eigenvectors with both large positive and negative values, and possibly near-zero entries, including the second eigenvectors of the subgraphs (Figure 1 left, the third eigenvector)
- eigenvectors with large entries for the relevant subgraph and near-zero values for the rest of vertices, including the first eigenvectors of the subgraphs (Figure 1 left, the first eigenvector) .

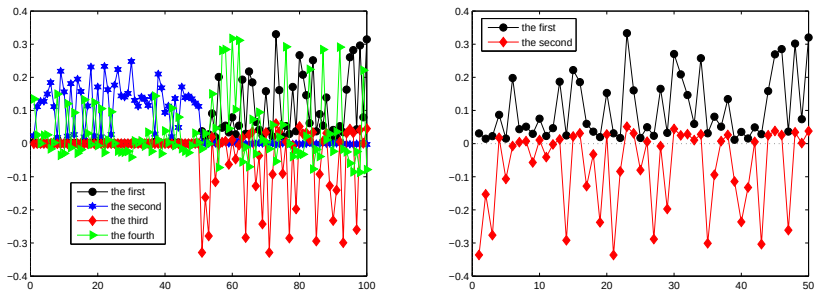

Fig. 1. The four principal eigenvectors of the adjacency matrix of the set consisting of two well separated groups (left) and two principal eigenvectors of the second subset of the set (right).

4 Preliminaries

Spectral clustering algorithms can be classified according to two approaches: recursive two-way spectral clustering algorithms (e.g. [6]) and direct K-way spectral clustering algorithms (e.g. [12]). The former finds the Fiedler eigenvector of the Laplacian matrix of a graph G and recursively partitions G until a K-way partition is found. The latter uses the first K eigenvectors and directly finds a partition using some heuristics. The K-way algorithms utilize K principal eigenvectors, however they do not take into consideration the special structure properties of the eigenvectors, using usually K-means algorithm for the final partitioning.

In the `SpecLoc2` algorithm we have applied the approach that utilizes ideas used in both of the described above solutions. It is also an extension of our previous work, presented in [9]. Similarly to the SpecLoc algorithm we perform clustering on the basis of a few localized eigenvectors, but also increase the flexibility of the former solution by exploiting properties of other eigenvectors. We take into consideration not only absolute values of eigenvectors, but also the sign of an eigenvector entry.

In order to explain in an intuitive manner our policy we will analyze adjacency matrix eigenvectors of the well known dataset Iris [15]. It consists of three groups, the first one can be separated very easily whereas the second and third ones are very close to one another.

The adjacency matrix is constructed on the basis of the k-nearest neighbor graph. The way of constructing the k-nearest neighbor graph is described in section 5.

We compare eigenvectors of two different graphs obtained on the basis of two different numbers of the nearest neighbors for the Iris dataset. Figure 2 shows the first three principal eigenvectors of the Iris adjacency matrix in two cases. The figure on the left illustrates the situation, when a small k ($k = 5$) results in a very sparse adjacency matrix and its principal eigenvectors are weakly localized. Each of them mimics the first eigenvector of the appropriate subregion.

As the number of the nearest neighbors increases and the matrix becomes less sparse, weak localization of eigenvectors disappears. Figure 2 (right) shows the case when $k = 30$ and the adjacency matrix eigenvectors have completely different shapes. The second eigenvector remains still localized in the first cluster, which is well separated from the others. The first eigenvector is localized in the second and third cluster, whereas the sign of the third eigenvector allows to distinguish between the overlapping clusters. We can see that the third eigenvector structure falls in with the second case of the Fiedler's theorem. It mimics the structure of the second eigenvector of the subregion consisting of the overlapping clusters.

In order to partition sets with different structures we have to take into consideration not only weakly localized eigenvectors but also the ones that have both positive and negative values. Identifying both types of eigenvectors enables us also to establish the number of clusters. As some groups are indicated by weakly localized eigenvectors and the others by eigenvectors with both positive

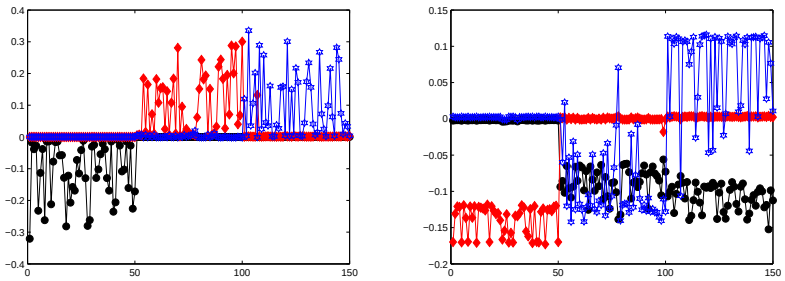

Fig. 2. The three principal eigenvectors of the adjacency matrix of the Iris dataset for k=5 (left) and k=30 (right)

and negative values, the number of clusters equals the number of localized eigenvectors of one sign plus the clusters corresponding to the eigenvectors that have positive and negative values and include an eigenvector described in the second part of the Fiedler's theorem.

In real datasets, the challenging task is to identify weakly localized eigenvectors, each representing different cluster and to distinguish between vectors changing sign between two groups and within one group.

5 The SpecLoc2 algorithm

The main steps of the SpecLoc2 algorithm are similar to these of the SpecLoc algorithm and they look in the following way:

The SpecLoc2 algorithm

```
Input: Data X, the number of nearest neighbors k
1. Form the adjacency matrix S.
2. Find c principal eigenvectors of S.
3. Calculate the eigenvector correlation matrix
4. Find uncorrelated weakly localized eigenvectors on the basis
    of the correlation matrix (eigenvector set WL).
5. Identify eigenvectors having positive values for one cluster
    and negative for others or vice versa (set PN).
6. Assign points to eigenvectors from the WL set
    and eigenvectors from the PN set.
```

The algorithm builds a graph, with points as vertices and similarities between points as edges. The weights of edges are calculated according to the Euclidean distance, using:

$$s_{ij} = \exp\Big(-\frac{d_{ij}}{d_{max}}\Big) \tag{1}$$

where d_{ij} is the Euclidean distance between objects i and j, and d_{max} is the maximum distance between any pair of objects from the dataset. On the basis of the metric we construct the k-nearest neighbor graph, connecting x_i to x_j if x_i is among the k-nearest neighbors of x_j. The algorithm uses the adjacency matrix because it is the simplest, nonnegative, and symmetric one. Its eigenvectors differ from L eigenvectors but they still obey assumptions of the Fiedler's theory. The vectors that take part in the partitioning are established on the basis of the pairwise correlation coefficients between each pair of c eigenvectors (c equals 20 in the algorithm, as we have assumed that the examined sets consist of maximum 20 clusters). The Pearson correlation coefficient between two eigenvectors v_i and v_j is defined as:

$$R_{ij} = \frac{\sum_{k=1}^{N}(v_{ik} - \overline{v_i})(v_{jk} - \overline{v_j})}{\sqrt{\sum_{k=1}^{N}(v_{ik} - \overline{v_i})^2}\sqrt{\sum_{k=1}^{N}(v_{jk} - \overline{v_j})^2}} \tag{2}$$

The coefficients R_{ij} range from -1 to 1. If two eigenvectors are linearly dependent, then the correlation between them will equal 1. The value of -1 indicates a perfect negative linear relationship between the vectors, and zero no linear relationship between the vectors. The correlation coefficient has been chosen as an efficient indicator whether two vectors have independent entries. If they are weakly localized in two different subregions their entries should differ from one another, so that they are linearly independent.

For the purpose of the algorithm we use absolute values of the eigenvector components in order to compute correlation coefficients. We have assumed that two eigenvectors are not localized in the same cluster if their correlation is smaller than 0.1 (the value has been established experimentally). Completely delocalized eigenvectors (with large, only nonpositive or only nonnegative values over the whole set) do not take part in calculation of the correlation coefficients, as they are not useful for partitioning purposes. We assume, that if the eigenvector median is higher than its standard deviation, the eigenvector is delocalized. The median is efficient location and dispersion measure for distributions revealed by delocalized eigenvectors.

Weakly localized eigenvectors, which are not correlated with any eigenvector related to higher eigenvalue, are written to the set WL and used for the final labeling of the data. Each eigenvector represents one cluster and each point is labeled according to the eigenvector with the highest entry for the point. Points assigned to the same eigenvector are further divided into two groups if the weakly localized eigenvector has both positive and negative values.

The next step of the `SpecLoc2` algorithm is the identification of eigenvectors with relatively large both positive and negative values, which do not belong to the set WL. We have to distinguish between an eigenvector, that enables to separate two different clusters, and a vector that changes its sign within one group of vertices. According to the Fiedler's theorem the eigenvectors, we are interested in, occur after delocalized eigenvectors or eig envectors with only positive or only negative values in their region of localization. Because of the spectra perturbation the vector allowing to distinguish between two clusters

sometimes will not appear as the second eigenvector in the complete spectrum, as it is the case for the Iris dataset, illustrated in Figure 2 right. Moreover, a few such vectors can exist. In order to find the right eigenvectors we check the vectors that appear after the localized ones or after a completely delocalized eigenvector (if it exists). We take into consideration only those that have the maximum values in that points, for which the already chosen eigenvectors are small (less than 0.1). Usually only one or two such eigenvectors are worth examination.

Next the assignment of points to the right partition is performed according to the procedure described below. For the purpose of partitioning two eigenvectors v and w are used. The first one is the vector with both positive and negative values from the set PN, whereas the second one serves for comparison. It is the closest to v vector in the set WL, related to a higher eigenvalue than v.

Partitioning on the basis of the Fiedler vectors

```
Input: A pair of vectors: the vector v and w
1. Find the sign of the maximum value vm of v absolute values
2. Set null values in v for all the points having opposite
    sign than the vm
3. Label each point x, having larger absolute value in v than in w
```

As each of the chosen weakly localized eigenvectors represents one cluster (or two if it changes its sign) and the vectors with positive and negative values divide unambiguously the set, we do not have to indicate the number of clusters manually or with the help of any other quality measure.

Computational complexity of the proposed algorithm is relatively small. First of all the adjacency matrix is sparse as we use the concept of k-nearest 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 the adjacency matrix we seek for the principal eigenvectors, which are easier to find than eigenvectors corresponding to the smallest eigenvalues. In such the situation solving the eigenproblem even for a large dataset is not very time consuming. The other steps of the algorithm take time $O(n)$ each. So the solution is scalable.

6 Experiments

In this section we justify our approach by presenting a set of clustering experiments and comparing its performance to different solutions. The algorithms are evaluated on a several benchmark datasets, including both synthetic and real-world examples. They cover a wide range of difficulties that can be met during data segmentation. The algorithm was implemented in MATLAB.

In the first part of our experiments we would like to emphasize differences in the performance of the `SpecLoc2` and the NJW [12] algorithms and show the dominance of our approach over the traditional solution. The NJW algorithm

uses the similarity measure based on the Gaussian kernel function, defined as:

$$W_{ij} = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\Big) \tag{3}$$

where $\|x_i - x_j\|$ denotes the Euclidean distance between points x_i and x_j and σ kernel parameter. In order to compare the best achievements of the algorithms the values of the σ parameter were chosen manually, as described by Fischer et al. in [5]. For each dataset they have systematically scanned a wide range of σ and ran the clustering algorithm. We use their results in this comparison.

In this part of experiments we would like not only to compare the two approaches, but also show that the K-means algorithm, employed for the final partitioning in the NJW algorithm in many cases does not take into consideration some important information included in the eigenvector structure, contrary to our solution. We have modified the NJW algorithm by using the adjacency matrix built on the basis of the mutual k-nearest neighbor graph for the normalized Laplacian construction. Thanks to this both solutions use the same graph.

The algorithms are evaluated on the following benchmark datasets: 2RG (two rather high density rings and a Gaussian cluster with very low density), 4G (four Gaussian clusters each of different density in 3D), Iris, and Jain's toy problem [7]. Table 1 summarizes the partitioning results obtained by the `SpecLoc2`, NJW, and modified NJW algorithms.

Table 1. Comparison of classification errors for partitioning with the `SpecLoc2`, NJW, and modified NJW algorithms

Dataset	**NJW**	**Modified NJW**	`SpecLoc2`
2RG	6	6	**3**
4G	18	13	**8**
Iris	14	8	**7**
Jain	19	33	**2**

As can be seen the `SpecLoc2` algorithm is the most flexible one and performs well independently on a dataset structure. Although both the `SpecLoc2` and the NWJ algorithms use the same concept of eigenvector properties the second one often fails on real-world data or clusters with different densities. We will explain the reasons why such results are observed, with the help of the 4G set. The NJW algorithm uses the four principal eigenvectors in order to obtain correct partitions. The fourth eigenvector, representing the sparsest cluster, is quite distorted and causes wrong partitioning of some points. For the `SpecLoc2` algorithm only the three principal vectors suffice for the correct partitioning (two uncorrelated localized eigenvectors and the vector with positive and negative values). In the other cases (e.g. the Wine dataset) the better performance of the `SpecLoc2` algorithm lies in small differences between two eigenvectors entries, which do not influence K-means partitioning but are taken into account by the presented method.

We have also compared the performance of the `SpecLoc2` algorithm to other methods, similarly as in [14]:

- NJW algorithm.
- Self-tune spectral clustering (SSC) algorithm [18], which computes automatically the scale and the number of groups and can handle multi-scale data.
- The KASP algorithm [17], fast approximate spectral clustering in which a distortion-minimizing local transformation is first applied to the data, based on local K-means clustering.
- The KWASP algorithm [19] that extends the Nyström method and improves the approximation of the eigensystem by introducing the probability density function as a natural weighting scheme.
- The Kernel-K-means-Ratio Assoc (KKRA) algorithm [1], that directly optimizes various weighted graph clustering objectives, such as the popular ratio cut, normalized cut, and ratio association.
- Fast Affinity Propagation clustering approach (FAP) [14] that simultaneously considers both local and global structure information contained in datasets.

In case of the NJW, SSC, and KKRA algorithms the adjacency matrix is computed as in [14].

We performed experiments on five UCI datasets, including Wine, Balance, Segments, Pendigits, and Optdigits [15]. The basic information of those real-world datasets are summarized in Table 2. Digits389 is a subset of the three classes 3, 8, 9 of the UCI handwritten digit recognition dataset from the UCI Machine Learning Repository– these three classes were chosen since distinguishing between sample handwritten digits from these classes visually is a difficult task.

Table 2. A summary of datasets

Dataset	Size	Dimensions	Classes
Wine	178	13	3
Balance	625	4	3
Segment	2310	18	7
Optdigits	1151	64	3
Pendigit-test	3498	16	10
Pendigit-train	7494	16	10

All the datasets are labeled, which enables evaluation of the clustering results against the labels using normalized mutual information (NMI) as a measure of division quality. We refer an interested reader to [10] for details regarding the measure.

The performance of the algorithms shows Table 3. We can see the superiority of the SpecLoc2 algorithm over the other tested solutions. The presented method is competitive to the other cases in terms of the quality of partitioning, measured with the help of the normalized mutual information.

Clustering results of our solution are stable, their do not depend on starting, randomly chosen settings, as those using K-means clustering. Moreover they are not very sensitive to the number of neighbors (the only parameter in our algorithm). In case of the Wine, Digits389, and Pendigit sets the NMI changes only by a few percent within a range of 10 subsequent number of neighbors. As the quality of partitioning for the Balance and Segment sets is not very high, the NMI remains stable for even much wider range of k values.

Table 3. Comparison of NMI for partitioning with different algorithms

Dataset	NJW	SSC	KKRA	KASP	KWASP	FAP	SpecLoc2
Wine	0.41587	0.43157	0.38027	0.40507	0.43447	0.44577	**0.8518**
Balance	0.14647	0.26267	0.21617	0.13057	0.18947	0.20467	**0.5278**
Segment	0.49407	0.56807	0.57117	0.56587	0.53117	0.64087	**0.6560**
Digits389	0.52647	0.60957	0.79357	0.76017	0.75417	0.90237	**0.9213**
Pendigit-test	0.68617	0.67997	0.69867	0.68967	0.68077	0.73627	**0.8239**
Pendigit-train	0	0	0.70357	0.70517	0.68167	0.75497	**0.8025**

Our approach shows the ability of discovering clusters that are difficult to distinguish. The SpecLoc2 algorithm has found the right number of clusters in case of the Wine, Optdigits, and Pendigit-train datasets, for Balance it has failed to detect one group and Pendigit-test has been divided into 11 clusters instead of 10. Only in case of the Segment dataset it was not able to find 3 groups.

We have compared the performance of the SpecLoc2 algorithm with our previous method the SpecLoc algorithm, which is also able to determine the number of clusters. In some cases, as for example for the Iris set, the results are the same for both solutions. However, in many cases the presented algorithm outperforms the older one either in terms of NMI or on account of the determined number of clusters. There are the cases, where the vectors with positive and negative values have great influence on partitioning. Moreover, the SpecLoc2 algorithm is less demanding, when it comes to tuning the parameter of nearest neighbors. It performs well in both localized and delocalized cases.

7 Conclusions

In this work, we have proposed a new approach for spectral clustering. Its goal is to make maximal use of information derived from the eigenvector structure in order to improve the quality of spectral partitioning and limit the number of parameters. We have analyzed the properties of eigenvectors and proposed methods for selecting the ones, that reveal the dataset structure in the best way.

Our algorithm is not only efficient, but also flexible to work with different cases that may occur in real-world datasets. We have used several UCI benchmark datasets to validate the advantage of our approach, by comparing to the classic and new clustering algorithms. Empirical results show that our method can find the true cluster assignment by using only one parameter, and it outperforms the other methods, even specifying more parameters.

References

1. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: a multilevel approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), pp. 1944–1957 (2007).
2. Fiedler, M.: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Mathematical Journal 25, 619–633 (1975)
3. Filoche M. and Mayboroda, S.: Universal mechanism for Anderson and weak localization, In: Proc. of the National Academy of Sciences, 109(37), pp. 14761–6 (2012)
4. Fridlyand, J., Dudoit, S.: A prediction-based resampling method to estimate the number of clusters in a dataset. Genome Biology, 3, 7 (2002).
5. Fischer, I., Poland, J.: Amplifying the Block Matrix Structure for Spectral Clustering. Technical Report No. IDSIA-03-05, Telecommunications Lab, pp. 21–28 (2005)
6. Hagen, L., Kahng, A.: New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design 11, 1074–1088 (1992)
7. Jain, A., M. Law: Data clustering: A user's dilemma. Lecture Notes in Computer Science, 3776 pp. 1–10 (2005)
8. Kannan, R., Vempala, S., Vetta, A.: On clusterings: good, bad and spectral. In: 41st Symposium on Foundations of Computer Science, FOCS (2000)
9. Lucińska, M.: A spectral clustering algorithm based on eigenvector localization. In: Pros. of the ICAISC 2014, Part II, LNAI 8468, pp. 761–771, (2014)
10. Manning, Ch.D., Raghavan, P., Schtäuze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
11. Meila M., Shi, J.: A random walks view of spectral segmentation. In: Proc. of 10th International Workshop on Artificial Intelligence and Statistics (2001)
12. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems, vol. 14. MIT Press, Cambridge, pp. 849–856 (2002)
13. Pentney, W., Meila, M.: Spectral clustering of biological sequence data. In: AAAI 2005, pp. 845–850 (2005)
14. Shang, F., Jiao, L.C., Shi, J., Wang, F., Gong, M.: Fast affinity propagation clustering: A multilevel approach, Pattern Recognition 45 pp. 474–486 (2012)
15. `http://mlearn.ics.uci.edu/MLRepository.html`
16. von Luxburg, U.: A Tutorial on spectral clustering, Statistics and Computing, 17(4), pp. 395–416 (2007)
17. Yan, D., Huang, L., Jordan, M.: Fast approximate spectral clustering, In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) pp. 907–916 (2009)
18. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering, In: Proceedings of the Advances in Neural Information Processing Systems, vol.17, pp. 1601–1608 (2005)
19. Zhang, K., Kwok, J.T.: Density-weighted Nyström method for computing large kernel eigen-systems, Neural Computation 21 pp. 121-146 (2009)