European Journal of Operational Research 211 (2011) 221–231

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

Spectral methods for graph clustering – A survey

Mariá C.V. Nascimento*, André C.P.L.F. de Carvalho

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa Postal 668, São Carlos-SP, CEP 13560-970, Brazil

ARTICLE INFO

Article history:
Received 20 January 2010
Accepted 11 August 2010
Available online 16 August 2010

Keywords:
Spectral clustering
Min-cut
Ratio cut
ncut
Modularity

ABSTRACT

Graph clustering is an area in cluster analysis that looks for groups of related vertices in a graph. Due to its large applicability, several graph clustering algorithms have been proposed in the last years. A particular class of graph clustering algorithms is known as spectral clustering algorithms. These algorithms are mostly based on the eigen-decomposition of Laplacian matrices of either weighted or unweighted graphs. This survey presents different graph clustering formulations, most of which based on graph cut and partitioning problems, and describes the main spectral clustering algorithms found in literature that solve these problems.

1. Introduction

Clustering is an unsupervised technique concerned with the grouping of related objects without taking their class or label into account. It is expected that objects belonging to the same cluster are more similar to each other than to objects belonging to different clusters. Cluster analysis has been applied to many areas, like, for example, gene expression analysis (Huttenhower et al., 2007), natural language processing (Ushioda and Kawasaki, 1996), galaxy formation (White and Frenk, 1991) and image segmentation (Wu and Leahy, 1993). Due to the wide range of applications, many clustering algorithms based on distinct principles have been proposed.

Clustering algorithms can be roughly divided into two main groups: hierarchical and partitioning algorithms. This division affects the solution representation and the algorithmic approach for generating clusters. In the traditional hierarchical algorithms, clusters are gradually formed through agglomerations (agglomerative algorithms) or divisions (divisive algorithms). These algorithms create a tree structure where either subgroups or supergroups of clusters are progressively built. Partitioning clustering algorithms split the dataset into $k \leqslant n$ groups, where n is the number of objects in the dataset. Solutions are obtained by moving objects between clusters until a stop criterion is satisfied. Each solution is assessed by a given objective function.

In the last years, graph clustering algorithms have become very popular. According to Schaeffer (2007), graph clustering groups vertices of a graph into clusters, based on the edge structure of the graph. The resulting vertex partition should have the property that within each cluster the vertices are highly connected whereas there are only few edges between clusters.

Graph clustering has been thoroughly studied and many different approaches have been investigated (Huttenhower et al., 2007). These approaches either explicitly adopt the concepts of graph theory in their formulations or just manipulate the graph-based data structure. In graph clustering, graph theory can provide the necessary definitions and mathematical formalism, resulting in an important support for the analysis of graph clustering models. Reviews on graph clustering can be found in Cormack (1971), Hansen and Jaumard (1997), Schaeffer (2007), Filippone et al. (2008). Specifically, Schaeffer (2007) discusses some of the most frequently used approaches, reporting many publications on this issue.

The main graph clustering formulations are based on graph cut and partitioning problems (Alpert et al., 1999; Chan et al., 1994; Shi and Malik, 2000). An alternative to heuristically solve these problems is to use spectral clustering algorithms. In the last decades, there has been a growing interest in spectral clustering algorithms, mainly because of their efficiency and mathematical elegance. Moreover, they have the advantage of providing lower/upper bounds for minimization/maximization graph cut and partitioning problems, due to their spectral relaxation. Examples of spectral clustering algorithms can be found in Ng et al. (2002), Saerens et al. (2004), Ding et al. (2005), Forman et al. (2005), Filippone et al. (2008). Another research issue related to graph clustering is clustering of graphs. This issue is concerned with finding clusters in a set of graphs based on their structural similarity (Schaeffer, 2007). In order to keep the focus of this survey, this issue will not be covered in the text.

* Corresponding author. Tel.: +55 16 33738632.
E-mail addresses: mariah@icmc.usp.br (M.C.V. Nascimento), andre@icmc.usp.br (A.C.P.L.F. de Carvalho).

0377-2217/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2010.08.012

The goal of this review is to survey the main spectral clustering algorithms found in literature for graph cut and graph partitioning problems. All the spectral graph theory necessary to understand these algorithms will be presented either before or during their descriptions. Moreover, important basic graph concepts are presented for those who are not familiar with graph notations and representations.

This survey is organized as follows. Section 2 presents graph concepts, definitions, terminologies and properties related with the issues covered in this text. Section 3 presents graph cut problems important for clustering. Section 4 shows the most representative spectral clustering algorithms. Section 5 presents experimental comparisons of different types of spectral clustering algorithms. Finally, Section 6 concludes the survey discussing future trends in spectral clustering algorithms research.

2. Graphs and Laplacians

Let $G = (V,E)$ be a graph, where V is a non empty set of n *nodes* (or *vertices*) and E is a set of m *edges*. Each edge in E can be defined by the pair (v_i, v_j), where v_i and v_j are nodes of G, i.e., elements from V. A *subgraph* of G is a graph G', $G' = (V',E')$, where $V' \subset V$ and $E' \subset E$. A *spanning subgraph* of G is a subgraph that contains all nodes of G. In this survey, we will only deal with undirected graphs, i.e., graphs without edge directions.

The *adjacency matrix* of G is a binary matrix, given by $A = [a_{ij}]_{n\times n}$, where $a_{ij} = 1$, if there is an edge connecting nodes v_i and v_j, and 0, otherwise. Moreover, weights may be associated with the graph's edges, resulting in weighted graphs. The edge weights are represented by a *weight matrix*, $W = [w_{ij}]_{n\times n}$, where $w_{ij} \in \mathbb{R}$ represents the edge weight between nodes v_i and v_j. If the edges of a graph have no weight, the graph is known as an *unweighted* graph. In an undirected graph, the degree of a node is given by the number of its adjacent edges. In an undirected weighted graph, the degree of a node can also be defined as the sum of the weights of its adjacent edges.

A *path* in a graph is a sequence of nodes with an edge connecting every two consecutive nodes. A *connected component* is a subgraph with a path connecting any pair of nodes. When there is at least one path where one node appears more than once, the graph is referred to as a *cyclic graph*, as opposed to an *acyclic graph*. If there is an edge connecting any two nodes of a graph, then the graph is considered to be a *complete graph*. A *tree* is a connected acyclic graph. A *spanning tree* is a connected acyclic spanning subgraph.

The cut associated with a set of nodes X of a graph G is the set of all its edges with an extremity in X and another in the complement of X, $V\backslash X$. A partition $\pi = \{C_1, C_2, \ldots, C_k\}$ of the set of nodes of a graph G is defined by k subsets of nodes, each one denoted by C_i, where $i = 1,\ldots,k$. Moreover, $V = \bigcup_{i=1}^{k} C_i$ and $\bigcap_{i=1}^{k} C_i = \emptyset$. A k-way partitioning, here denoted by π^k, is a partition of V into k parts such that $C_i \neq \emptyset$, for $1 \leqslant i \leqslant k$.

There are several other important concepts in graph theory (Diestel, 2005; Schrijver, 2003), but the aspects presented in this section are the most relevant for the understanding of the algorithms presented in the next sections. The following section shows some constructions of similarity graphs from datasets.

2.1. The construction of similarity graphs

There are cases where data are not originally structured in graphs. In these cases, a similarity graph can be constructed from these data. For such, consider an undirected weighted graph $G = (V,E)$ where each node v_i is represented by the ith object from a given dataset. The edges of G are defined according to a similarity measure between pairs of objects from this dataset. One of the most frequently used similarity measures is given by the sigmoid function. For such, let $d(i,j)$ be the dissimilarity between objects i and j from the dataset, like, for example, the Euclidean distance. Then, the weight matrix W of a similarity graph G from the given dataset can be calculated by making $w_{ij} = e^{-d(i,j)^2/\sigma^2}$, if $i \neq j$, and 0, otherwise.

The parameter σ has a high impact on the clustering partition obtained. Different strategies have been investigated to find its best value. One of them, presented in Ng et al. (2002), consists in running a clustering algorithm for different values of σ. The σ that provides the least squared intra-cluster distance to its centroid is chosen. However, this method might not be effective as some additional parameters need to be set and due to its high computational cost.

Another strategy was the one suggested by Zelnik-Manor and Perona (2004), where a local scaling of the parameter σ was proposed. In this approach, the parameter σ was tuned according to the distance between the pair of objects being evaluated. The values of the weights in the weight matrix W were calculated as: $w_{ij} = e^{-d(v_i,v_j)^2/\sigma_i\sigma_j}$ if $i \neq j$ and 0, otherwise. $\sigma_i = d(v_i, v_\kappa)$, where v_κ is defined as the κth neighbor of v_i. Again, an additional parameter, called κ, needed to be tuned, and was manually fixed by the authors for their experiments.

The similarity graph resulting from this strategy can either be used as a complete graph or be processed in order to eliminate some of its edges. An alternative for the latter case is the elimination of the edges of a similarity graph whose weights are lower than a predefined threshold. Further details and more options for constructing similarity graphs can be found in Von Luxburg (2007). Moreover, (Maier et al., 2009) is a good source for additional information about the impact of different graph constructions on graph clustering results. Since spectral clustering algorithms are based on the eigen-decomposition of graph Laplacian matrices, these matrices will be discussed in this survey.

2.2. Spectral graph partitioning

The study of spectral graph theory started in Quantum Chemistry, with a theoretic model of non saturated hydrocarbon molecules (Hückel, 1931; Cvetković et al., 1979). These molecules have chemical linkages with many electron energy levels. Some of these energy levels can be represented by the eigenvalues of a graph. The study of eigenvectors and eigenvalues of a squared matrix is the essence of the spectral theory. The theoretical foundation of spectral graph theory started with the works by Hall (1970), Fiedler (1975), Cvetković et al. (1979) and was further developed in the last two decades by Mohar (1991), Chung (1994), Kim and Choi (2006).

The application of spectral theory to graph clustering problems is usually based on the relaxation of some graph partitioning problems. Spectral clustering algorithms are generally based on fast iterative methods and can benefit from the use of linear algebra packages, like the linear algebra package (LAPACK) (Anderson et al., 1990). In the next section, we show some properties of graph Laplacian matrices important for the comprehension of the spectral clustering algorithms presented in Section 4.

2.2.1. Some properties of Laplacian matrices

Consider a graph $G = (V,E)$ and its weighted matrix W, such that $w_{ij} \geqslant 0$ for $i, j = 1,\ldots,n$. Let $D = [d_{ij}]_{n\times n}$, with $d_{ij} \in \mathbb{R}$, be a diagonal matrix defined by $d_{ii} = \sum_{j=1}^{n} w_{ij}$, i.e., d_{ii} is the degree of node v_i, with $i = 1,\ldots,n$. For simplicity reasons, d_{ii} will be referred to here as d_i. The unnormalized graph Laplacian matrix, defined by $L = [l_{ij}]_{n\times n}$ is given by

$$L = D - W. \tag{1}$$

If a graph is unweighted, consider its adjacency matrix A instead of the weight matrix W in Eq. (1): $L = D - A$.

The Laplacian matrix L is also known as the Kirchhoff matrix, due to its role in the *Matrix-Tree-Theorem* (Mohar, 1991). According to this theorem, for two vertices v_i and v_j of a graph G, the absolute value of the determinant of the matrix produced with the elimination of the ith row and the jth column of the matrix L is equal to the number of spanning trees of G.

Laplacian matrices are the heart of the majority of the spectral clustering algorithms. For this reason, some theorems and properties concerning the Laplacian matrix L, considered to be relevant for the spectral relaxation of graph partitioning problems, are presented next. These statements are based on the study made by Hagen and Kahng (1992), Von Luxburg (2007). Matrix L has the following properties.

(i) For each $x \in \mathbb{R}^n$, it is assured that $x^tLx = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2$, where x_i is the ith component of x.
(ii) L is a symmetric positive-semi-definite matrix.
(iii) The smallest eigenvalue of L is $\lambda_1 = 0$ with the associated eigenvector $\mathbb{1}$, where $\mathbb{1}$ is the indicator vector, i.e., $\mathbb{1} = (1, \ldots, 1)^t$.

To prove Property (i), consider the following equalities:

$$x^tLx = x^tDx - x^tWx = \sum_i d_i x_i^2 - \sum_{i,j} w_{ij}x_ix_j = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2.$$

By Property (i), we have that L is positive-semidefinite, because the weights of the graph are all positive, which implies that $x^tLx \geqslant 0$. As a consequence, all its eigenvalues are non-negative. Furthermore, L is symmetric, since $L = D - W = D^t - W^t$, which proves Property (ii). Property (iii) can be easily deduced, because all eigenvalues are greater than or equal to zero. Moreover, $L\alpha\mathbb{1} = 0\alpha\mathbb{1}$, $\alpha \in \mathbb{R}$ and $\alpha \neq 0$. Consequently, *0* is an eigenvalue of L. The multiplicity of this eigenvalue indicates the number of connected components of a graph.

Taking the last property into account, an approximation for the number of natural clusters can be specified. Considering $\mathscr{C}_i$ as the ith connected component of a graph G, we have $\mathbb{1}_{\mathscr{C}_i}$ is an eigenvector associated with the eigenvalue 0. $\mathbb{1}_{\mathscr{C}_i}$ is the vector with the value 1 in the positions relative to the order of the vertex in the connected component $\mathscr{C}_i$, i.e., if the vertex v_j belongs to the connected component $\mathscr{C}_i$, then the jth position of vector $\mathbb{1}_{\mathscr{C}_i}$ is 1, otherwise, it is 0. As a result, the property found in Boccaletti et al. (2006), which claims that if a connected component presents a structure with k apparent clusters (not necessarily perfect), its Laplacian matrix will have, besides the null eigenvalue, $k - 1$ eigenvalues significantly close to 0. The other eigenvalues will be significantly larger than 0.

In this paper, the eigenvalues of the Laplacian matrices are considered in their increasing order, $\lambda_1 \leqslant \lambda_2 \leqslant, \ldots, \lambda_n$, always taking their multiplicities into account, except when another order is re-defined a priori. Furthermore, the eigenvectors do not need to be normalized and the first k eigenvectors correspond to the eigenvectors associated with the first k eigenvalues arranged in the predefined order.

A particular eigenvalue of L that represents relationships among connected components of a graph is λ_2. Fiedler (1975) suggested that the eigenvector associated with λ_2, named Fiedler eigenvector, can be used to find an approximation for the graph bipartitioning problem. This approximation is based on the signs of the components of the Fiedler eigenvector. Fiedler presented an expression for this eigenvalue and named it the algebraic connectivity of a graph.

Unlike the formulation proposed by Fiedler (1975), the following equation for λ_2 of L is more commonly used:

$$\lambda_2 = \min_{x\perp\mathbb{1}, x\neq\mathbb{0}} \frac{x^tLx}{x^tx} = \min_{x\perp\mathbb{1}, x\neq\mathbb{0}} \frac{1}{2}\frac{\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2}{||x||^2}, \tag{2}$$

where $\mathbb{0}$ is the null vector, i.e., $\mathbb{0} = (0, \ldots, 0)^t$.

Two other Laplacian matrices will be discussed in this paper. The first, the normalized Laplacian matrix, is given by Eq. (3):

$$L_N = I - D^{-1/2}WD^{-1/2}. \tag{3}$$

The second, the random walk Laplacian matrix, is illustrated by Eq. (4):

$$L_{rw} = I - D^{-1}W. \tag{4}$$

Some properties of these matrices are discussed in (Von Luxburg, 2007). A number of them will be presented later in this paper, when necessary. All notations used in this section and in Section 2 will be adopted in the remainder of the text. For example, G will always be a graph, as defined in Section 2. The next section presents different graph cut and partitioning problems for which spectral methods can be found in the literature.

3. Graph cut and partitioning problems

A large number of graph clustering algorithms are based on graph partitioning problems. This survey is concerned with a particular class of these algorithms, known as spectral clustering algorithms. Spectral clustering algorithms are mostly based on the solution of graph cut problems. For such, they use one or more eigenvectors from Laplacian matrices of a graph to be partitioned that are solutions for the relaxation of some graph cut problems. The most commonly used graph cut problems and their spectral algorithms will be presented in the next section.

The k-way partitioning problem aims at eliminating edges from a graph in order to produce k connected subgraphs. Its goal is to produce well separated clusters, represented by the k connected subgraphs. Several different criteria can be considered when choosing the set of edges to be eliminated. Some of these criteria, as well as their advantages and limitations, will be presented next. In all of them, consider $W(C_r, C_t) = \sum_{\substack{v_i \in C_r \\ v_j \in C_t}} w_{ij}$, $\bar{C}_i$ is every cluster C_j such that $j \neq i$ and Π^k is the set of all possible k-way partitions of G.

3.1. Minimum cut problem

The first problem to be presented is the k-way minimum cut problem:

$$\min_{\pi^k \in \Pi} cut(\pi^k), \tag{5}$$

where:

$$cut(\pi^k) = \frac{1}{2}\sum_{i=1}^{k} W(C_i, \bar{C}_i). \tag{6}$$

It aims at minimizing the sum of weights of the edges whose nodes come from different clusters. Wu and Leahy (1993) observed that, in many tested graphs, the solutions of this problem are partitions with isolated nodes in clusters. This might be a drawback for many applications, such as in VLSI domain (Alpert et al., 1999). Other formulations, which take additional aspects into account, were proposed to solve this limitation. For example, Alpert et al. (1999) considered some lower and upper bounds to delimit the size of the clusters of a partition π^k. In this case, a set of constraints can be added to the k-way min-cut formulation:

$\mathscr{L}_i \leqslant | C_i | \leqslant \mathscr{U}_i$, for $1 \leqslant i \leqslant k$. The parameters $\mathscr{L}_i$ and $\mathscr{U}_i$ are, respectively, the lower and the upper bounds for the size of the i-th cluster and $|C_i|$ is the number of nodes in cluster C_i.

3.2. Minimum ratio cut problem

Another approach to avoid finding partitions with isolated nodes in clusters is to consider Eq. (5) divided by the number of elements in each cluster. This formulation was first proposed to solve the bipartitioning problem (Leighton and Rao, 1988; Wei and Cheng, 1989), also known as two-way ratio cut problem. Later, this formulation was generalized by Chan et al. (1994) for the k-way ratio cut problem through its connection with the weighted quadratic placement problem formulated by Hall (1970). The k-way minimum ratio cut formulation is represented by Eq. (7):

$$\min_{\pi^k \in \Pi^k} ratiocut(\pi^k) \tag{7}$$

where:

$$ratiocut(\pi^k) = \frac{1}{2}\sum\nolimits_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{|C_i|}. \tag{8}$$

3.3. Minimum normalized cut problem

An alternative to the k-way minimum cut problem is to divide the sum parcel $W(C_i, \bar{C}_i)$ by the sum of the node degrees of the objects inside the cluster C_i, which is given by $vol(C_i) = \sum_{j \in C_i} d_j$. This problem is known as the k-way *ncut* problem and is shown in Eq. (9):

$$\min_{\pi^k \in \Pi^k} ncut(\pi^k), \tag{9}$$

where:

$$ncut(\pi^k) = \frac{1}{2}\sum\nolimits_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{vol(C_i)}. \tag{10}$$

The k-way *ncut* problem was proposed by Shi and Malik (1997, 2000) and was derived from the relation between the normalized association and dissociation measures of a partition. The former measure reflects the average connectivity of the nodes within each cluster of a partition. The latter measures the cut cost of a partition as a percentage of the cut edge connections regarding all the nodes in a graph. Shi and Malik (1997, 2000) concluded that the problem of finding a partition that minimizes the dissociation inter-clusters is the same as the problem of finding a partition that maximizes the association inside each cluster.

3.4. Min–max cut problem

A distinct cut problem was proposed by Ding et al. (2001, 2004), which creates the min–max cut problem for clustering. The min–max cut formulation aims at minimizing the inter-cluster similarities maximizing the intra-cluster similarities, as can be seen in Eq. (11):

$$\min_{\pi^k \in \Pi^k} MinMaxCut(\pi^k) \tag{11}$$

where:

$$MinMaxCut(\pi^k) = \sum\nolimits_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{W(C_i, C_i)} \tag{12}$$

3.5. Modularity maximization problem

Modularity is a quality validation measure for graph clustering partitions (Newman and Girvan, 2004). As illustrated by Eq. (13), it measures, in a partition π, the difference between the sum of the edge weights connecting nodes inside cluster and the expected sum of edge weights connecting nodes inside each cluster of the same partition in a random graph:

$$q(\pi) = \frac{1}{2m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij} - \frac{d_i d_j}{2m}\right)y_{ij}, \tag{13}$$

where $m = \frac{\sum_i^n d_i}{2}$ and y_{ij} is a variable whose value is 1 if nodes v_i and v_j belong to the same cluster, and 0, otherwise. It can be noticed that the sum of intra-cluster edge weights is counted twice. This is the reason for the multiplication by 1/2. The multiplication by $1/m$ is due to the normalization of the measure, assuring that its value is less than or equal to 1. The portion $\frac{d_i d_j}{2m}$ indicates the probability of having the edge (v_i, v_j) in a random graph with the same node degree sequence as the original graph G. Throughout the text, this portion will be referred to as p_{ij}. If the graph is unweighted, $w_{ij} = a_{ij}$. The closer the value of $q(\pi)$ is to 1, the better the connectivity of the partition π. In order to find partitions that maximize the modularity measure, the following mathematical formulation can be used:

$$\max_{\pi \in \Pi} q(\pi), \tag{14}$$

where Π is the set of all possible k-way partitions of G, with $k = 1, \ldots, n$.

Spectral algorithms have been extensively used to solve the previous problems as will be seen in the next section.

4. Spectral clustering algorithms

As mentioned previously, an alternative to solve graph partitioning problems is to use the spectral graph theory. A proper relaxation of graph partitioning problems enables the exploration of the eigenvalues and eigenvectors properties of their Laplacian and adjacency matrices. One positive aspect of these methods is the possibility of defining upper or lower bounds for the objective function of the graph partitioning problems. Some promising results of the application of spectral graph theory to data clustering can be found in computer vision (Shi and Malik, 2000; Luo et al., 2003), VLSI design (Alpert et al., 1999) and detection of clusters in protein structures (Kannan and Vishveshwara, 1999).

Spectral clustering algorithms can be classified according to two approaches: recursive two-way spectral clustering algorithms and direct k-way spectral clustering algorithms. The former finds the Fiedler eigenvector of a Laplacian matrix of a graph G and recursively partitions G until a k-way partition is found. The latter uses the first $d \geqslant k$ eigenvectors and directly finds a partition using some heuristics. Throughout this paper, algorithms based on these two approaches will be introduced, highlighting the main aspects of each of them.

4.1. Two-way partitioning algorithms

The two way min-cut problem without constraints that delimit the size of the clusters can be efficiently solved using the max-flow, min-cut theorem (Ford and Fulkerson, 1962). Hagen and Kahng (1992) proposed a spectral clustering algorithm using a recursive two-way partitioning approach for the k-way ratio cut problem. This algorithm was based on a study about the one-dimensional quadratic placement (assignment) problem (Hall, 1970). Hagen and Kahng, 1992 established the connection between the eigenvectors of the Laplacian matrix L of a graph G and the solution of the relaxed two-way ratio cut problem. To show this connection, consider a two-way partition $\pi^{(2)}$ of G and a vector x, $x \in \mathbb{R}^n$, where $x_i = |C_2|/n$ if $v_i \in C_1$, and $-|C_1|/n$, otherwise. Notice that $x \perp \mathbb{1}$, since

$\sum_i x_i =\mid C_1 \mid\mid C_2 \mid /n- \mid C_2 \mid\mid C_1 \mid /n = 0$. Moreover, $x_i - x_j = (|C_2| + |C_1|)/n = 1$ if $x_i \neq x_j$, and 0, otherwise.

By considering Property (i) of the Laplacian matrix L:

$$x^t Lx = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2 = cut(C_1, C_2). \tag{15}$$

Besides, $\|x\|^2 = |C_1||C_2|/n$. Since $\lambda_2 = \min_{x\perp\mathbb{1},x\neq\mathbb{0}} \frac{1}{2}\frac{\sum_{i,j=1}^{n} w_{ij}(x_i-x_j)^2}{\|x\|^2}$, it is possible to show that:

$$\lambda_2 \leqslant \frac{cut(C_1, C_2)n}{|C_1||C_2|} \Rightarrow ratiocut(C_1, C_2) \geqslant \frac{\lambda_2}{n}.$$

As a result, Hagen and Kahng, 1992 found a lower bound for the two-way ratio cut problem. The relaxation of the two-way min-cut problem is given by $\min_{x\perp\mathbb{1},x\neq\mathbb{0}} \frac{1}{2}\frac{\sum_{i,j=1}^{n} w_{ij}(x_i-x_j)^2}{\|x\|^2}$, whose solution vector is provided by the Fiedler eigenvector of the Laplacian matrix L. With this result, the authors proposed a recursive two-way spectral clustering algorithm for the k-way ratio cut problem. This algorithm, known as EIGl, is based on the linear ordering of the Fiedler eigenvector.

Algorithm 1. EIGl

```
1: Input: A graph $G = (V,E)$; its Laplacian matrix $L$; and $r$, a threshold value
2: Find the second eigenvalue $\lambda_2$ of $L$ and its associated eigenvector $x^{(2)}$ using the Lanczos algorithm
3: For all $v_i \in V$, if $x_i^{(2)} > r$, $v_i \in C_1$, otherwise, $v_i \in C_2$
4: Output: The resulting partition
```

In algorithm EIGl, whose pseudocode can be seen in Algorithm 1, the inputs are a graph G; its Laplacian matrix L; and r, which is a threshold value that produces the partition with the best ratio cut cost.

Later, Von Luxburg (2007) presented a different way to define x for the same formulation: $x_i = \sqrt{\mid C_2 \mid / \mid C_1 \mid}$, if $v_i \in C_1$, and $-\sqrt{\mid C_1 \mid / \mid C_2 \mid}$, otherwise. With this modified vector, $x^t Lx = ratiocut(C_1,C_2)n$. In this case, $\|x\|^2 = n$ and x is orthogonal to the n-dimensional vector $\mathbb{1}$. Conducing the linear relaxation of x, subject to the previous constraints $\|x\|^2 = n$ and to be orthogonal to $\mathbb{1}$), x must be the second eigenvector of the Laplacian matrix L.

In order to find out a feasible solution to the two-way ratio cut problem from the Fiedler eigenvector, other heuristics different from EIGl can be adopted (Hagen and Kahng, 1992): (a) if $x_i^{(2)} \geqslant 0$ then $v_i \in C_1$, otherwise, $v_i \in C_2$; (b) if $x_i^{(2)} \geqslant med$ then $v_i \in C_1$, otherwise, $v_i \in C_2$, where med is the median of $x^{(2)}$; (c) sort $x^{(2)}$ to a vector y and find the largest *eigengap* between two consecutive elements: $lgap = max_{1\leqslant i\leqslant n-1} eigengap_i$, where $eigengap_i = |y_i - y_{i+1}|$. If $x_i^{(2)} \geqslant lgap$ then $v_i \in C_1$, otherwise, $v_i \in C_2$.

The linear ordering always finds a solution better than or equal to these previous heuristics. According to the authors, the complexity of EIGl is dominated by the order of the Lanczos method, which is $O(n + |E|)$. If G is a complete graph, the algorithm has complexity $O(n^2)$.

Shi and Malik (1997, 2000) proposed a recursive two-way *ncut* algorithm for the k-way *ncut* problem. The authors proved that the minimization of the two-way *ncut* problem can be relaxed to the Rayleigh quotient. For such, consider the two-way *ncut* problem rewritten in the following way:

$$\min_{\pi^k \Pi^k} ncut(\pi^k) = \min_{x\neq 0, x^T D\perp\mathbb{1}} \frac{x^T Lx}{x^T Dx}, \tag{16}$$

where $x \in \mathbb{R}^n$ is defined by $x_i = 1$, if $v_i \in C_1$, and $-vol(C_1)/vol(C_2)$, otherwise. With this definition, it is possible to deduce that $x^T D\mathbb{1} = 0$ and that $x^t Lx = ncut(C_1,C_2)vol(V)$. For more details about these deductions, see (Von Luxburg, 2007). By relaxing x such that $x \in \mathbb{R}^n$, the problem can be seen as the Rayleigh quotient. To solve the relaxed problem, it is necessary to solve the generalized eigenvalue system:

$$Lx = \lambda Dx. \tag{17}$$

For such, consider $y = D^{1/2}x$ and $L_N = D^{-1/2}LD^{-1/2}$. Thus, Eq. (17) can be rewritten as

$$L_N y = \lambda y. \tag{18}$$

Matrix L_N is a Laplacian matrix, known as the normalized graph Laplacian matrix, and is also positive semi-definite. Let $x^{(i)}$ and $y^{(i)}$ be the ith eigenvectors of System (17) and of the matrix L_N, respectively. As a result of the Raleigh–Ritz theorem, $y^{(2)} = \arg\min_{y^T y^{(1)}=0} \frac{y^T L_N y}{y^T y}$ and $x^{(2)} = \arg\min_{x^{(1)T} D\mathbb{1}=0} \frac{x^T Lx}{x^T Dx}$.

Therefore, the relaxed solution of the generalized Eigensystem (18) is given by its second eigenvector, resulting in $x = D^{-1/2}y^{(2)}$. It is worth to observe that x is the second eigenvector of the Laplacian matrix L_{rw}.

Based on the previous deductions, Shi and Malik (2000) proposed their two-way normalized spectral clustering algorithm (2NSC). For such, in Algorithm 1, consider as inputs a graph G; its Laplacian matrix L_N; and r. The parameter r has the same role as the parameter r in algorithm EIGl. However, in this case, r is the real number that makes the algorithm produce the partition with the best *ncut* value. Again, the complexity of this algorithm is dominated by the running time of the Lanczos algorithm.

Regarding the min–max cut formulation, Ding et al. (2001, 2004) developed a graph partitioning algorithm to solve problems according to this formulation. Their algorithm follows the same principle found in Algorithm 1, using as input a graph G; its matrix L_{rw}; and the value of r that provides the partition with the best *ncut* value.

Concerning the modularity maximization problem, Newman (2006) proposed a recursive two-way spectral clustering algorithm. To understand how it works, a vector x can be defined according to a two-way partition $\pi^{(2)}$ as follows: $x_i = 1$, if $v_i \in C_1$, and -1, otherwise. Thus, the previously mentioned function y_{ij}, defined in Section 3.5, can be rewritten as $y_{ij} = \frac{1}{2}(x_i x_j + 1)$. In this equation, if v_i and v_j belong to the same cluster, then $y_{ij} = 1$, and *0*, otherwise. Therefore, Eq. (13) can be reformulated as

$$q(\pi) = \frac{1}{4m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij} - p_{ij}\right)\left(x_i x_j + 1\right). \tag{19}$$

Since the objective function of the considered problem tries to maximize the modularity measure, the constant term of Eq. (19) can be eliminated. Considering a matrix $B = [b_{ij}]_{n\times n}$, such that b_{ij} = -$w_{ij} - p_{ij}$, the following expression can be used for the modularity measure:

$$q(\pi) = \frac{1}{4m}x^t Bx. \tag{20}$$

Notice that the objective function of the relaxation of the two-way ratio cut problem, Eq. (15), is similar to Eq. (20), except for the constant factor and the matrix B playing the role of the matrix L.

Some inferences about the matrix B can be made. As $d_i = \sum_j w_{ij}$ and $\sum_j p_{ij} = d_i$ (this last equation is derived in Newman (2006), the sum of each row (and column) of B is equal to 0. Moreover, for any weight matrix W, the n-dimensional vector $\mathbb{1}$ is the eigenvector of B associated with the eigenvalue 0, similar to the Laplacian matrix L. However, unlike L, which just allows non-negative eigenvalues, B can have both positive and negative eigenvalues. According to

Newman (2006), these observations are tied to the clustering structure of a graph.

Since B is a symmetric matrix, its eigenvectors u_i associated with the eigenvalues λ_i, with $i = 1,\ldots,n$, are orthogonal to each other and, therefore, form a basis for $\mathbb{R}^n$. Since this is a maximization problem, the eigenvectors can be sorted in decreasing order of eigenvalues: $\lambda_1 \geqslant \lambda_2,\ldots, \geqslant \lambda_n$. Considering $U = [u_{ij}]_{n\times n}$ to be a matrix whose columns indicate each of the normalized eigenvectors of B, it is easy to see that $U^TU = I$. This implies that $x = U^TUx \Rightarrow x = \sum_{j=1}^{n}(u_j^Tx)u_j$. Considering $\alpha_j = u_j^Tx$, it is possible to rewrite x as $x = \sum_{j=1}^{n}\alpha_j u_j$.

Since $x^Tx = n$, then $\sum_j \alpha_j^2 = n$. The substitution of this expression in Eq. (20) results in:

$$q(\pi) = \frac{1}{4m}\sum_i \alpha_i^2 \lambda_i. \tag{21}$$

As the problem aims at finding the partition with the largest modularity, the best solution is achieved when the largest eigenvalues (positive eigenvalues) are considered in the equation. Based on these considerations, Newman (2006) proposed a two-way modularity heuristic, named leading eigenvector algorithm (LE). To describe it, in Algorithm 1, consider as input a graph G; its matrix B; and the parameter $r = 0$. Besides, instead of the Fiedler eigenvector $x^{(2)}$, the largest eigenvector of the matrix B should be used to define the partition. According to some tests performed by the authors, the method works satisfactorily when looking for two-way partitions.

In order to find a k-way partition, all previously presented two-way partitioning algorithms can be recursively applied to the clusters until a k-way partition is obtained. The cluster selected to be split at each iteration can be, for example, the cluster that produces the best objective function criterion (Ding et al., 2001, 2004; Newman, 2006). However, according to the authors, if a graph has a structure with more than two clusters, these algorithms may produce unsatisfactory and unstable results. Moreover, these two-way algorithms may have a high computational cost, since they require the reconstruction of the Laplacian matrix and the calculation of its eigenvalues and eigenvectors for each iteration of the heuristic. Alternatively, many authors proposed direct k-way spectral clustering algorithms based on these problems. These algorithms will be described in the next section.

4.2. k-way partitioning algorithms

Let us start the discussion of k-way partitioning algorithms with the algorithms proposed to solve the k-way min-cut problem. The algorithm introduced by Alpert et al. (1999) is based on a reduction from the k-way min-cut problem to the vector partitioning problem. The vector partitioning problem looks for a k-way partition $P^k = \{P_1,\ldots,P_k\}$ in a set of vectors Y. Lower and upper bounds are employed to delimit the size of the sets P_i, with $1 \leqslant i \leqslant k$. The function to be optimized is given by

$$f(P^k) = \sum_{j=1}^{k} ||Y_j||^2, \quad \text{where} \quad Y_j = \sum_{\vec{y}\in P_j}\vec{y}.$$

This function may either be minimized (min–sum vector partitioning problem) or maximized (max–sum vector partitioning problem). To perform the reduction, the k-way min-cut problem was reformulated as follows:

$$\max_{\pi^k} nH - cut(\pi^k), \quad \text{where} \quad H \geqslant \lambda_n, \quad \text{and} \quad \mathscr{L}_i \leqslant |C_i| \leqslant \mathscr{U}_i, \quad \text{for } 1 \leqslant i \leqslant k.$$

Next, the authors defined a matrix $V_d = [v_{ij}]_{n\times d}$ of scaled eigenvectors, where:

$$v_{ij} = u_{ij}\sqrt{H - \lambda_j}, \tag{22}$$

and $U = [u_{ij}]_{n\times d}$ is the matrix of the first d eigenvectors of the Laplacian matrix L of a graph G, each one represented by a column of U. Considering $P = \{\vec{y}_1^n,\ldots,\vec{y}_n^n\}$, where $\vec{y}_i^n$ is the i-th row of V_n, the authors proved that if $\forall\ i = 1,\ldots,n$, $v_i \in C_l$ if and only if $y_i^d \in P_l$, then $nH - cut(\pi^k) = f(P^k)$.

Using this reduction, Alpert et al. (1999) proposed a greedy algorithm based on linear ordering, named multiple eigenvector linear orderings (MELO), whose pseudocode can be seen in Algorithm 2. After the linear ordering step, MELO generates the k-way final partition by using a dynamic programming procedure from Alpert and Kahng (1994).

Algorithm 2. MELO

```
Input: A graph $G = (V,E)$; the number of desired clusters, $k$;
  and the number of eigenvectors to be used, $d$
Construct the matrix of scaled eigenvectors $V_d$ according to
  Eq. (22), (set $P = \emptyset$)
Make $Y = [\vec{y}_i^{\,d}]_{n\times d}$, for $1 \leqslant i \leqslant n$, where $\vec{y}^d$ is the $i$-th row of
  $V_d$
for $j = 1$ to $n$ do
    Find $\vec{y}_i^{\,d}$, $1 \leqslant i \leqslant n$, that maximizes $||\sum_{\vec{y}\in P}\vec{y} + \vec{y}_i^{\,d}||$
    Add $\vec{y}_i^{\,d}$ to $P$ and remove $\vec{y}_i^{\,d}$ from $Y$
    Label $v_i$ as the $j$th vertex in the ordering
end for
Find the final $k$-way partition using the linear ordering
  found in the previous steps and the strategy proposed by
  Alpert and Kahng (1994)
Output: The final partition
```

Algorithm MELO has complexity $O(dn^2)$ and two parameters to be set: d and H. Alpert et al. (1999) analyzed the performance of MELO using integer values in the range [1,10] for d. The authors concluded that it is not possible to predict the influence of this parameter on the quality of the results. The best value for d depended on the configuration of the problem as well as the value chosen for k. Nevertheless, in general, $d > k$ eigenvectors are necessary in order to a good partition to be found. Regarding the parameter H, Alpert et al. (1999) tested different alternatives for its value, and the best results were achieved with $H = \lambda_2 + \lambda_d$. In order to calculate the Laplacian matrix eigenvectors, the authors also used the Lanczos algorithm.

Donath and Hoffman (1973) proposed a theorem that allows the calculation of a lower bound for the k-way min-cut problem. This lower bound is based on the eigenvectors of a matrix M_l resulting from the sum of the adjacency matrix of a graph G with any diagonal matrix. The unitary vector would be the eigenvector of M_l associated with the eigenvalue 0. In particular, if the matrix $-D$ is assumed to be this diagonal matrix, it satisfies these conditions, since $M_l = A - D = L$. For this reason, the eigenvalues and eigenvectors of the Laplacian matrix L (λ_i and u_i, respectively) are considered in the lower bound proposed by Donath and Hoffman (1973). This lower bound is given by the following inequality:

$$cut(\pi^k) \geqslant \frac{1}{2}\sum_{i=1}^{k}\lambda_i u_i. \tag{23}$$

Chan et al. (1994) presented a spectral relaxation of the k-way ratio cut problem illustrated by Eq. (7). Together with their proposal, they introduced a spectral heuristic based on the orthonormality among the eigenvectors of the Laplacian matrix L. For such, the authors represented a k-way partition π^k of a graph G through an assignment matrix $X = [x_{ij}]_{n\times k}$, where $x_{ij} = 1/\sqrt{|\,C_j\,|}$, if

$v_i \in C_j$, and 0, otherwise; and a rationed partitioned matrix $R = [r_{ij}]_{n\times n}$, with $r_{ij} = 1/|C_l|$, if $v_i, v_j \in C_l$, and 0, otherwise. Some relations between the matrices X and R and properties of the matrix X can be deduced and are useful for the understanding and development of the heuristic and lower bound presented in Chan et al. (1994).

It can be deduced that $x_j^t L x_j = ratiocut(C_j, \bar{C}_j)$ and that $x_j^t L x_j$ is the element of the jth row and column of $X^t LX$. Therefore, $x_j^t L x_j = (X^t LX)_{jj}$, resulting in:

$ratiocut(C_1, C_2, \dots, C_k) = \sum_{j=1}^{k} x_j^t L x_j = \sum_{j=1}^{k} (X^t LX)_{jj} = \mathrm{tr}(X^t LX)$, where $tr(M)$ is the trace of a matrix M. Moreover, with some simple algebraic manipulations, it can be observed that $X^t X = I$ and $XX^T = I$.

With these properties, it is possible to observe that:

$$\min_{\pi^k \in \Pi^k} \frac{1}{2} \sum_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{|C_i|} = \min_{X^T X = I} tr\left(X^T LX\right) = tr\left(U^T LU\right) = \sum_{i=1}^{k} \lambda_i,$$

where $U = [u_{ij}]_{n\times k}$, whose k columns are the first k normalized eigenvectors of the matrix L. The first equality is due to a proof by Chan et al. (1994), which is a version of the Raleigh–Ritz theorem. Therefore, $\sum_{i=1}^{k} \lambda_i$ is a lower bound for the k-way ratio cut problem. The authors also used this result to propose their clustering heuristic. As U is the solution of a relaxed form of the k-way ratio cut problem, i.e., it is an approximation for X, then, as $XX^T = I$, $Z = UU^T$ is an approximation for the matrix R. Considering $U^T = [u'_1, \dots, u'_n]$, each element of the matrix Z can be written as $z_{ij} = u_i'^T u'_j$.

Let $\widehat{P}$ be the directional cosine matrix of the rows of the matrix U. Each of its elements can be calculated by the equation: $\hat{p}_{ij} = \frac{u_i'^T u'_j}{\|u'_i\| \|u'_j\|}$. The idea behind the heuristic proposed by Chan et al. (1994) is to measure how close nodes are to each other by evaluating the cosine between pairs of rows from U. In this heuristic, the cosine between pairs of vectors is only calculated if necessary, since estimating $\widehat{P}$ in all iterations is an expensive operation. The heuristic is summarized by Algorithm 3, which has complexity $O(n(bk^2 + bk\log(n)))$, where $b = \max_{1 \leqslant i \leqslant n} d_i$.

Since, in some situations, the outsiders from Algorithm 3 share more edges between them than with nodes from the clusters, a variation of the algorithm is also proposed by Chan et al. (1994). In this variation, after each allocation of an outsider, the solution that merges the whole set of the remaining outsiders with the cluster that gives the best ratio cut is calculated. If this solution is better than the best ratio cut produced by merging one element from the set S with one cluster from the current partition, then all outsiders are assigned to the cluster that gives the best ratio cut.

More recently, other approaches that group the rows of the matrix U have been investigated. For example, Von Luxburg (2007) presented the Unnormalized k-means algorithm (Ukmeans), illustrated in Algorithm 4.

Algorithm 3. KP

```
1: Input: A graph $G = (V,E)$; and the number of desired
   clusters, $k$
2: Find the first $k$ eigenvectors of $L$ and sort them in the
   columns of the matrix $U$
3: Select $k$ nodes to represent each of the $k$ prototype clusters
   using their magnitude ($\|u'_i\|$) and their mutual (near)
   orthogonality relationship
4: During 4 iterations: Calibrate the $k$ prototypes by
   calculating their average with prototypes from the previous
   iteration and by the posterior selection of the closest node
   as the seed for each prototype
5: For every $i = 1,\dots,n$, verify if the cosine between $u'_i$ and one
   of the prototypes is higher than $\cos(\pi/8)$. If so, assign node $i$
   to the prototype with the largest cosine
```

* (continued)

Algorithm 3. KP

```
6: Let $S$ be the set of non-allocated nodes (outsiders)
7: For all $s \in S$, find the largest weight cut between $s$ and all
   existing clusters and mark the cluster (target) with the
   largest cut value (key)
8: While $S \neq \emptyset$, the outsider with the largest cut is inserted to
   its target and all targets and keys of the remaining neighbor
   outsiders are recalculated
9: Output: The final partition
```

Algorithm 4. Ukmeans

```
1: Input: A graph $G = (V,E)$; and the number of desired
   clusters, $k$
2: Find the first $k$ eigenvectors of $L$ and sort them in the
   columns of matrix $U$. The $i$-th row of the matrix $U$ will
   represent node $v_i$ from graph $G$
3: Apply the $k$-means algorithm to the matrix $U$ and find a $k$-
   way partition $\pi'^k = \{C'_1, \dots, C'_n\}$
4: Form the final partition assigning every node $v_i$, with
   $1 \leqslant i \leqslant n$, to cluster $C_l$, if the $i$th row of $U$ belongs to $C'_l$ in the
   partition $\pi'^k$
5: Output: The final partition
```

According to Von Luxburg (2007), there is no foundation for applying the k-means algorithm in step 3 of Algorithm 4. The author justifies that, as the Laplacian matrix translates many connectivity properties of a graph, this step of the algorithm should be simple. Therefore, the k-means algorithm is used. Nevertheless, other approaches can be explored for the same purpose.

Regarding the k-way *ncut* problem, a spectral clustering algorithm (KNSC) was introduced by Shi and Malik (2000) to solve it. KNSC is based on the first k eigenvectors of the generalized Eigensystem (17). To explain in more details how this algorithm works, Von Luxburg (2007) defined the binary matrix $X = [x_{ij}]_{n\times k}$, where $x_{ij} = 1$, if $v_i \in C_j$, and 0, otherwise.

By considering x_j as the jth column of the matrix X, $x_j^t L x_j = ncut(C_j, \bar{C}_j)$. Furthermore, it is possible to see that $X^t DX = I$. Thus, the k-way *ncut* problem can be rewritten in the following way:

$$\min_{\pi^k \in \Pi^k} \mathrm{tr}(X^t LX), \quad \text{subject to} \quad X^t DX = I, \quad \text{with} \quad X \quad \text{as previously defined}.$$

By relaxing the integrality conditions of X, $Y = D^{1/2}X$. If $L_N = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$ is considered, the following problem is a relaxation of the k-way *ncut* problem:

$$\min_{Y \in \mathbb{R}^{n\times k}} \mathrm{tr}\left(Y^t L_N Y\right), \quad \text{subject to} \quad Y^t Y = I.$$

The solution of this problem is the matrix with the first k eigenvectors of the matrix L_N arranged as columns of Y. Substituting X by $D^{-1/2}Y$ and taking some properties of L_N into account (Von Luxburg, 2007), the solution of the problem is the set of the first k eigenvectors of $L_{rw} = I - D^{-1}W$. Shi and Malik (2000) used this result in their spectral clustering algorithm. They applied the k-means algorithm to cluster the rows of the matrix Y, in which each row represents an object from a dataset. Algorithm KNSC is presented in Algorithm 5. Its complexity is equal to the complexity of the Lanczos algorithm.

Meilă and Shi (2001), Meilă and Xu (2004) analyzed the conditions for the set of k eigenvectors to be piecewise constant with relation to a partition π^k in the k-way *ncut* clustering algorithm proposed by Shi and Malik (2000). Consider P to be the stochastic matrix $P = D^{-1}W$. A set of eigenvectors $u_1,\ldots,u_k$ are said to be piecewise constant with relation to a partition π_k when, for every $v_i,v_j \in C_s$, $u_{i1} = u_{j1},\ldots,u_{ik} = u_{jk}$. The authors found that this condition is satisfied when $P_{is} = \sum_{v_j \in C_s} P_{ij}$ is constant for all $v_i \in C_{s'}$ with $\forall s' \in 1,\ldots,k$ and $P' = [P_{ss'}]$ with $s, s' \in 1,\ldots,k$ is non-singular.

Algorithm 5. KNSC

```
1: Input: A graph $G = (V,E)$; and the number of desired
   clusters, $k$
2: Find the first $k$ eigenvectors of the generalized Eigensystem
   (17) and sort them in the columns of the matrix $U$. The $i$th
   row of the matrix $U$ will represent node $v_i$ from graph $G$
3: Apply the $k$-means algorithm to the matrix $U$ and find a $k$-
   way partition $\pi'^k = \{C'_1,\ldots,C'_n\}$
4: Form the final partition assigning every node $v_i$, with
   $1 \leqslant i \leqslant n$, to cluster $C_l$ if the $i$th row of $U$ belongs to $C'_l$ in the
   partition $\pi'^k$
5: Output: The final partition
```

A k-way *ncut* spectral clustering algorithm proposed by Ng et al. (2002), named KNSC1 and summarized in Algorithm 6, differs from Algorithm 5 in the sense that the eigenvectors to be clustered are obtained from the Laplacian matrix L_N. However, like Shi and Malik (2000), its computational complexity is of the same order of the Lanczos algorithm.

Algorithm 6. KNSC1

```
1: Input: A graph $G = (V,E)$; and the number of desired
   clusters, $k$
2: Find the first $k$ eigenvectors of the Laplacian matrix $L_N$ and
   sort them in the columns of the matrix $U'$. Form matrix
   $U = [u_{ij}]_{n \times k}$ from $U'$ by normalizing each row of $U'$ using
   $u_{ij} = u'_{ij} / \sqrt{\sum_k u'^2_{ik}}$
3: The $i$th row of the matrix $U$ will represent node $v_i$ from
   graph $G$
4: Apply the $k$-means algorithm to the matrix $U$ and find a $k$-
   way partition $\pi'^k = \{C'_1,\ldots,C'_n\}$
5: Form the final partition assigning every node $v_i$, with
   $1 \leqslant i \leqslant n$, to cluster $C_l$, if the $i$th row of $U$ belongs to $C'_l$ in the
   partition $\pi'^k$
6: Output: The final partition
```

Additionally, Ng et al. (2002) studied the stability of clusters from a partition through an *eigengap* analysis according to the matrix perturbation theory (Stewart and Sun, 1990). If a graph has, for example, a three-cluster structure, then the difference between the third and forth eigenvalues of the matrix L is relatively large (with regard to the previous mutual differences).

Similar to Ng et al. (2002), Yu and Shi (2003) proposed a spectral algorithm for clustering based on the normalized graph Laplacian matrix L_N. Aiming at minimizing the k-way *ncut*, the first k eigenvectors of L_N are normalized before the final partition is calculated. This calculation is carried out by an iterative procedure.

Regarding the k-way min–max cut problem, Gu et al. (2001) investigated and presented a lower bound for this problem. The authors showed that:

$$MinMaxcut(\pi^k) = \sum_{i=1}^{k} \frac{x_i^T L x_i}{x_i^T W x_i} = \sum_{i=1}^{k} \frac{1}{y_i^T L_N y_i} - k, \tag{24}$$

where $y_i = D^{1/2}x_i / \|D^{1/2}x_i\|$. It is known that L_N is a positive semi-definite matrix (Von Luxburg, 2007). Considering Y the matrix whose columns are the vectors y_i, $i = 1,\ldots,k$, it is possible to state that $Y^TY = I$. Gu et al. (2001) reported that a lower bound for the *min–max* cut problem is given by

$$\min_{Y^TY=I, y_i^T L_N y_i > 0} \ \max_{1 \leqslant i \leqslant k} \frac{1}{y_i^T L_N y_i} = \frac{k^2}{\sum_{i=1}^{k} \lambda_i}, \tag{25}$$

where λ_i are eigenvalues from L_N set in decreasing order of value.

Finally, concerning the modularity maximization problem to find k-way partitions, Newman (2006) proposed a k-way spectral clustering algorithm. For such, they defined a matrix $X = [x_{ij}]_{n \times k}$, where $x_{ij} = 1$, if $v_i \in C_j$, and 0, otherwise. Each column of X represents a cluster of a k-way partition π^k. Therefore, y_{ij} can be re-formulated as $y_{ij} = \sum_{r=1}^{k} x_{ir}x_{jr}$. Thus, the modularity of a partition π^k of a graph G can be calculated by $q(\pi^k) = \frac{1}{2m}\sum_{i,j} \sum_{r=1}^{k} b_{ij}x_{ir}x_{jr} = tr(X^TBX)$. It is worth to remember that the columns of X are orthogonal to each other. Moreover, $X^TX = I$.

It is possible to write B as a function of its eigenvectors and eigenvalues matrices: $B = U\Lambda U^T$, where Λ is the diagonal matrix whose ith element of its diagonal is given by the eigenvalue λ_i associated with the eigenvector u_i. Thus, $q(\pi^k)$ can be rewritten as

$$q(\pi^k) = \sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_i (u_i^T x_j)^2. \tag{26}$$

By Eq. (26), $q(\pi^k)$ is maximized if the largest k positive eigenvalues of B and their associated eigenvectors (leading eigenvectors) are used. Observing that just the eigenvectors associated with the positive eigenvalues add a positive value to $q(\pi^k)$, it is possible to deduce an upper bound for the number of clusters, k, of the optimal partition from the modularity maximization problem. This upper bound would be the number of positive eigenvalues plus 1, named c here.

The k-way spectral clustering algorithm proposed by Newman (2006) relies on a *vector partitioning* algorithm. Newman (2006) presented a reduction from the modularity maximization problem to the vector partitioning problem based on a matrix of scaled eigenvectors (likewise the study performed by Alpert et al. (1999)). For this reduction, the function $q(\pi^k)$ was rewritten as

$$q(\pi^k) = n\alpha + \sum_{i=1}^{n} \sum_{j=1}^{k} (\lambda_i - \alpha) \left(\sum_{t=1}^{n} u_{ti}x_{tj} \right)^2, \tag{27}$$

where $\alpha = \frac{1}{n-c}\sum_{i=c+1}^{n} \lambda_i$ is a constant tunned after a study of the error minimization of the $q(\pi^k)$ approximation. Let $R = [r_{ij}]_{n \times k}$ be the matrix of scaled eigenvectors, where the ith row represents node v_i of a graph G and each of its element is defined by $r_{ij} = u_{ij}\sqrt{\lambda_j - \alpha}$, where $\alpha \leqslant \lambda_k$. Each vector $\vec{r}_i$, for $1 \leqslant i \leqslant n$, denotes a k-dimensional vector that corresponds to the i-th row of R. If the terms associated with the negative eigenvectors are eliminated from Eq. (27), (supposing that they correspond to the last $n - c$ eigenvalues), $q(\pi^k)$ could be approximated by $q(\pi^k) = n\alpha + \sum_{j=1}^{k} \|R_j\|^2$.

It is already known that $c \geqslant k$, since it is an upper bound for k. By considering $R_j = \sum_{v_i \in C_j} \vec{r}_i$, the problem becomes a vector partitioning problem. According to Newman (2006), this problem can be heuristically solved by a vector partitioning algorithm (similar to MELO's algorithm) adapted to this problem. The final partition can be refined using a local search. According to the author, the results obtained using this heuristic for $k > 2$ were acceptable.

In the literature, it is possible to find spectral clustering algorithms to graph partitioning problems different from those presented in this survey (Bach and Jordan, 2004; Zha et al., 2002). In particular, Bach and Jordan (2004) introduced a new cost function that evaluates how close the eigenstructure of a similarity matrix W is to a given partition. Zha et al. (2002) proposed a clustering algorithm and a lower bound for the minimum sum of squares of clusters from a partition using the first eigenvalues of a data matrix. In spite of the existence of many others spectral clustering algorithms, due to space constraints, we cover in this survey only the algorithms that we consider as milestones in the evolution of spectral clustering algorithms.

4.3. Some comments with respect to spectral clustering algorithms

A research topic closely related to spectral relaxation is semi-definite programming (SDP) relaxation. The connection between these topics is due to the use of eigen-decomposition of graph Laplacian matrices. A SDP relaxation aims to obtain a set of feasible solutions defined by a convex region. However, SDP is known to be computationally expensive, since its solution methods are mostly based on interior point strategies (Xing and Jordan, 2003), thus being efficient only for small datasets. Nevertheless, the spectral relaxation is based on results from spectral graph theory. Some SDP clustering algorithms can be found in literature (De Bie and Cristianini, 2006; Kim and Choi, 2006; Kulis et al., 2007; Singh et al., 2008).

According to Fowlkes et al. (2004), despite the growing use of spectral clustering methods, their application to large problems has not been explored. Their high computational cost reduces the chances of a good performance. To overcome this problem, the authors proposed the use of an extension of the Nyström method in order to find numerical approximations for the eigenvectors of a similarity graph.

Although they have been investigated for many decades, few spectral clustering algorithm studies tackled the consistency of the existing algorithms. Brand and Huang (2003) performed a theoretical study to analyze the reasons why spectral clustering algorithms based on the eigenvalues and eigenvectors of similarity matrices usually work. The authors investigated a nonlinear dimensionality reduction of similarity matrices through the matrix of their first k eigenvectors. They concluded that if the correlation between two points from the similarity matrix is high (low), the correlation between this pair of points from its eigenvector's matrix is even higher (lower). Recently, Von Luxburg et al. (2008) showed the superiority of the normalized spectral clustering algorithms over the unnormalized algorithms, pointing out their convergence under more general conditions.

5. Computational experiments

In order to experimentally evaluate and compare the performance of the two-way and k-way partitioning algorithms, experiments were carried out with the problems: minimization of ratio cut and *ncut*, and modularity maximization. For these experiments, the following graphs from literature were used: *adjnoun* (Newman, 2006), *dolphins* (Lusseau et al., 2003), *jazz* (Gleiser and Danon, 2003) and *karate* (Zachary, 1977). These experiments were all run using a code based on the Venables and Smith (2010). The algorithms tested were: EIGl, Ukmeans, 2KNSC, KNSC, LE and LE-kmeans. This last algorithm applies the k-means algorithm to the matrix of the $k-1$ largest eigenvectors and selects the partition that provides the best modularity for $k = 2,\ldots,n$. The function *leading.eigenvector.community* from *igraph* package was employed to test LE.

Table 1
Comparison between solutions found by LE and LE-kmeans algorithms.

n	Graph	LE		LE-kmeans	
		k	Solution	k	Solution
112	Adjnoun	17	0.2215	5	**0.2665**
62	Dolphins	6	**0.4894**	3	0.4678
198	Jazz	8	0.3529	4	**0.3960**
34	Karate	5	0.3776	5	**0.4062**

Table 1 presents the results obtained by LE and LE-kmeans. The first column reports the number of nodes in each graph, whereas the third to sixth columns present the number of clusters and the solution value of the final partition found by LE and LE-kmeans, respectively. To report the results of the EIGl, Ukmeans, 2KNSC and KNSC, Fig. 1(a) and (b) illustrate the relation between the number of clusters and the normalized solution value. The maximum number of clusters for each graph is given by max{30, n/2}.

It is possible to observe that, in general, the solutions found by k-way outperformed the recursive two-way partitioning algorithms. However, the stability of the solutions when the numbers of clusters is increased did not present a significant difference. The Pearson correlation between the solutions found by EIGl and Ukmeans algorithms, for each graph, had an average of 0.9960. Regarding the 2NSC and KNSC algorithms, an average of 0.9990 was found. Both correlation values indicate a high correlation between the solutions, with a similar behavior when the number of clusters is increased. Regarding the computational time, EIGl and 2NSC presented considerable higher running time than Ukmeans and KNSC, respectively. It is important to mention that the difference got larger when the number of clusters was increased. As LE respects the algorithm proposed by Newman (2006), $r = 0$, its time difference with respect to LE-kmeans was not considerable. Unlike the other recursive two-way partitioning algorithms, it did not spend time searching for the best value for the parameter r.

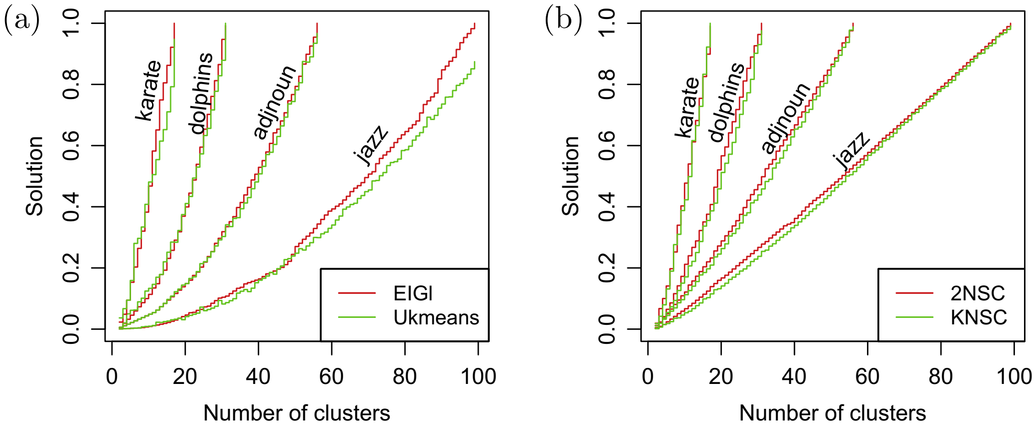

Fig. 1. (a) EIGl versus Ukmeans, (b) 2NSC versus KNSC.

6. Final remarks

This survey was concerned with a particular class of clustering algorithms, known as spectral clustering algorithms. These algorithms have been used to solve distinct graph cut and partitioning problems. They are characterized by their elegant mathematical basis and the production of high quality partitions. Moreover, they present the advantage of giving lower and upper bounds for the studied problems.

The recursive two-way algorithms were the pioneers in spectral clustering algorithms and, along the years, authors have claimed that they are inefficient and unstable. However, their instability was not observed in the experiments performed in this paper. The direct k-way spectral clustering algorithms have become more popular in the last decades, mainly because they give better results for the problems.

In this survey, it was observed that most of the direct k-way partitioning algorithms found in literature uses a correspondence between the vertices from a graph G and the rows from the eigenvectors' matrix of a Laplacian matrix of G. To group the rows of the eigenvectors' matrix, authors have adopted two main strategies: linear ordering (Alpert et al., 1999; Newman, 2006) and the use of the k-means algorithm (Ng et al., 2002; Von Luxburg, 2007). Additionally, few works use other strategies for grouping the matrix of eigenvectors, like, for example, metaheuristics.

It was also observed that there are a few open issues in the area of cluster analysis where spectral theory can provide new insights, such as the choice of the value of k, i.e., the number of clusters. For example, it can be done by choosing the value of k that provides the largest *eigengap*. In an empirical study performed for this survey, some effective results using this approach show how eigenvalues and eigenvectors can significantly contribute to define the cluster structure.

Acknowledgments

The authors acknowledge CAPES, CNPq and FAPESP for their funding support. The authors are also grateful for the valuable referee's suggestions.

References

Alpert, C.J., Kahng, A.B., 1994. Multi-way partitioning via spacefilling curves and dynamic programming. In: Proceeding of the 31st ACM/IEEE Design Automation Conference, pp. 652–657.

Alpert, C., Kahng, A.B., Yao, S.Z., 1999. Spectral partitioning with multiple eigenvectors. Discrete Applied Mathematics 90, 3–26.

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., Mckenney, A., Sorensen, D., 1990. LAPACK: A Portable Linear Algebra Library for High-Performance Computers, Technical Report CS-90-105, Computer Science Department, University of Tennessee, Knoxville, TN.

Bach, F.R., Jordan, M.I., 2004. Learning spectral clustering. In: Thrun, S., Saul, L., Schölkopf, B. (Eds.), Advances in Neural Information Processing Systems, vol. 16. MIT Press, Cambridge, pp. 305–312.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U., 2006. Complex networks: Structure and dynamics. Physics Reports 424, 175–308.

Brand, M., Huang, K., 2003. A unifying theorem for spectral embedding and clustering. In: Bishop, C., Frey, B. (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics.

Chan, P.K., Schlag, M.D.F., Zien, J.Y., 1994. Spectral k-way ratio-cut partitioning and clustering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 13, 1088–1096.

Chung, F.R.K., 1994. Spectral graph theory. In: No. 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society.

Cormack, R.M., 1971. A review of classification. Journal of the Royal Statistical Society 134, 321–367.

Cvetković, D., Doob, M., Sachs, H., 1979. Spectra of Graphs: Theory and Application. Academic Press, New York.

De Bie, T., Cristianini, N., 2006. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. Journal of Machine Learning Research 7, 1409–1436.

Diestel, R., 2005. Graph theory, third ed. Graduate Texts in Mathematics, vol. 173 Springer-Verlag, Heidelberg.

Ding, C., He, X., Zha, H., Gu, M., Simon, H., 2001. A min–max cut algorithm for graph partitioning and data clustering. In: Proceedings of the IEEE International Conference on Data Mining, ICDM 2001, pp. 107–114.

Ding, C., He, X., Meraz, R.F., Holbrook, S.R., 2004. A unified representation of multiprotein complex data for modeling interaction networks. Proteins: Structure, Function, and Bioinformatics 57, 99–108.

Ding, C., He, X., Simon, H.D., 2005. Nonnegative Lagrangian relaxation of k-means and spectral clustering. In: Machine Learning: ECML 2005. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 530–538.

Donath, W., Hoffman, A., 1973. Lower bounds for the partitioning of graphs. IBM Journal Research and Development 17 (5), 420–425.

Fiedler, M., 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Mathematical Journal 25, 619–633.

Filippone, M., Camastra, F., Masulli, F., Rovetta, S., 2008. A survey of kernel and spectral methods for clustering. Pattern Recognition 41, 176–190.

Ford Jr., L.R., Fulkerson, D.R., 1962. Flows in Networks. Princeton University Press, Princeton.

Forman, J.J., Clemons, P.A., Schreiber, S.L., Haggarty, S.J., 2005. SpectralNET – An application for spectral graph analysis and visualization. BMC Bioinformatics 6, 260.

Fowlkes, C., Belongie, S., Chung, F., Malik, J., 2004. Spectral grouping using the Nyström method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 214–225.

Gleiser, P., Danon, L., 2003. Community structure in jazz. Advances in Complex Systems 6, 565–573.

Gu, M., Zha, H., Ding, C., He, X., Simon, H., 2001. Spectral Relaxation Models and Structure Analysis for k-way Graph Clustering and Bi-clustering, Technical Report. CSE-01-007, The Pennsylvania State University.

Hagen, L., Kahng, A., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design 11, 1074–1088.

Hall, K.M., 1970. An r-dimensional quadratic placement algorithm. Management Science 17, 219–229.

Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. Mathematical Programming 79, 191–215.

Hückel, E., 1931. Quantentheoretische Beiträge zum Benzolproblem. Zeitschrift füur Physik a Hadrons and Nuclei 72, 310–337.

Huttenhower, C., Flamholz, A.I., Landis, J.N., Sahi, S., Myers, C.L., Olszewski, K.L., Hibbs, M.A., Siemers, N.O., Troyanskaya, O.G., Coller, H.A., 2007. Nearest neighbor networks: Clustering expression data based on gene neighborhoods. BMC Bioinformatics 8, 250.

Kannan, N., Vishveshwara, S., 1999. Identification of side-chain clusters in protein structures by a graph spectral method. Journal of Molecular Biology 292, 441–464.

Kim, J., Choi, S., 2006. Semidefinite spectral clustering. Pattern Recognition 39 (11), 2025–2035.

Kulis, B., Surendran, A.C., Platt, J.C., 2007. Fast low-rank semidefinite programming for embedding and clustering. In: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics.

Leighton, T., Rao, S., 1988. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In: Annual IEEE Symposium on Foundations of Computer Science, pp. 422–431.

Luo, B., Wilson, R.C., Hancock, E.R., 2003. Spectral embedding of graphs. Pattern Recognition 36, 2213–2230.

Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M., 2003. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology 54, 396–405.

Maier, M., von Luxburg, U., Hein, M., 2009. Influence of graph construction on graph-based clustering measures. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L., Curran, Hook, R. (Eds.), Advances in Neural Information Processing Systems, vol. 21, pp. 1025–1032.

Meilă, M., Shi, J., 2001. A random walks view of spectral segmentation. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics.

Meilă, M., Xu, L., 2004. Multiway Cuts and Spectral Clustering, Technical Report. 442, University of Washington.

Mohar, B., 1991. The Laplacian spectrum of graphs. Graph Theory, Combinatorics, and Applications 2, 871–898.

Newman, M.E.J., 2006. Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74, 036–104.

Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. Physical Review E 69, 026113.

Ng, A., Jordan, M., Weiss, Y., 2002. 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, MA, pp. 849–856.

Saerens, M., Fouss, F., Yen, L., Dupont, P., 2004. The principal components analysis of a graph, and its relationships to spectral clustering. In: Machine Learning: ECML 2004, Lecture Notes in Computer Science, vol. 3201, pp. 371–383.

Schaeffer, S.E., 2007. Graph clustering. Computer Science Review 1, 27–64.

Schrijver, A., 2003. Combinatorial Optimization. Polyhedra and Efficiency, vol. B. Springer-Verlag, Berlin, Heidelberg.

Shi, J., Malik, J., 1997. Normalized cuts and image segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 731–737.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905.

Singh, V., Mukherjee, L., Peng, J., Xu, J., 2008. Ensemble clustering using semidefinite programming. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.), Advances in Neural Information Processing Systems, vol. 20. MIT Press, Cambridge, MA, pp. 1353–1360.

Stewart, G.W., Sun, J., 1990. Matrix Perturbation Theory. Computer Science and Scientific Computing. Academic Press.

Ushioda, A., Kawasaki, J., 1996. Hierarchical clustering of words and application to nlp tasks. In: Ejerhed, E., Dagan, I. (Eds.), Fourth Workshop on Very Large Corpora. Association for Computational Linguistics, Somerset, New Jersey. pp. 28–41.

Venables, W.N., Smith, D.M., 2010. An Introduction to R. R Development Core Team, The R Foundation for Statistical Computing, version 2.11.1.

Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17 (4), 395–416.

Von Luxburg, U., Belkin, M., Bousquet, O., 2008. Consistency of spectral clustering. The Annals of Statistics 36, 555–586.

Wei, Y. C., Cheng, C. K., 1989. Towards efficient hierarchical designs by ratio cut partitioning. In: Proceedings of the IEEE International Conference on Computer-Aided Design. pp. 298–301.

White, S.D.M., Frenk, C.S., 1991. Galaxy formation through hierarchical clustering. Astrophysical Journal 379 (Part 1), 52–79.

Wu, Z., Leahy, R., 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11), 1101–1113.

Xing, E., Jordan, M., 2003. On semidefinite relaxation for normalized *k*-cut and connections to spectral clustering, Technical Report. CSD-03-1265, University of California, Berkeley.

Yu, S.X., Shi, J., 2003. Multiclass spectral clustering. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, pp. 313–319.

Zachary, W., 1977. An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473.

Zelnik-Manor, L., Perona, P., 2004. Self-tuning spectral clustering. Advances in Neural Information Processing Systems, vol. 17. MIT Press, pp. 1601–1608.

Zha, H., Ding, C., Gu, M., He, X., Simon, H., 2002. Spectral relaxation for *k*-means clustering. In: Advances in Neural Information Processing Systems, vol. 14. pp. 1057–1064.