WEIGHTED LAPLACIANS OF GRIDS AND THEIR APPLICATION FOR INSPECTION OF SPECTRAL GRAPH CLUSTERING METHODS

MIECZYSŁAW KŁOPOTEK[1], SŁAWOMIR WIERZCHOŃ[1] AND ROBERT KŁOPOTEK[2]

[1] *Institute of Computer Science, Polish Academy of Sciences ul. Jana Kazimierza 5, 01-248 Warsaw, Poland*

[2] *Faculty of Mathematics and Natural Sciences, The Cardinal Stefan Wyszyński University, ul. Wóycickiego 1/3, 01-938 Warsaw, Poland*

(received: 27 June 2021; revised: 21 July 2021; accepted: 14 August 2021; published online: 30 November 2021)

Abstract: This paper investigates the relationship between various types of spectral clustering methods and their kinship to relaxed versions of graph cut methods. This predominantly analytical study exploits the closed (or nearly closed) form of eigenvalues and eigenvectors of unnormalized (combinatorial), normalized, and random walk Laplacians of multidimensional weighted and unweighted grids. We demonstrate that spectral methods can be compared to (normalized) graph cut clustering only if the cut is performed to minimize the sum of the weight square roots (and not the sum of weights) of the removed edges. We demonstrate also that the spectrogram of the regular grid graph can be derived from the composition of spectrograms of path graphs into which such a graph can be decomposed, only for combinatorial Laplacians. It is impossible to do so both for normalized and random-walk Laplacians. We investigate the in-the-limit behavior of combinatorial and normalized Laplacians demonstrating that the eigenvalues of both Laplacians converge to one another with an increase in the number of nodes while their eigenvectors do not. Lastly, we show that the distribution of eigenvalues is not uniform in the limit, violating a fundamental assumption of the compact spectral clustering method.

Keywords: grid grap, analytical form of graph Laplacians, spectral clustering versus graph cuts

DOI: https://doi.org/10.34808/tq2021/25.3/d

1. Introduction

A considerable number of practical clustering tasks can be viewed as special cases of the general problem of graph clustering, and especially of weighted graph clustering. Graph clustering is understood as finding a cut through the graph removing edges of combined weight as low as possible, with additional restrictions such as balancing the subgraph sizes or volumes. Cut or normalized cut methods may be used for this purpose. With growing graph sizes, the application of cut or normalized cut suffers from a combinatorial explosion, making one interested in approximated and relaxed versions of the problem.

This has awaken interest in spectral clustering methods which exhibit relatively low computational complexity and produce apparently splits of graphs similar to those obtained by the original cut methods. In fact, there have been numerous attempts to align the results of spectral clustering with some of its specific brands, like clustering based on various forms of Laplacians (see e.g. [1]). Von Luxburg [2] presents an overview of these concepts in her tutorial, pointing at prior work demonstrating that the clustering target of spectral clustering is related to various forms of graph-cuts: the ratio cut [3] (when based on combinatorial Laplacians), normalized graph cuts [4], [5] (for normalized Laplacians) and walks [6] (for random walk Laplacians), given that the mentioned fuzzy versions of graph-cuts are considered.

In this paper, we look at these relationships more closely. However, instead of considering the graphs in their generality, we concentrate on weighted regular grids. Grid graphs are of special interest because they can be used for practical implementation of parallel algorithms, as well as in efficient management of wireless sensor networks, [7], [8]. Moreover, such graphs seem to be a useful test bed for investigating the properties of graph clustering algorithms, because such graphs can be constructed with a predefined clear cluster structure. Furthermore, they have the advantage that their spectral properties (eigenvalues and eigenvectors of various Laplacians) can be determined analytically, via closed-form or nearly closed form formulas so that they can be used for an analytical investigation of graph algorithms. In particular, if all edges have the same weight, no cluster structure is present, hence, the clustering algorithms can be tested on whether or not they have a tendency of detecting non-existent clusters. On the other hand, weighted graphs may represent quite a regular set of clusters, and hence, the algorithms can be tested on their capability of detecting such clusters, again in an analytically closed or nearly closed form.

An increasing interest is seen in unweighted and weighted graph Laplacians. They are relevant e.g. for image segmentation [5], indoor space analysis [9], etc. Thus, in the paper we recall the analytical forms of eigenvalues and eigenvectors of combinatorial Laplacians (Sec. 3), signless Laplacians (Sec. 4, sometimes referred to as "unoriented"), normalized Laplacians (Sec. 5) and random-walk Laplacians (Sec. 6) of weighted regular grids. We hope that it would be an interesting research

topic to find also closed form solutions to Laplacians of weighted grid graphs with other weighting schemata than those assumed in this work.

Thereafter, in Section 7, we discuss the implications of these analytical properties for various aspects of spectral cluster analysis. In particular, we investigate:

- the differences between various types of Laplacians, and hence, between various spectral clustering methods that they underpin, (Sec. 7.1);
- whether or not the results of Spectral Clustering match those of graph-cut and normalized graph-cut in case of clustering into two clusters using the so-called Fiedler vector (Sec. 7.2);
- whether or not the results of Spectral Clustering match those of graph-cut and normalized graph-cut in case of clustering into more than two clusters using k-means algorithm (Sec. 7.3);
- we investigate the justifiability of the choice of k eigenvectors for clustering into k-clusters in spectral clustering (Sec. 8.1);
- we investigate whether or not the conclusions drawn may be extended if the weights are not uniform (Sec. 8.2).

2. Notation

A neighborhood matrix S of any graph shall be defined as a matrix with entries $s_{jk} > 0$ if there is a link between nodes j,k, and otherwise it is equal to 0. We assume that $s_{jj} = 0$. s_{jk} is always considered as a weight of the link (edge) between nodes j,k, being deemed as a kind of similarity between the nodes. However, by setting $s_{jj} = 0$, this is not strictly a similarity measure. If $s_{jk} \in 0,1$, we will talk about an unweighted graph, otherwise about a weighted one. An unnormalized (combinatorial) Laplacian L of such a (weighted or unweighted) graph is defined as $L = D - S$, where D is the diagonal matrix with $d_{jj} = \sum_{k=1}^{n} s_{jk}$ for each $j = 1,...n$.

The respective signless Laplacian K of a graph is defined as: $K = D + S$. A normalized Laplacian $\mathfrak{L}$ of a graph is defined as $\mathfrak{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} S D^{-1/2}$, and a random walk Laplacian $\mathbb{L}$ of a graph is $\mathbb{L} = D^{-1} L = I - D^{-1} S$. The matrix $D^{-1} S$ is interpreted as the transition matrix for the random walk on the graph.

Note that while $\mathfrak{L}$ is a symmetric matrix, $\mathbb{L}$ is not symmetric. However, the two matrices are similar as $\mathfrak{L} = D^{1/2} \mathbb{L} D^{-1/2}$. It is easy to verify that if (λ, u) is an eigenpair of $\mathfrak{L}$ then (λ, w) is the eigenpair of $\mathbb{L}$ and $w = D^{-1/2} u$). Alternatively, the eigenpair (λ, w) solves generalized eigen-problem $Lw = \lambda D w$. It should be stressed that while eigenvectors of $\mathfrak{L}$ are orthogonal, these of $\mathbb{L}$ are not.

The eigenvalues of L and K will differ unless we have to do with a bipartite graph which is the case with a grid graph.

A two-dimensional (unweighted) grid graph [10], (called also a square grid graph, or a rectangular grid graph, or $m \times n$ grid) is an $m \times n$ lattice graph $G_{(m,n)}$, meaning the graph Cartesian product $P_m \times P_n$ of path graphs on m and n vertices, respectively. In this paper we go beyond the concept of unweighted grid graphs.

Let us define a weighted generalized grid graph as $G_{(n_1)(\mathfrak{w}_1)}$ being a weighted path graph of n_1 vertices with weight $\mathfrak{w}_1$ for any link in this graph, and the d dimensional weighted grid graph $G_{(n_1,...,n_d)(\mathfrak{w}_1,...,\mathfrak{w}_d)}$ being the weighted graph Cartesian product $G_{(n_1,...,n_{d-1})(\mathfrak{w}_1,...,\mathfrak{w}_{d-1})} \times G_{(n_d)(\mathfrak{w}_d)}$. Thus a d-dimensional weighted grid graph is uniquely defined by a grid graph identity vector pair $[n_1,...,n_d][\mathfrak{w}_1,...,\mathfrak{w}_d]$. We can imagine that the grid graph is embedded into a hyper-cuboid, as in Figure 1 (for two dimensions) where n_j is the number of layers of nodes in the jth dimension and $\mathfrak{w}_j$ is the weight of links between layers in the jth dimension. For example, in Figure 1, we have a two-dimensional hypercuboid (rectangle) with an embedded grid in such a way that there are 3 layers in the horizontal direction and 5 layers of nodes in the vertical direction. The links parallel to the horizontal direction have weights 2, and the links in the vertical direction have weight 7. By the notation 2:7 we will subsequently express the proportion between the weights in the horizontal and vertical directions. In the drawing, higher weights will be expressed by shorter edges.

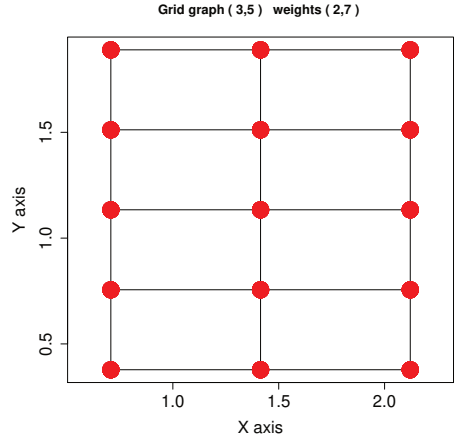

Figure 1. A sample two-dimensional grid graph

Let us recall a special way of assigning (integer) identities to the weighted grid graph $G_{(n_1,...,n_d)(\mathfrak{w}_1,...,\mathfrak{w}_d)}$ nodes, following the ideas of [11, 12]. The identity numbers run consecutively from 1 to $\prod_{j=1}^{d} n_j$. Each node identity number i is uniquely associated with a identity vector $\mathbf{x} = [x_1,...,x_d]$ via the (invertible) formula:

$$i = 1 + \sum_{j=1}^{d} (x_j - 1) \cdot \prod_{k=j+1}^{d} n_k, \tag{1}$$

Let $i(i)$ be a function turning the node identity number i to the corresponding identity vector $\mathbf{x}$.[1]

A node with identity vector $[x_1,...,x_d]$ is connected for each j with the node $[x_1,...,x_j-1,x_d]$ if $x_j>1$ and with node $[x_1,...,x_j+1,x_d]$ if $x_j<n_j$ with weight $\mathfrak{w}_j$ and there are no other edges in the graph.

We will index the eigenvalues and the corresponding eigenvectors with an identity vector of d integers $\mathbf{z}=[z_1,...,z_d]$. Note that increasing/decreasing a z_j by $2n_j$ will leave any eigenvalue and eigenvector unchanged. Also replacing z_j with $-z_j$ (occasionally together with replacing the corresponding shift δ with $-\delta$ to be explained later) will leave the eigenvalue and the eigenvector unchanged. Accordingly, the value range of z_j can be smoothly reduced to the range $[0,n_j]$. We are subsequently interested only in the range $[-n_j+1,2n_j-1]$ for z_j. In case that eigenvalues/eigenvectors depend on the weights $\mathfrak{w}$ we will also use the weights when identifying them.

The similarity matrix S of the weighted grid graph $G_{(n_1,...,n_d)(\mathfrak{w}_1,...,\mathfrak{w}_d)}$ is a $(\prod_{j=1}^{d} n_j)\times(\prod_{j=1}^{d} n_j)$ matrix with $s_{il}=\mathfrak{w}_j$ if nodes with identities i,l are connected and their connection is in dimension j and $s_{il}=0$ otherwise.

3. Combinatorial Laplacians of Weighted Grid Graphs

Combinatorial Laplacians of weighted grid graphs are easily derived from path graph Laplacians via combination proposed e.g. by Fiedler [13], and others, [14, 15]. The form of eigenvectors is identical as in case of unweighted graphs, while the eigenvalues differ and are susceptible to scale (they increase when the weights of edges are proportionally increased). Let us define

$$\lambda_{[z_1,...,z_d]}=\sum_{j=1}^{d} 2\mathfrak{w}_j\cdot\left(1-\cos\left(\frac{\pi z_j}{n_j}\right)\right), \tag{2}$$

where for each $j=1,...,d$ z_j is an integer such that $0\le z_j\le n_j-1$.
Let us furthermore define

$$\nu_{[z_1,...,z_d],[x_1,...,x_d]}=\prod_{j=1}^{d}\cos\left(\frac{\pi z_j}{n_j}\left(x_j-0.5\right)\right), \tag{3}$$

where for each $j=1,...,d$ x_j is an integer such that $1\le x_j\le n_j$.
And finally let us define the n dimensional vector $v_{[z_1,...,z_d]}$ such that

$$v_{[z_1,...,z_d],i}=\nu_{[z_1,...,z_d],[x_1,...,x_d]}, \tag{4}$$

Note at this place that we are talking about two distinct multidimensional spaces. The first one is the space in which we locate the drawing of the grid, like

1. The identity number of the node [4] in a graph $G_{(7)(...)}$ is computed as (4-1)*1+1=4. The identity number of the node [2,4] in a graph $G_{(7,6)(...)}$ is computed as (2-1)*6+(4-1)*1+1=10. The identity number of the node [2,4,5] in a graph $G_{(7,6,8)(...)}$ is computed as $(2-1)*48+(4-1)*8+(5-1)*1+1=77$.

the two-dimensional space in Figure 1. If a confusion may occur anywhere, we will refer to the grid space, the grid dimension, etc. The grid space is d-dimensional. When performing spectral analysis, each node is placed in a different space, called hereafter the spectral space. This space is n-dimensional, where $n = \prod_{j=1}^{d} n_j$. We will refer to spectral space and spectral dimensions in this space. So if a node has coordinates $[x_1,...,x_d]$ in the grid space, that it belongs to the layer x_1 in the grid dimension 1, ..., x_d in the grid dimension d, then in the spectral space it will have the coordinate $\nu_{[0,...,0],[x_1,...,x_d]}$ in the spectral dimension $[0,...,0]$, $\nu_{[1,...,0],[x_1,...,x_d]}$ in the spectral dimension $[1,...,0]$, $...\nu_{[n_1-1,...,0],[x_1,...,x_d]}$ in the spectral dimension $[n_1-1,...,0]$, $...\nu_{[n_1-1,...,n_d-1],[x_1,...,x_d]}$ in the spectral dimension $[n_1-1,...,n_d-1]$. Nevertheless, the spectral analysis does not make use of all these dimensions and usually restricts itself to some k spectral dimensions associated with k lowest eigenvalues.

Theorem 1 Given the combinatorial Laplacian L of the weighted grid graph $G_{(n_1,...,n_d)}(\mathfrak{w}_1,...,\mathfrak{w}_d)$, for each vector of integers $[z_1,...,z_d]$ such that for each $j=1,...,d$, $0 \le z_j \le n_j - 1$, the $\lambda_{[z_1,...,z_d]}$, as defined by (2), is an eigenvalue.[2] of L and $v_{[z_1,...,z_d]}$, as defined by (4) is a corresponding eigenvector.[3]

This theorem is proven in [16]. The proof is rather technical and straightforward based on the paper of [13] and Fiedler's results on the path graph combination.

4. Unoriented Laplacian of a Weighted Grid Graph

Like in the case of unweighted grid graphs, there exists an elegant solution to the eigen-problem of the unoriented Laplacian defined as $K = D + S$, [17].

Theorem 2 The unoriented Laplacian eigenvalues[4] for a weighted grid graph are of the same form as for the combinatorial Laplacian that is

$$\lambda_{[z_1,...,z_d]} = \sum_{j=1}^{d} \mathfrak{w}_j \left(2\sin\left(\frac{\pi z_j}{2n_j} \right) \right)^2, \tag{5}$$

2. For instance, the combinatorial Laplacian eigenvalue $\lambda_{[3,4]}$ in a graph $G_{(5,6)(0.7,2.8)}$ is computed as $2\cdot(1-\cos(\pi\cdot 3/5))\cdot 0.7 + 2\cdot(1-\cos(\pi\cdot 4/6))\cdot 2.8 = 10.2326$. Similarly the combinatorial Laplacian eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(0.7,2.8)}$ is computed as $2\cdot(1-\cos(\pi\cdot 0/2))\cdot 0.7 + 2\cdot(1-\cos(\pi\cdot 1/3))\cdot 2.8 = 2.8$.

3. The Combinatorial Laplacian eigenvector corresponding to the eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(...)}$ is computed as $[\cos(\frac{1-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{1-0.5}{3}\cdot 1\cdot\pi), \cos(\frac{1-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{2-0.5}{3}\cdot 1\cdot\pi), \cos(\frac{1-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{3-0.5}{3}\cdot 1\cdot\pi), \cos(\frac{2-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{1-0.5}{3}\cdot 1\cdot\pi), \cos(\frac{2-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{2-0.5}{3}\cdot 1\cdot\pi), \cos(\frac{2-0.5}{2}\cdot 0\cdot\pi)\cdot\cos(\frac{3-0.5}{3}\cdot 1\cdot\pi)]$.

4. The Signless Laplacian eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(0.7,2.8)}$ is computed as $2\cdot(1-\cos(\pi\cdot 0/2))\cdot 0.7 + 2\cdot(1-\cos(\pi\cdot 1/3))\cdot 2.8 = 2.8$.

The corresponding eigenvectors[5] have components of the form

$$\nu_{[z_1,...,z_d],[x_1,...,x_d]} = \prod_{j=1}^{d} (-1)^{x_j} \cos\left(\frac{\pi z_j}{n_j}\left(x_j - 0.5\right)\right), \tag{6}$$

The proof is a variation on the proof of theorem 1.

5. Normalized Laplacians of Weighted Grid Graphs

The approach to the eigen-problem of a normalized Laplacian differs strongly from the approach used in the previous sections. The principal difference is that the path combination of Fiedler does not work due to the normalization.

The solution does not have a completely closed-form. An iterative component is needed when identifying an eigenvalue. Once the eigenvalue is identified, the so-called *shifts* or δ's are also identified and then the eigenvalue and eigenvectors are in a closed form with respect to these shifts δ. The problem of only a partial closed-from is strongly related to the fact that the eigen-problem for the normalized Laplacian cannot be decomposed in a way that could be done for the combinatorial Laplacians.

Normalization causes that the eigenvectors of weighted grid graph normalized Laplacians, contrary to their combinatorial counterparts, depend also on weights because the respective eigenvalues depend on them.

As in the previous sections, we shall index the eigenvalues and eigenvectors with the vector $\mathbf{z} = [z_1, ..., z_d]$ such that $0 \leq z_j < n_j$ for $j = 1, ..., d$.

Theorem 3 The normalized Laplacian $\mathfrak{L}$ of a d-dimensional weighted grid graph has the eigenvalues

$$\lambda_z = 1 + \sum_{j=1}^{d} \frac{\mathfrak{w}_j}{\sum_{j=1}^{d} \mathfrak{w}_j} \cos\left(\frac{1}{n_j - 1}\left(z_j \pi - 2\delta_j^z\right)\right), \tag{7}$$

with the δ^z vector, called shift vector, defined as a solution of the equation system consisting of the subsequent equation (10) and the equations (11) for each $l = 1, ..., d$.

---

5. The Signless Laplacian eigenvector corresponding to the eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(...)}$ is computed as $[\cos(\frac{1-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{1-0.5}{3} \cdot 1 \cdot \pi), -\cos(\frac{1-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{2-0.5}{3} \cdot 1 \cdot \pi), \cos(\frac{1-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{3-0.5}{3} \cdot 1 \cdot \pi), -\cos(\frac{2-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{1-0.5}{3} \cdot 1 \cdot \pi), \cos(\frac{2-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{2-0.5}{3} \cdot 1 \cdot \pi), -\cos(\frac{2-0.5}{2} \cdot 0 \cdot \pi) \cdot \cos(\frac{3-0.5}{3} \cdot 1 \cdot \pi)]$.

The corresponding eigenvectors[6] v_z have components of the form

$$\nu_{z,[x_1,\ldots,x_d]} = D^{1/2}_{[x_1,\ldots,x_d],[x_1,\ldots,x_d]} \tag{8}$$

$$\prod_{j=1}^{d}(-1)^{x_j}\cos\left(\frac{x_j-1}{n_j-1}\left(z_j\pi-2\delta_j^z\right)+\delta_j^z\right) \tag{9}$$

Proof Notwithstanding the fact that this theorem may bear a resemblance to the theorem 1, it is different in nature. The known results of [13] on the path graph products do not apply because of the nature of the normalized Laplacian. The proof relies on distinguishing the inner and border nodes of the grid and showing that the formulas are valid at the border points and inductively hold for inner points too. A detailed proof can be found in [16]. *Q.E.D.*

The defining equations for δ (shifts) are:

$$2\mathfrak{w}_\Sigma\lambda_z = \sum_{j=1}^{d}\mathfrak{w}_j\left(2+2\cos\left(\frac{1}{n_j-1}\left(z_j\pi-2\delta_j^z\right)\right)\right), \tag{10}$$

$$\lambda_z = 1+\cos\left(\frac{1}{n_l-1}\left(z_l\pi-2\delta_l^z\right)\right)+\tan(\delta_l^z)\sin\left(\frac{1}{n_l-1}\left(z_l\pi-2\delta_l^z\right)\right), \tag{11}$$

Interestingly, the last equation (11) does not depend explicitly on weights. Hence, it is formally identical with the very same equation for unweighted graphs. Nonetheless, it should be kept in mind that λ_z depends on the weights and therefore the impact of weighting is present also in this equation. The equation (10), after dividing by $2\mathfrak{w}_\Sigma$ reduces to the formula (7).

By combining the equation (10) with equations (11) for each l we get a system of $d+1$ equations from which λ and δ's can be determined. The equation (11) may be transformed to:

$$\begin{aligned}(\lambda_z-1)\cos(\delta_l^z) = \cos(\delta_l^z)\cos\left(\frac{1}{n_l-1}\left(z_l\pi-2\delta_l^z\right)\right)\\ +\sin(\delta_l^z)\sin\left(\frac{1}{n_l-1}\left(z_l\pi-2\delta_l^z\right)\right),\end{aligned} \tag{12}$$

that is

$$(\lambda_z-1)\cos(\delta_l^z) = \cos\left(\delta_l^z-\frac{1}{n_l-1}\left(z_l\pi-2\delta_l^z\right)\right), \tag{13}$$

which is simpler to solve for δ knowing λ. The solution can be obtained using the bisectional method on λ using the above formula to obtain δs, and using (7) to

6. The Normalized Laplacian eigenvector corresponding to the eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(0.7,2.8)}$ is computed as $[\cos(\frac{1-1}{2-1}\cdot(0\cdot\pi-2\delta_1^z)+\delta_1^z)\cdot\cos(\frac{1-1}{3-1}\cdot(1\cdot\pi-2\delta_1^z)+\delta_1^z)\sqrt{3.5},-\cos(\frac{1-1}{2-1}\cdot(0\cdot\pi-2\delta_2^z)+\delta_2^z)\cdot\cos(\frac{2-1}{3-1}\cdot(1\cdot\pi-2\delta_2^z)+\delta_2^z)\sqrt{6.3},\cos(\frac{1-1}{2-1}\cdot(0\cdot\pi-2\delta_3^z)+\delta_3^z)\cdot\cos(\frac{3-1}{3-1}\cdot(1\cdot\pi-2\delta_3^z)+\delta_3^z)\sqrt{3.5}-\cos(\frac{2-1}{2-1}\cdot(0\cdot\pi-2\delta_4^z)+\delta_4^z)\cdot\cos(\frac{1-1}{3-1}\cdot(1\cdot\pi-2\delta_4^z)+\delta_4^z)\sqrt{3.5},\cos(\frac{2-1}{2-1}\cdot(0\cdot\pi-2\delta_5^z)+\delta_5^z)\cdot\cos(\frac{2-1}{3-1}\cdot(1\cdot\pi-2\delta_5^z)+\delta_5^z)\sqrt{6.3},-\cos(\frac{2-1}{2-1}\cdot(0\cdot\pi-2\delta_6^z)+\delta_6^z)\cdot\cos(\frac{3-1}{3-1}\cdot(1\cdot\pi-2\delta_6^z)+\delta_6^z)\sqrt{3.5}]$ where δ^z is a vector of the form $[-0.4636,0.1002]$.

get the value of λ' and then reducing bisectionally the difference between λ and λ' down to zero.

6. Random Walk Laplacians of Weighted Grid Graph

As already mentioned in Section 2, the eigenvalues and eigenvectors for Random Walk Laplacians could be conveniently derived from those for Normalized Laplacians. Thus, as proved in [16]

Theorem 4 The random walk Laplacian $\mathbb{L}$ of a weighted d-dimensional grid with at least one inner node, has the eigenvalues[7] of the form

$$\lambda_z = 1 + \sum_{j=1}^{d} \frac{\mathfrak{w}_j}{\mathfrak{w}_\Sigma} \cos\left(\frac{1}{n_j - 1}\left(z_j\pi - 2\delta_j\right)\right), \tag{14}$$

with the δ^z vector defined as a solution of the equation system consisting of the preceding equation (10) and the equations (11) for each $l = 1,...,d$.

The corresponding eigenvectors[8] v_z have components of the form

$$\nu_{z,[x_1,...,x_d]} = D_{[x_1,...,x_d],[x_1,...,x_d]} \times \prod_{j=1}^{d} (-1)^{x_j} \cos\left(\frac{x_j - 1}{n_j - 1}\left(z_j\pi - 2\delta_j^z\right) + \delta_j^z\right), \tag{15}$$

7. Implications for Spectral Clustering

Spectral clustering encompasses the algorithms that cluster points using eigenvectors of matrices derived from the data, see e.g. [1] for a deeper discussion.

7.1. Relations between Laplacian types

It is also worth having a look at the comparison of aligned eigenvalues of combinatorial and normalized Laplacians, as visible in Figure 2. Though they appear to be nearly placed on a straight line, they are not, they lie above it in the middle.

In Figure 3, one eigenvector for combinatorial and normalized Laplacians is compared for each weighting of edges. They exhibit similar patterns, with weights being responsible for some spreading of the values.

Figure 4 illustrates the relationship between eigenvalues and the shifts of normalized Laplacians in grid graphs. This relationship seems not to be simplistic

---

7. The Random Walk Laplacian eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(0.7,2.8)}$ is computed as $1 + \frac{0.7}{3.5}\cos(\frac{1}{2-1}(\pi \cdot 0 - 2\delta_1^z)) + \frac{2.8}{3.5}\cos(\frac{1}{3-1}(\pi \cdot 1 - 2\delta_2^z)) = 1.2$ where δ^z is a vector of the form [-0.4636,0.1002] .

8. The Random Walk Laplacian eigenvector corresponding to the eigenvalue $\lambda_{[0,1]}$ in a graph $G_{(2,3)(0.7,2.8)}$ is computed as $[\cos(\frac{1-1}{2-1}\cdot(0\cdot\pi - 2\delta_1^z) + \delta_1^z)\cdot\cos(\frac{1-1}{3-1}\cdot(1\cdot\pi - 2\delta_1^z) + \delta_1^z)\cdot 3.5, -\cos(\frac{1-1}{2-1}\cdot(0\cdot\pi - 2\delta_2^z) + \delta_2^z)\cdot\cos(\frac{2-1}{3-1}\cdot(1\cdot\pi - 2\delta_2^z) + \delta_2^z)\cdot 6.3, \cos(\frac{1-1}{2-1}\cdot(0\cdot\pi - 2\delta_3^z) + \delta_3^z)\cdot\cos(\frac{3-1}{3-1}\cdot(1\cdot\pi - 2\delta_3^z) + \delta_3^z)\cdot 3.5, -\cos(\frac{2-1}{2-1}\cdot(0\cdot\pi - 2\delta_4^z) + \delta_4^z)\cdot\cos(\frac{1-1}{3-1}\cdot(1\cdot\pi - 2\delta_4^z) + \delta_4^z)\cdot 3.5, \cos(\frac{2-1}{2-1}\cdot(0\cdot\pi - 2\delta_5^z) + \delta_5^z)\cdot\cos(\frac{2-1}{3-1}\cdot(1\cdot\pi - 2\delta_5^z) + \delta_5^z)\cdot 6.3, -\cos(\frac{2-1}{2-1}\cdot(0\cdot\pi - 2\delta_6^z) + \delta_6^z)\cdot\cos(\frac{3-1}{3-1}\cdot(1\cdot\pi - 2\delta_6^z) + \delta_6^z)\cdot 3.5]$ where δ^z is a vector of the form $[-0.4636, 0.1002]$.

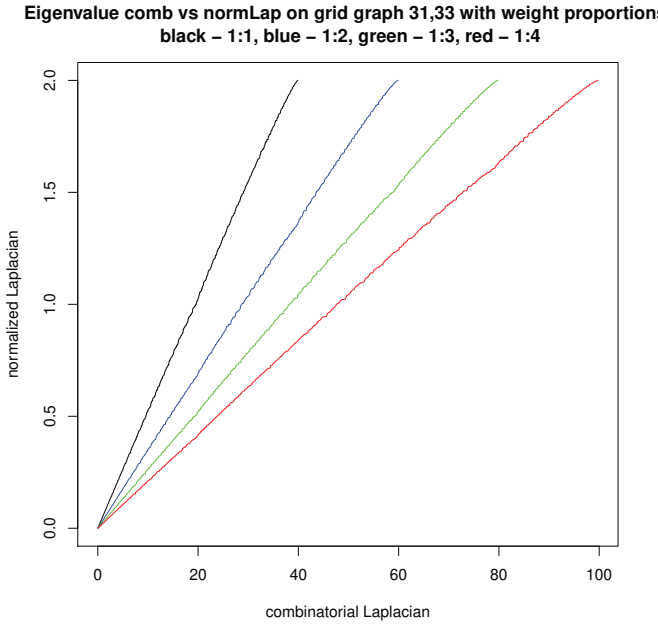

Figure 2. A comparison of eigenvalue distributions between combinatorial and normalized Laplacians, for structurally the same grid graph, but with various proportions of weights in both directions. The colors indicate: black - 1:1 (the unweighted case), blue - 1:2, green - 1:3, red - 1:4.A sample two-dimensional grid graph

and may at least partially explain why we did not find a closed-form solution for identifying eigenvalues and shifts. Nevertheless, we can see that the patterns are similar for various proportions of weights of edges of the grid graph.

Finally, we shall pose the question how the cumulative distribution of eigenvalues of a normalized Laplacian of a grid graph would look like in the limit (when the number of nodes grows). Let us consider the unweighted case. If we keep in mind that $|\delta_j| < \pi$, then for sufficiently high n_j and z_j the contribution of δ_j in the equation (7) will vanish and

$$\lambda_z \approx 1 + \frac{1}{d}\sum_{j=1}^{d} \cos\left(\frac{z_j \pi}{n_j}\right) = 1 + \frac{1}{d}\sum_{j=1}^{d}\left(1 - 2\sin^2\left(\frac{z_j\pi}{2n_j}\right)\right) =$$

$$2 - 2\frac{1}{d}\sum_{j=1}^{d}\sin^2\left(\frac{z_j\pi}{2n_j}\right), \tag{16}$$

which, up to a scaling factor, resembles the defining equation of combinatorial Laplacian eigenvalue (2). This means that the in-the-limit behavior of normalized Laplacian eigenvalues will resemble that of combinatorial Laplacian eigenvalues, i.e., uniformity can be assumed.

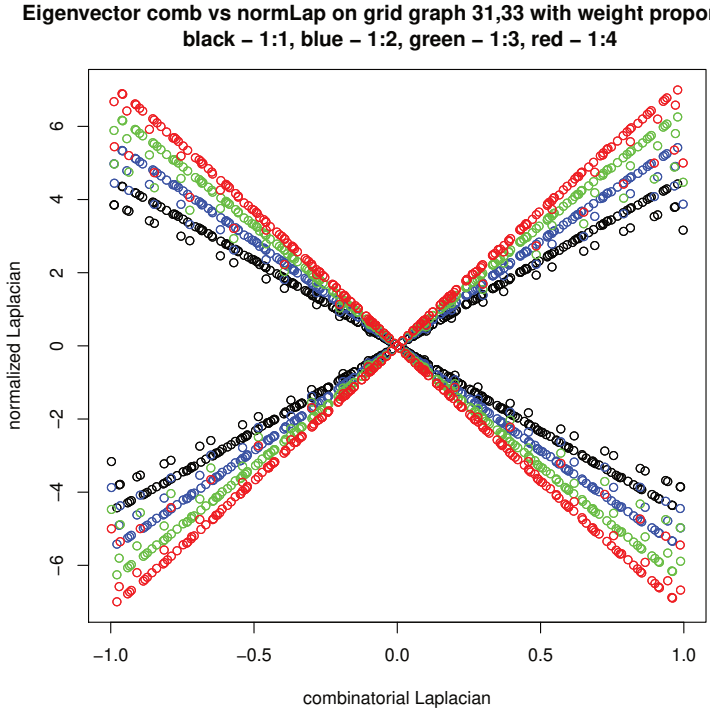

Figure 3. A comparison of sample eigenvectors of combinatorial and normalized Laplacians of weighted two-dimensional grid graphs of approximately 1,000 nodes for $\mathbf{z}=[1,1]$ with various proportions of weights in both directions The colors indicate: black - 1:1 (the unweighted case), blue - 1:2, green - 1:3, red - 1:4.

Let us consider also the "within-the-limit" behavior of the normalized Laplacian eigenvectors, as described by the expression (9). For simplicity, let us set the weights to 1. For sufficiently large $\mathbf{z}$ and $\mathbf{x}$

$$\nu_{\mathbf{z},\mathbf{x}} = D_{\mathbf{x},\mathbf{x}}^{1/2} \prod_{j=1}^{d} (-1)^{x_j} \cos\left(\frac{x_j - 1}{n_j - 1} \left(z_j \pi - 2\delta_j^z \right) + \delta_j^z \right) \approx$$

$$D_{\mathbf{x},\mathbf{x}}^{1/2} \prod_{j=1}^{d} (-1)^{x_j} \cos\left(\frac{z_j \pi}{n_j} \left(x_j - 1 \right) + \delta_j^z \right), \tag{17}$$

which nevertheless differs from the combinatorial Laplacian eigenvector components (even in an unweighted case) (3),

$$\nu_{\mathbf{z},\mathbf{x}} = \prod_{j=1}^{d} \cos\left(\frac{\pi z_j}{n_j} \left(x_j - 0.5 \right) \right), \tag{18}$$

in terms of the shift.

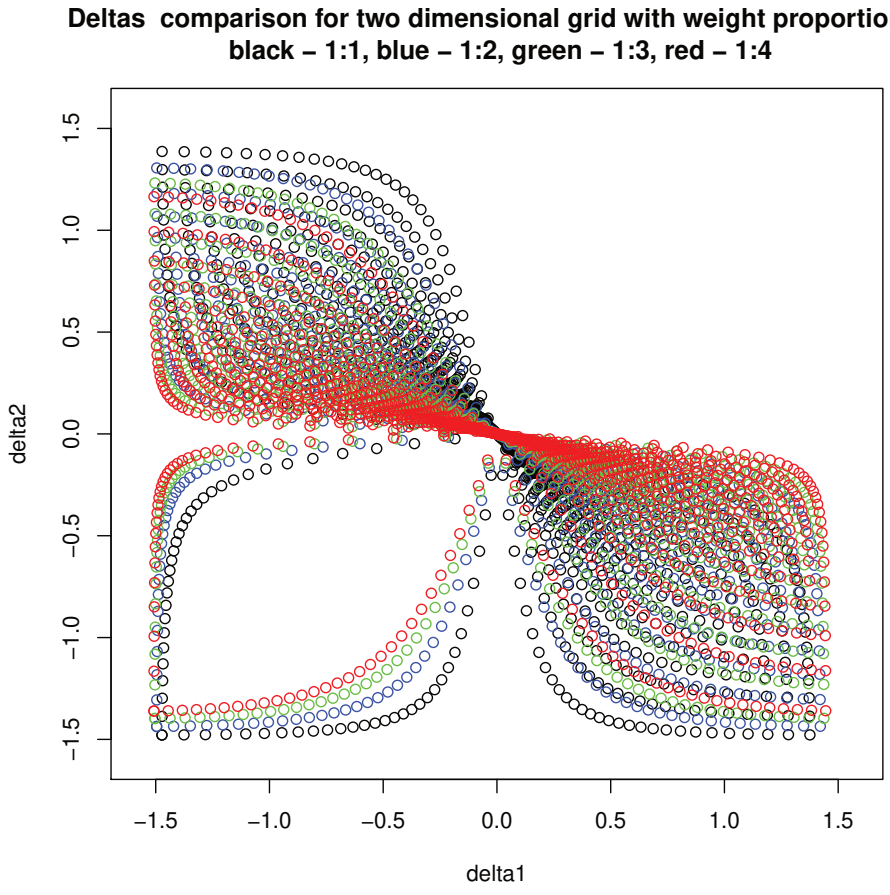

Figure 4. Relationships of normalized Laplacian eigenvalues λ and shifts δ of grid graphs of approximately 1,000 nodes with various proportions of weights in both directions. The colors indicate: black - 1:1 (the unweighted case), blue - 1:2, green - 1:3, red - 1:4.

7.2. Spectral Clustering versus Cut and Ncut clustering methods

Frequently, the Spectral Clustering is represented as a kind of relaxation of graph cuts. A graph cut means removal of some edges in order to obtain a disconnected graph. Graph clustering into two clusters may be deemed as a graph cut task in which the total weight of removed edges is minimized, or alternatively, to obtain balanced subgraphs, the cut value is normalized via the sum of reciprocals of subgraph volumes (the latter is called a normalized cut). Both the cut and the normalized cut are claimed to be related to the second smallest eigenvalue eigenvector (Fiedler vector) of a combinatorial Laplacian or a normalized Laplacian, respectively (if clustered into two clusters).

The argument goes as follows. Let f be an indicator vector telling whether a node belongs to cluster 1 or 2: if a node i belongs to cluster 1, then $f_i = 1$, and

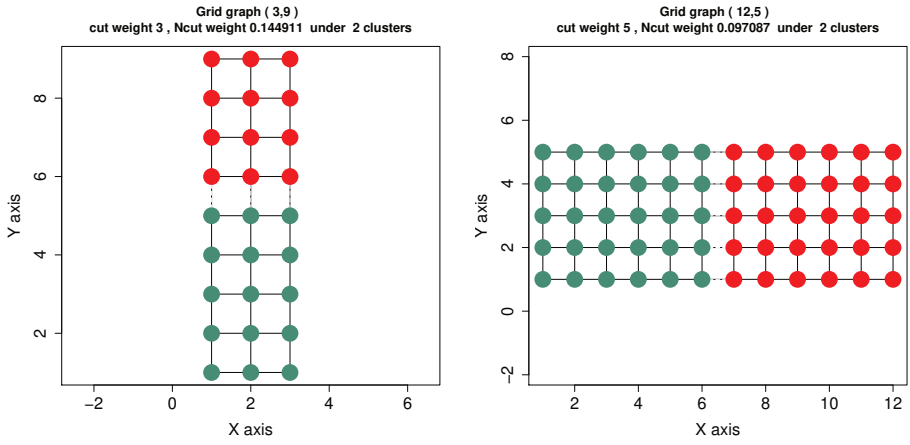

Figure 5. The plots of clustering of unweighted two-dimensional grid graphs

if a node i belongs to cluster 2, then $f_i = -1$. Hence, if graph nodes are split into sets $B, \overline{B}$, then the cut can be expressed as

$$cut(B,\overline{B}) = \sum_{i\in B}\sum_{j\in\overline{B}} \mathfrak{w}_{ij}\frac{(f_i - f_j)^2}{4}, \tag{19}$$

It turns out that, for the Laplacian L of the connectivity matrix of such a graph,

$$f^T L f = 2\sum_{i\in B}\sum_{j\in\overline{B}} \mathfrak{w}_{ij}(f_i - f_j)^2, \tag{20}$$

The condition that $f_i \in \{-1,1\}$ is relaxed and instead it is allowed that $f_i \in [-1,1]$ so that we can speak about a kind of fuzzy membership f which may be defuzzified later. Under such circumstances and by imposing the condition that both clusters must be non-empty, and imposing the additional constraint that the scale of f should not matter, minimizing the "cut"

When we apply this procedure to an unweighted grid graph, then it is obvious from the derived analytical formulas that the cut will run along the dimension for which the number of nodes is the biggest, see Figure 5.

However, in the case of weighted grid graphs, the proportions between weights of edges start to play a role. See Figure 6.

We observe some disturbing behavior. Higher edge weights lead to lower weight cuts! More specifically, in the left part of Figure 6 cutting vertically would lead to a cut weight of 27, while the Fiedler vector chooses a cut of 60, if we perform the split into nodes with positive and negative values of this eigenvector (following initial Fiedler proposal). Not to say that cutting out a corner point would yield a cut weight of 23.

This means that the spectral clustering performs a different clustering from the one that is claimed in the literature that is the optimization of *cut*.

In order to understand the issue, let us consider the analytical forms of grid graph eigenvalues and vectors.

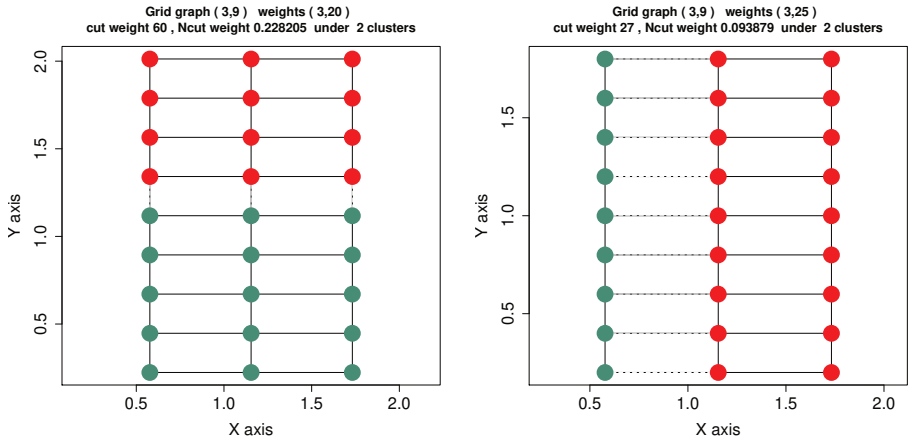

Figure 6. The plots of clustering of weighted two-dimensional grid graphs. Edges are omitted as their placement is obvious. The smaller the distances, the higher the weights

Let us look at the precise formula for the Fiedler eigenvalue. Let this eigenvalue be in the dimension d_1.

$$\lambda_{[0,\ldots,1_{d_1},\ldots,0]} = 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right) \tag{21}$$

Consider a competing dimension d_0 in which we would like to have the Fieldler eigenvalue.

$$\lambda_{[0,\ldots,1_{d_0},\ldots,0]} = 2\mathfrak{w}_{d_0} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_0}}\right)\right), \tag{22}$$

We can achieve the effect that $\lambda_{[0,\ldots,1_{d_0},\ldots,0]} < \lambda_{[0,\ldots,1_{d_1},\ldots,0]}$ in two different ways. Either we increase n_{d_0} or increase $\mathfrak{w}_{d_0}$. While the former is fixed, let us look what we can do with the latter.

$$2\mathfrak{w}_{d_0} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right), \tag{23}$$

$$2\mathfrak{w}_{d_0} \cdot \left(2\sin^2\left(\frac{\pi/2}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(2\sin^2\left(\frac{\pi/2}{n_{d_1}}\right)\right), \tag{24}$$

$$\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}} < \frac{\sin^2\left(\frac{\pi/2}{n_{d_1}}\right)}{\sin^2\left(\frac{\pi/2}{n_{d_0}}\right)}, \tag{25}$$

If both n_{d_0} and n_{d_1} are large,

$$\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}} < \frac{\frac{\pi^2/4}{n_{d_1}^2}}{\frac{\pi^2/4}{n_{d_0}^2}} \Rightarrow \sqrt{\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}}} < \frac{n_{d_0}}{n_{d_1}}, \tag{26}$$

This means that the spectral clustering optimizes the sum of square roots of edge weights

$$srcut(B,\overline{B}) = \sum_{i\in B}\sum_{j\in\overline{B}} \sqrt{\mathfrak{w}_{ij}}\frac{(f_i - f_j)^2}{4} \tag{27}$$

and not the sum of cut edges for large grids, as suggested by (19) (There are some subtleties for smaller ones).

This obviously provides only a partial explanation because not cutting out a single corner point is not explained. We can speculate that the distance from the "cluster center" plays a role in calculating the edge weight.

One way to deal with the cutting of small node groups is to make a correction for the volumes of the clusters (sums of weights of edges coinciding with the nodes of a cluster)

$$Ncut(B,\overline{B}) = cut(B,\overline{B})\left(\frac{1}{vol(B)} + \frac{1}{vol(\overline{B})}\right), \tag{28}$$

This is also transformed to a spectral clustering task with the following argument, [5]. Let f be an indicator vector telling whether a node belongs to cluster 1 (B) or 2 ($\overline{B}$): if a node i belongs to cluster 1, then $f_i =$

$$f^T L f = 2\sum_{i\in B}\sum_{j\in\overline{B}} \mathfrak{w}_{ij}(f_i - f_j)^2, \tag{29}$$

$$= 2\sum_{i\in B}\sum_{j\in\overline{B}} \mathfrak{w}_{ij}\left(\frac{1}{vol(B)} + \frac{1}{vol(\overline{B})}\right)^2, \tag{30}$$

On the other hand

$$f^T D f = \sum_i d_{ii} f_i^2 = \sum_{i\in B}\frac{d_i}{vol(B)^2} + \sum_{i\in\overline{B}}\frac{d_i}{vol(\overline{B})^2} =$$

$$= \frac{vol(B)}{vol(B)^2} + \frac{vol(\overline{B})}{val(\overline{B})^2} = \frac{1}{vol(B)} + \frac{1}{vol(\overline{B})}, \tag{31}$$

This means that $Ncut$ is proportional to

$$\frac{f^T L f}{f^T D f} = \frac{g^T D^{-1/2} L D^{-1/2} g}{g^T g} = \frac{g^T \mathfrak{L} g}{g^T g}, \tag{32}$$

The conditions imposed on g are relaxed so that we can speak again about a kind of fuzzy membership g which may be defuzzified later. Under such circumstances and by imposing the condition that both clusters must be non-empty, and imposing the additional constraint that the scale of g should not matter, minimizing the "cut"

By applying this procedure to grid graphs, we obtain clusterings identical with those in the combinatorial Laplacian examples. Due to the shifts, there are some differences when using concrete examples close to those with the change of the cut dimension. Nonetheless, we can show by a similar argument that the square roots of the weights matter in minimizing the normalized cut and not the weights themselves. Let us recall that from the formula (7) we have

$$\lambda_{[n_1-1,\ldots,n_{d_0}-2,\ldots,n_d-1]} = 1 + \frac{\mathfrak{w}_{d_0}}{\sum_{j=1}^{d} \mathfrak{w}_j} \cos\left(\pi\left(1 - \frac{1}{n_{d_0}-1}\right) - \frac{2\delta_j}{n_j - 1}\right), \tag{33}$$

$$+ \sum_{j=1, j\neq d_0}^{d} \frac{\mathfrak{w}_j}{\sum_{j=1}^{d} \mathfrak{w}_j} \cos\left(\pi - \frac{2\delta_j}{n_j - 1}\right), \tag{34}$$

The above formula approaches the formula (22) for large values of n_j in all the dimensions up to a scaling factor (the sum of weights). Hence, the argument can be repeated.

7.3. Beyond Fiedler vector – multiple classes

The spectral clustering theory recommends to proceed as follows: If a split in more than 2 clusters is to be produced, we take k eigenvectors corresponding to k smallest eigenvalues of the respective Laplacian (normalized or combinatorial), form a matrix with columns being these eigenvectors and cluster rows of this matrix using k-means. The cluster assignment of such data means the cluster assignment for graph nodes. Figure 7 presents such clusterings of a 10 by 10 grid into ten clusters.

If the graph is to be split along one dimension, a modified approach needs to be used to determine the weights.

In the dimension d_0, we want to have n_{d_0} clusters. Therefore, the maximum eigenvalue along this dimension

$$\lambda_{[0,\ldots,(n_{d_0}-1)_{d_0},\ldots,0]} = 2\mathfrak{w}_{d_0} \cdot \left(1 - \cos\left(\frac{\pi(n_{d_0}-1)}{n_{d_0}}\right)\right), \tag{35}$$

needs to be smaller than the smallest one in any other dimension d_1.

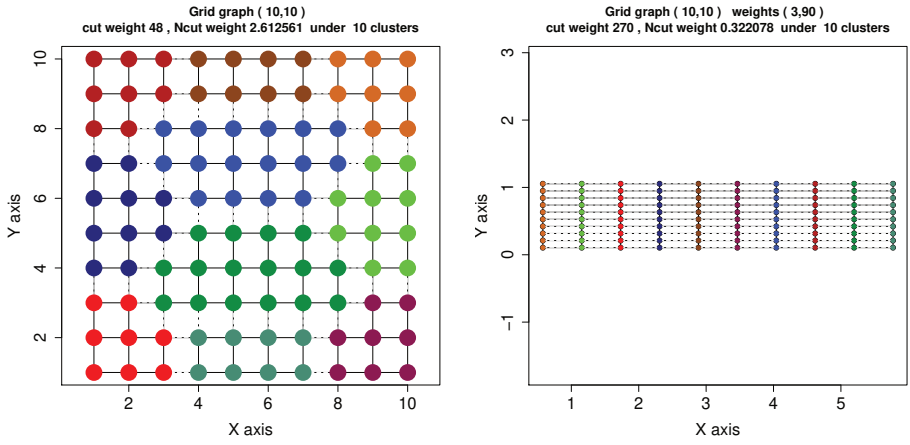

Figure 7. The plots of clustering of unweighted and weighted two-dimensional grid graphs. The smaller the distances, the higher the weights

$$\lambda_{[0,\ldots,1_{d_1},\ldots,0]} = 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right), \tag{36}$$

We can achieve the effect that

$$\lambda_{[0,\ldots,(n_{d_0}-1)_{d_0},\ldots,0]} < \lambda_{[0,\ldots,1_{d_1},\ldots,0]}, \tag{37}$$

as follows: from

$$2\mathfrak{w}_{d_0} \cdot \left(1 - \cos\left(\frac{\pi(n_{d_0}-1)}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right), \tag{38}$$

$$2\mathfrak{w}_{d_0} \cdot \left(1 + \cos\left(\frac{\pi}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right), \tag{39}$$

$$2\mathfrak{w}_{d_0} \cdot \left(2 - 2\sin^2\left(\frac{\pi/2}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(2\sin^2\left(\frac{\pi/2}{n_{d_1}}\right)\right), \tag{40}$$

$$2\mathfrak{w}_{d_0} \cdot \left(1 - \cos\left(\frac{\pi(n_{d_0}-1)}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(1 - \cos\left(\frac{\pi}{n_{d_1}}\right)\right), \tag{41}$$

we conclude that

$$\mathfrak{w}_{d_0} \cdot \left(2 - 2\sin^2\left(\frac{\pi/2}{n_{d_0}}\right)\right) < 2\mathfrak{w}_{d_1} \cdot \left(2\sin^2\left(\frac{\pi/2}{n_{d_1}}\right)\right), \tag{42}$$

For large graphs we obtain an approximation

$$2\mathfrak{w}_{d_0}\cdot\left(2-2\left(\frac{\pi/2^2}{n_{d_0}^2}\right)\right)<2\mathfrak{w}_{d_1}\cdot\left(2\left(\frac{\pi^2/2^2}{n_{d_1}^2}\right)\right)\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}}<\frac{\frac{\pi^2/2^2}{n_{d_1}^2}}{1-\left(\frac{\pi/2^2}{n_{d_0}^2}\right)}\Rightarrow$$

$$\Rightarrow\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}}<\frac{\frac{\pi^2/2^2n_{d_0}^2}{n_{d_1}^2}}{n_{d_0}^2-\pi/2^2}, \tag{43}$$

and for really large n_{d_0} we obtain

$$\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}}<\frac{\pi^2/2^2}{n_{d_1}^2}\Rightarrow\sqrt{\frac{\mathfrak{w}_{d_0}}{\mathfrak{w}_{d_1}}}<\frac{\pi/2}{n_{d_1}}, \tag{44}$$

which means that the competing dimension size alone drives the weight proportion. Again the square root cut and not the cut are really optimized.

8. Experiments

8.1. *k*-means and the number of eigenvectors to use in spectral clustering

We shall still ask whether or not fulfilling the formula (37) enables the k-means algorithm to produce n_{d_0} clusters by cutting the graph along the dimension d_0. We analyze this with respect to combinatorial Laplacians. The eigenvectors for the first n_{d_0} eigenvalues $\lambda_{[0,\ldots,(j-1)_{d_0},\ldots,0]}$ for $j=1,\ldots n_{d_0}$ will be of the form: the vector $v_{[0,\ldots,(j-1)_{d_0},\ldots,0]}$ will consist of the components

$$\nu_{[0,\ldots,(j-1)_{d_0},\ldots,0],[x_1,\ldots,x_d]}=\cos\left(\frac{\pi(j-1)}{n_{d_0}}\left(x_{d_0}-0.5\right)\right), \tag{45}$$

that is all data points $[x_1,\ldots,x_d]$ with the same coordinate x_{d_0}, i.e. lying in the same layer in the dimension d_0 will have identical coordinates in the spectral dimension $[0,\ldots,(j-1)_{d_0},\ldots,0]$ that is they will collapse to a single point in the spectral space. Thus, we will have n_{d_0} clusters with zero variance each, i.e. it is the clear case for a k-means algorithm.

But what will happen if we use more than k eigenvectors in spectral clustering, where k is the intrinsic number of clusters? Let us assume a 2D setting in which there are k layers in the first direction ($n_1=k$) and the weights of edges in the other direction (with n_2 layers) were too big for the eigenvalues to be included in the first k smallest eigenvalues. The next eigenvector would be of the form the vector $v_{[0,1]}$ which will consist of the components

$$\nu_{[0,1],[x_1,x_2]} = \cos\left(\frac{\pi}{n_2}(x_2 - 0.5)\right), \tag{46}$$

An experiment on taking more than k eigenvectors when clustering into k clusters was performed. The experiment had the following setup: three (two-dimensional) grid network types (*nettype*) were considered, with approx. 150, 200 and 250 nodes that were clustered into $k = 5,...,25$ clusters. The clusters were the layers in one dimension. The weights of the connections within each cluster were set to 1 and the weights between clusters were set so that the quotient of $k+1$st eigenvalue to kth eigenvalue was 1.1. The exact size of the cluster was set to the closest number bigger than $nettype/k$ that had the GCD with k equal to 1. k-means with 40-fold restart was used for groups of j first eigenvectors with $j = 2,...,nettype$. The number of the partitions deviating from the partition into exactly the predetermined k layers was counted as errors. It should be stressed that an error for $j = k$ never occurred. Figure 8 summarizes the results. Up to 70% of all the runs produced erroneous results. This means that adding eigenvectors beyond the first k produces noise only. This agrees with the general intuition that the eigenvectors should not be used as is, but rather they should be weighted inversely to the corresponding eigenvalue.

This problem is more general, applying to more general types of for was noticed also in [18] their Fig. 4. The spectral clustering should be unable to detect cuts with low isoperimetric ratios.

8.2. Impact of the distortion of edge weights

The question may be asked whether or not studying regular grid graphs with fixed edge weights in each direction of the grid is not too rigid an assumption. Hence, an experiment was performed the results of which are illustrated in Figures 9 and 10.

A regular grid with dimensions $k = 20$ and $n_2 = 17$ was constructed in such a way that $\lambda_{k+1}/\lambda_k = 1 + \lambda_{gap}$, where $\lambda_{gap} = 0.1$ in Fig. 9 and $\lambda_{gap} = 0.1$ in Fig. 10. Then edge weights were distorted by factors uniformly sampled (for each weight) from the interval from $1 - d$ to $1 + d$, where $d = 0.1$ in Fig. 9 and $d = 0.9$ in Fig. 10.

As can be seen in the Figures, the first k lowest eigenvalues do not deviate significantly in the original grid and in the distorted one (images to the right), though deviation can be observed for higher eigenvalues (images to the left) and the bigger the distortion factor the bigger the deviation.

Hence, the question seemed to be justified whether or not the application of k-means would yield same results for the original and the distorted grid. It turned out that under 16,000 fold restart of k-means in R implementation produced the same results when the recommended number of eigenvectors, i.e. k was used. However, if the number of eigenvectors was increased to $3k$, over 25% of nodes were clustered differently for the distorted grid compared to the regular one.

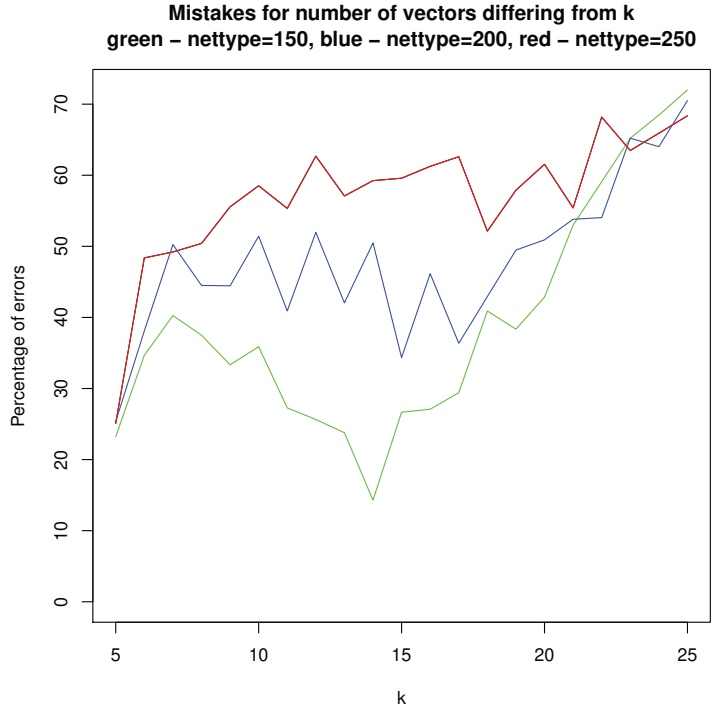

Figure 8. Errors committed by taking more than k eigenvectors for the purposes of clustering. X-axis: the number of clusters, Y-axis: the percentage of wrong partitions, net type: the approximate number of nodes in the network

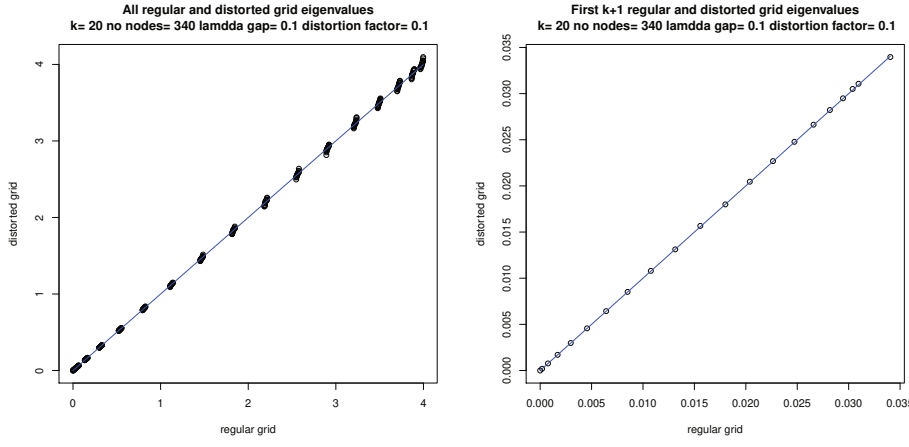

Figure 9. Relationship between the eigenvalues of regular grids and eigenvalues of grids with distorted edge weights. Distortion of up to 0.1. All eigenvalues to the left, $k+1$ lowest to the right.The plots of clustering of unweighted and weighted two-dimensional grid graphs. The smaller the distances, the higher the weights

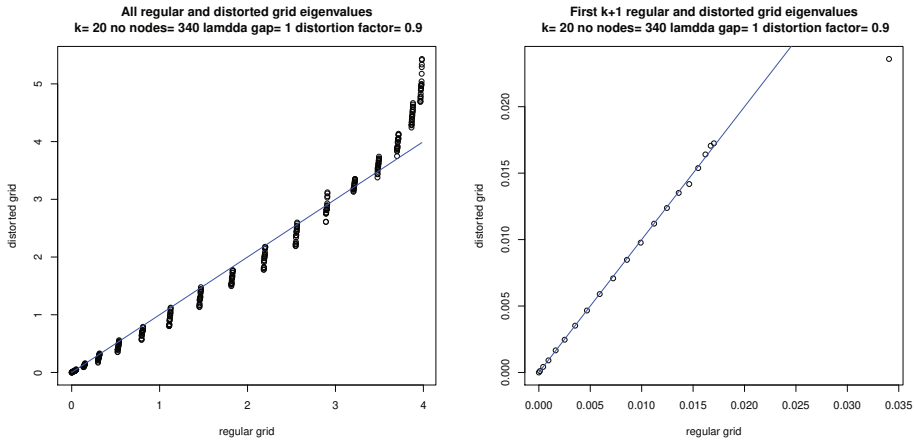

Figure 10. Relationship between the eigenvalues of regular grids and eigenvalues of grids with distorted edge weights. Distortion of up to 0.9. All eigenvalues to the left, $k+1$ lowest to the right

8.3. Ncut in other graphs

We performed also experiments checking how the Ncut would differ from spectral cut for non-grid graphs. Especially we considered graphs consisting of 4 components that were loosely coupled in pairs (1-2,3-4) and (1-3,2-4). It turned out that depending on the inner structure and inner weights of the four components, the split into two parts occurred at some threshold between weights and square-rooted weights of the connecting edges. For this reason our findings seem to be applicable predominantly to grid-like graphs.

9. Conclusions and Future Research

In this paper we presented a (closed- or nearly-closed form) method of computation of all eigenvalues and eigenvectors of a multi-dimensional weighted grid graph for unnormalized (or combinatorial), signless, normalized and random walk Laplacians.While the combinatorial and signless Laplacians of multi-dimensional weighted grid graph can be constructed from path graph Laplacians as a combination of them, it is no longer the case with normalized and random walk Laplacians.

The closed-form or nearly closed-form formulas for eigenvalues and eigenvectors for multidimensional weighted grid graphs may be of high interest for researchers dealing with cluster analysis of graphs [19], especially with spectral cluster analysis, and compressive spectral clustering (CSC) [20]. In particular, the CSC is based on the assumption that the eigenvalues of normalized Laplacian are uniformly distributed. However, when analyzing the cumulative distribution function of these eigenvalues in grids with different weights proportions, we observe the violation of such an assumption. This violation increases with an increase

in the unbalancedness of weights, see Figure 11, where the distribution of the eigenvalues of 2-dimensional weighted grid graphs is depicted.

Weighted grid graphs can be considered as types of graphs that have either no intrinsic cluster structure (when the weights are equal) or the structure of which can be twisted in various ways. Hence, the spectral clustering algorithms should be checked against such structures getting advantage of the fact that the eigenvectors and eigenvalues are quite easy to obtain even for large graphs. The weights permit to simulate node clusters not perfectly separated from each other, with various shades of this imperfection.

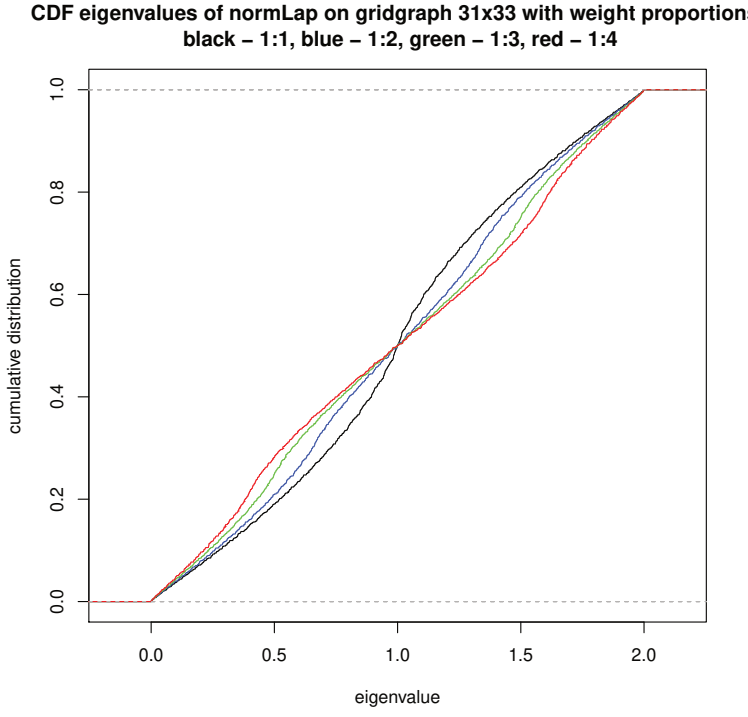

Figure 11. Cumulative distributions of eigenvalues of normalized Laplacians of grid graphs of approximately 1,000 nodes with various proportions of weights in both directions. The colors indicate: black - 1:1 (the unweighted case), blue - 1:2, green - 1:3, red - 1:4.

This fact opens new possibilities for exploitation of closed-form or nearly closed form solution eigenvectors and eigenvalues of graphs while testing and/or developing such algorithms and exploring their theoretical properties. Furthermore, in Sec. 7.1, we investigated the differences between various types of Laplacians, and hence, between various spectral clustering methods underpinned by them. It should be noted that eigenvectors of combinatorial and signless Laplacians of weighted graphs are identical with those for unweighted graphs. In the case of normalized and random walk Laplacians, the eigenvectors for weighted

graphs are superficially identical with those for unweighted ones, but nevertheless they differ because the shifts are influenced by weights.

The study of differences between the weighted and unweighted cases allows new insights into the nature of normalized and unnormalized Laplacians. Edge weights have no impact on the eigenvectors of combinatorial and signless Laplacians. It is only the presence or absence of an edge that impacts them. This is not the case with normalized and random walk Laplacians. Here the relative edge weights influence the shifts in the vector formulas. The weighting scheme opens up the possibility of manipulating the magnitude of eigenvalues of combinatorial and signless Laplacians, related to various grid dimensions. This has an interesting impact, for example, on the concept of the Fiedler vector, associated with the second lowest eigenvalue. The preferences which eigenvector to choose as the Fiedler vector (from among those with the lowest components) can be modified with the weight changes. The order of magnitude of eigenvectors associated with some direction can be changed and the impact on k-means clustering in spectral graph analysis can be observed. We have also the possibility to study the impact of relative weights of various dimensions in a grid graph on the normalized and random walk Laplacians, while, for example, the connection between various grid layers is fading.

In Sec. 7.2 we investigated whether or not the results of Spectral Clustering would match those of graph-cut and normalized graph-cut in the case of clustering into two clusters using the Fiedler vector. This study pointed out to some discrepancies between the common understanding of the relationship between spectral clustering and the cut and N-cut methods of graph clustering. It was demonstrated that spectral clustering would seek a cut with the smallest sum of square roots of edge weights and not of the edge weights themselves as commonly assumed.

In Sec. 7.3, we examined whether or not the results of Spectral Clustering would match those of graph-cut and normalized graph-cut in the case of clustering into more than two clusters using the k-means algorithm.

In Sec. 8.1, we probed the justifiability of the choice of k eigenvectors for clustering into k-clusters in spectral clustering. The paper provides supportive evidence that spectral clustering into k clusters should use (at most) k eigenvectors associated with the k lowest eigenvalues as an increase in the number of eigenvectors will introduce noise inducing blurring of the clusters.

In Sec. 8.2, we checked whether or not the conclusions drawn may be extended, if the weights were not uniform. It was shown that the regular weighted grid graphs quite accurately approximated the behavior of grids with distorted weights, if the lowest k eigenvalues with their eigenvectors were used in spectral clustering.

These results constitute a significant conceptual expansion of [16], allowing the weighing of edges in the grid graph. In order to be useful, all the theorems presented in [16] were reconsidered and rendered to fit the weighting scheme and the corresponding proofs were revised, wherever necessary.

Their properties may be of interest as generalizations of the results on unweighted grid graphs.

It is worth noticing that the multidimensional weighted grid graphs are bipartite graphs so that they may be exploited in the investigations of the properties of Laplacians of weighted bipartite graphs.

Further research should investigate the relationship between unweighted and weighted grid graphs and other analytical results in the domain of spectral clustering. For example, there exist results relating the kth eigenvector to a very special way of cutting a graph (the so-called Nodal Domain Theorem, [21], [1]). Let us assume that each node is assigned the corresponding coordinate of the mentioned eigenvector. Then, the edges connecting the nodes with different coordinate signs (nodes with the zero value assigned to any neighboring graph) are removed from the graph. In this case the graph will fall apart into k parts at the most. It would be worth investigating under what circumstances the k-means-based spectral clustering of a grid would agree or disagree with the mentioned partition.

References

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

[2] von Luxburg U 2007 *A tutorial on spectral clustering.*, *Statistics and Computing* **17** (4) 395

[3] Dingo C H Q, He X, Zha H, Gu M, and Simon H D 2001 *A min-max cut algorithm for graph partitioning and data clustering.*, *Proceedings of the 2001 IEEE International Conference on Data Mining*, IEEE Computer Society 107

[4] Dhillon I S, Guan Y and Kulis B 2005 *A unified view of kernel k-means, spectral clustering and graph cuts.*, *Technical Report TR-04-25*, University of Texas Dept. of Computer Science

[5] Shi J and Malik J 2000 *Normalized cuts and image segmentation.*, *IEEE Trans. Pattern Anal. Mach. Intell.* **22** (8) 888

[6] Meila M and Shi J 2001 *A random walks view of spectral segmentation*, *AI and STATISTICS (AISTATS)* 203

[7] Afsar M M and Tayarani-N M-H 2014 *Clustering in sensor networks: A literature survey.*, *Journal of Network and Computer Applications* **46** 198

[8] Shahraki A, Taherkordi A, Haugen O and Eliassen F 2020 *Clustering objectives in wireless sensor networks: A survey and research direction analysis.*, *Computer Networks* **180** 107376

[9] Li X, Claramunt Ch and Ray C 2010 *A grid graph-based model for the analysis of 2d indoor spaces*, *Computers, Environment and Urban Systems* **34** 532

[10] Weisstein E W 2017 *Grid graph.*, *From MathWorld - A Wolfram Web Resource*

[11] Edwards T 2013 *The discrete laplacian of a rectangular grid.*

[12] Bos A 2012 *Index notation of grid graphs.*, https://sites.math.washington.edu/~reu/papers/2013/tom/Discrete(2013)

[13] Fiedler M 1973 *Algebraic connectivity of graphs*, *Czech. Math. J.* **23** (98) 298

[14] Pozrikidis C 2014 *An Introduction to Grids, Graphs, and Networks*, OUP USA

[15] Sorkine O 2005 *Laplacian mesh processing.*, *Eurographics 2005 – State of the Art Reports*, The Eurographics Association 53

[16] Kłopotek M A 2019 *Spectral Analysis of Laplacian of a Multidimensional Grid Graph - Combinatorial versus Normalized and Random Walk Laplacians*, arXiv:1707.05210

[17] Cvetković D, Rowlinson P, and Simić S 2007 *Signless laplacians of finite graphs*, *Linear Algebra and its Applications* **423** (5) 155
[18] Grady L and Schwartz EL 2006 *Isoperimetric graph partitioning for image segmentation*, *IEEE Trans. on Pat. Anal. and Mach. Int* **28** 469
[19] Gallier J 2017 *Spectral Theory of Unsigned and Signed Graphs. Applications to Graph Clustering: a Survey*, arXiv:1601.04692
[20] Tremblay N, Puy G, Gribonval R and Vandergheynst P 2016 *Compressive spectral clustering.*, *Proc. of the 33rd Intl. Conf. on Machine Learning* **48** 1002
[21] Davies E, Gladwell G, Leydold J, and Stadler P 2001 *Discrete nodal domain theorems*, *Linear Algebra and its Applications*

Mieczysław Kłopotek is a professor of Computer Science at Polish Academy of Sciences, Warsaw, Poland. He obtained his Ph.D. in Computer Science in 1984 from the Faculty of Electrotechnics and Electronics Engineering at Dresden University of Technology, Germany. He got Habilitation Degree in 1999 from the Central Commission for Academic Degrees in Warsaw and the Professor title from the President of Poland in 2009. He previously held academic positions at Warsaw University of Technology and at Siedlce University of Natural Sciences and Humanities. He worked on projects for Polish Air Force Academy at Dęblin, Foundation for German-Polish Cooperation in Warsaw, European Union and for Polish Ministry of Science. He has consulted for a number of private companies, including Netezza, IBM, Semiconductor Research and Production Center "CEMI", iQor. His research interests include expert systems, machine learning, computer vision, intelligent search engines, text and web mining, highly parallel databases

Sławomir Wierzchoń is a professor of Computer Science at Polish Academy of Sciences, Warsaw, Poland. He obtained his Ph.D. in Computer Science and Engineering in 1979 from the Faculty of Electrical Engineering at Warsaw University of Technology. He got Habilitation Degree in 1997 from the Central Commission for Academic Degrees in Warsaw and the Professor title from the President of Poland in 2003. He previously held academic positions at Białystok University of Technology, Polish-Japanese Academy of Information Technology and Gdańsk University. In the past, he worked on projects for Polish Air Force Academy at Dęblin, and Military Institute of Health Services. His research interests include expert systems, machine learning, data analysis, evolutionary computations, text and web mining

Robert Kłopotek works as assistant professor at the Faculty of Mathematics and Natural Sciences, School of Exact Sciences of Cardinal Stefan Wyszyński University in Warsaw and is the deputy director at the Institute of Computer Science at Cardinal Stefan Wyszyński University in Warsaw, Poland. He obtained his Ph.D. in Computer Science in 2015 from the Institute of Computer Science of Polish Academy of Sciences in Warsaw. He has consulted for the USA company iQor. His research interests include graph visualization, social network analysis, text and web-mining, machine learning, recommender systems, parallel and distributed data processing, application of GPGPU in selected machine learning algorithms, solving polynomial equation systems using CUDA architecture, data mining and data analysis and survival analysis.