Multilevel Diffusion Based Spectral Graph Clustering

Malik Lechekhab, Dimosthenis Pasadakis, and Olaf Schenk, *Senior Member, IEEE*
Advanced Computing Laboratory
Università della Svizzera italiana
Lugano, Switzerland
{malik.lechekhab, dimosthenis.pasadakis, olaf.schenk}@usi.ch

***Abstract*—Spectral clustering and kernel k-means are ubiquitous methods for dividing non-linearly separable data into distinct groups. Spectral methods, while effective in partitioning non-convex spaces, are computationally intensive as they involve eigenvector computations. Conversely, kernel k-means maps data to higher dimensions and circumvents the need for this costly computation, with a weighted variant of it being mathematically equivalent to spectral clustering. This work extends this equivalence to diffusion based spectral methods and introduces the Multilevel Diffusion Clustering (MDC) algorithm. MDC leverages diffusion principles to minimize the normalized cut, adds flexibility through a diffusion parameter, and retrieves high-quality partitions in a multilevel fashion without computing eigenpairs. Our numerical examples and comparative results with modern multilevel graph clustering packages reveal that the proposed method can improve the clustering of graphs both in terms of balanced cut criteria and classification accuracy.**

***Index Terms*—Graph clustering, kernel k-means, spectral clustering, multilevel framework**

I. Introduction

Clustering data into distinct subsets with strong internal and weak external connections of roughly equal size is widespread in scientific fields that investigate interaction and interconnectedness. Over the years, numerous algorithms and techniques have been introduced, including spectral clustering [1] and kernel k-means [2], both designed to address non-linearly separable data. By utilizing the eigenvectors of a similarity matrix, spectral methods excel in partitioning in non-convex sample spaces through the optimization of graph cut objectives like the normalized cut [3]. Kernel k-means, on the other hand, can identify clusters by implicitly mapping data to higher dimensions, without the need for the computationally intensive process of calculating eigenpairs [4]. In [5], the authors establish a mathematical equivalence between a weighted form of the kernel k-means (WKKM) and general weighted graph clustering objectives, thus allowing WKKM to be used as an alternative to spectral methods. Leveraging this equivalence, a multilevel clustering algorithm was developed in [6] that progressively constructs coarser graphs, performs the initial partitioning on the coarsest instance using the Kernighan-Lin (KL) algorithm [7], and refines the partition at each subsequent level using WKKM.

Our work aims to build on these advancements by integrating concepts from a diffusion model-based spectral clustering approach, introduced in [8] for partitioning protein-protein interaction (PPI) networks, into the multilevel WKKM algorithm. Our method, called Multilevel Diffusion Clustering (MDC), leverages diffusion principles to minimize the normalized cut, while avoiding the eigenvector calculations inherent in diffusion model-based spectral clustering. Additionally, our method adds flexibility to the multilevel WKKM algorithm by allowing for the variation of a diffusion parameter β that controls partition quality. We achieve this through two main steps: first, by replacing the KL algorithm at the coarsest level with diffusion model-based spectral clustering, ensuring a high quality initial partitioning at the coarsest level. Second, we define a new kernel for WKKM partitioning at the subsequent levels that incorporates diffusion principles in its formulation. At each level, different values of the parameter β generate various candidate partitions, with the one achieving the lowest normalized cut being selected and used for projection and refinement at the next level.

II. Background on graph clustering

We recap in this section fundamental concepts of graph clustering. In Subsection II-A we introduce the matrices that characterize a graph, and in II-B the metrics that define a balanced partitioning. Then, in Subsection II-C we briefly revisit spectral graph clustering, and in II-D how diffusion principles can be incorporated in its formulation. Last, in II-E we present how a weighted kernel k-means algorithm can replace the eigenvector computations of spectral methods.

A. Graph notation

Let a set of n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a measure of similarities s_{ij} between all pairs of points $\mathbf{x}_i$ and $\mathbf{x}_j$. An effective method for representing the data and their similarities is to define an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V} = \{v_1, \ldots, v_n\}$ where each vertex v_i represents a data point $\mathbf{x}_i$, and an edge set $\mathcal{E}$ where each edge $e_{ij} \in \mathcal{E}$ represents a relationship between two vertices v_i and v_j.
The weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ of the graph stores the nonnegative, real-valued weights $w_{ij} \geq 0$ of the edge e_{ij}. The weight w_{ij} represents the similarity between vertices v_i and v_j; thus, $w_{ij} = 0$ indicates that there is no edge connecting the vertices v_i and v_j. This notion of similarity is commonly computed via a Gaussian kernel measuring distances between the data points [1], or via constrained inverse covariance matrix estimation [9]. In the case of a simple, undirected graph we consider that there are no self-loops, i.e., $w_{ii} = 0$, and that the adjacency matrix is symmetric with $w_{ij} = w_{ji}$.

The link function $\mathrm{links}(\mathcal{A},\mathcal{B})$ is defined as the sum of the edge weights between vertices in a subset $\mathcal{A} \in \mathcal{V}$ and another subset $\mathcal{B} \in \mathcal{V}$ by

$$\mathrm{links}(\mathcal{A},\mathcal{B}) = \sum_{v_i \in \mathcal{A}, v_j \in \mathcal{B}} w_{ij}. \tag{1}$$

The degree d_i of a vertex v_i refers to the sum of the weights of edges connecting v_i to the other vertices

$$d_i = \mathrm{links}(v_i, \mathcal{V}) = \sum_{j}^{n} w_{ij}. \tag{2}$$

Then, the degree matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ is defined as a diagonal matrix with the degrees of all vertices $d_1, \ldots, d_n$ in its diagonal.

B. Balanced graph clustering

The goal of clustering is to partition a graph into k disjoint clusters $\pi_1, \ldots, \pi_k$ such that the edges between groups have the lowest possible weights, while the edges within groups have the highest possible weights. To this end, the normalized association (NAssoc) and the normalized cut (NCut) [3] are widely adopted discrete graph clustering objectives, defined as

$$\mathrm{NAssoc}(\{\pi\}_{c=1}^{k}) = \max_{\pi_1,\ldots,\pi_k} \sum_{c=1}^{k} \frac{\mathrm{links}(\pi_c,\pi_c)}{\sum_{v_i\in\pi_c} d_i}, \tag{3}$$

$$\mathrm{NCut}(\{\pi\}_{c=1}^{k}) = \min_{\pi_1,\ldots,\pi_k} \sum_{c=1}^{k} \frac{\mathrm{links}(\pi_c,\mathcal{V}\setminus\pi_c)}{\sum_{v_i\in\pi_c} d_i}. \tag{4}$$

NAssoc measures the ratio between intracluster connectivity and the cluster's overall connectivity, while NCut places the intercluster connectivity in the numerator. Thus, maximizing the NAssoc objective is equivalent to minimizing NCut [6], [10]. Spectral methods that utilize the eigenvectors of matrices derived from $\mathbf{W}$ can be employed to optimize these discrete graph cut objectives.

C. Spectral graph clustering

Spectral clustering algorithms approximate solutions to graph partitioning problems by computing the eigenvectors of the symmetric positive semi-definite Laplacian matrix, which captures the connectivity and the edge weights of the graph. The Laplacian matrix of G is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. By analyzing the eigenvalues and eigenvectors of $\mathbf{L}$, spectral methods optimize the discrete objectives described in the Subsection II-B. Distance-based clustering algorithms like k-means can be applied on the eigenvectors corresponding to the k smallest eigenvalues to produce a discrete k-way partitioning of G. Note that vertices with large degrees d_i exert a larger influence on the eigenvector components and impact spectral graph analysis techniques such as clustering and centrality measures [11]. To mitigate this, normalization balances the influence of each vertex through

$$\mathbf{L}_{\mathrm{norm}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \tag{5}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix.

D. Diffusion matrix-based spectral clustering

The spectral analysis of the graph Laplacian can be realized as a diffusion-based probabilistic model [8]. From a probabilistic point of view, we consider a particle traveling between graph vertices in the entire vertex set $\mathcal{V}$ [12]. The matrix $\mathbf{\Omega} \in \mathbb{R}^{n\times n}$ is the transition matrix, where each entry $\mathbf{\Omega}_{ij}$ represents the probability of transitioning from vertex i to j. To incorporate the stationary distribution and the properties of the walk, $\mathbf{\Omega}$ can be defined as $\mathbf{\Omega} = \mathbf{D}^{-\beta}\mathbf{W}$ [8] where β is a parameter influencing the vertex degrees, and the diffusion matrix as

$$\mathbf{\Gamma} = \mathbf{\Delta} - \mathbf{\Omega}, \tag{6}$$

where $\mathbf{\Delta}$ is a diagonal matrix given by

$$\mathbf{\Delta}_{ij} = \delta_{ij} \sum_{l=1}^{n} \mathbf{\Omega}_{il} \tag{7}$$

and

$$\delta_{ij} = \begin{cases} 1, & \text{if i = j,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The diffusion matrix can be symmetrized by

$$\mathbf{\Gamma}_{\mathrm{sym}} = \mathbf{\Delta} - \mathbf{D}^{-\beta/2}\mathbf{W}\mathbf{D}^{-\beta/2}. \tag{9}$$

Spectral analysis of $\mathbf{\Gamma}_{\mathrm{sym}}$ and k-means on its eigenvectors can be utilized to efficiently identify the clusters of a graph.

E. Weighted kernel k-means

The weighted kernel k-means algorithm [13] seeks to find clusters $\pi_1, \pi_2, \ldots, \pi_k$ by minimizing the distance

$$\mathcal{D}(\{\pi\}_{c=1}^{k}) = \sum_{c=1}^{k} \sum_{\mathbf{x}_i\in\pi_c} d_i \|\phi(\mathbf{x}_i) - \mathbf{m}_c\|^2, \tag{10}$$

where $\mathbf{m}_c$ is the centroid of cluster π_c, defined as

$$\mathbf{m}_c = \frac{\sum_{\mathbf{x}_i\in\pi_c} d_i\phi(\mathbf{x}_i)}{\sum_{\mathbf{x}_i\in\pi_c} d_i}, \tag{11}$$

and ϕ is a function mapping data points to a higher-dimensional feature space allowing for nonlinear separators. The squared distance $\|\phi(\mathbf{x}_i) - \mathbf{m}_c\|^2$ can be expressed in terms of inner products

$$\begin{aligned} \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_i)\rangle - 2\frac{\sum_{\mathbf{x}_j\in\pi_c} d_j \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle}{\sum_{\mathbf{x}_j\in\pi_c} d_j} \\ + \frac{\sum_{\mathbf{x}_j,\mathbf{x}_l\in\pi_c} d_j d_l \langle\phi(\mathbf{x}_j),\phi(\mathbf{x}_l)\rangle}{(\sum_{\mathbf{x}_j\in\pi_c} d_j)^2}. \end{aligned} \tag{12}$$

Given a kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, where $\mathbf{K}_{ij} = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$, kernel methods allow us to compute distances between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space, represented by $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$, without directly evaluating these feature space representations [14]. Thus, (12) can be rewritten as

$$\mathbf{K}_{ii} - 2\frac{\sum_{\mathbf{x}_j\in\pi_c} d_j\mathbf{K}_{ij}}{\sum_{\mathbf{x}_j\in\pi_c} d_j} + \frac{\sum_{\mathbf{x}_j,\mathbf{x}_l\in\pi_c} d_j d_l \mathbf{K}_{jl}}{(\sum_{\mathbf{x}_j\in\pi_c} d_j)^2}. \tag{13}$$

Note that any positive semidefinite matrix can be considered as a kernel matrix, as demonstrated in [15].

The maximization of NAssoc (3) can be expressed as

$$\max \left\{ \sum_{c=1}^{k} \frac{\text{links}(\pi_c, \pi_c)}{\sum_{v_i \in \pi_c} d_i} = \sum_{c=1}^{k} \tilde{\mathbf{x}}_c^T \mathbf{W} \tilde{\mathbf{x}}_c \right\}, \tag{14}$$

where $\mathbf{x}_c \in \{0,1\}^n$ is an indicator vector for partition π_c and $\tilde{\mathbf{x}}_c = \mathbf{x}_c / (\mathbf{x}_c^T \mathbf{D} \mathbf{x}_c)^{1/2}$. Equivalently, the NCut objective (4) can be reformulated as

$$\min \left\{ \sum_{c=1}^{k} \frac{\text{links}(\pi_c, \mathcal{V} \setminus \pi_c)}{\sum_{v_i \in \pi_c} d_i} = \sum_{c=1}^{k} \tilde{\mathbf{x}}_c^T \mathbf{L} \tilde{\mathbf{x}}_c \right\}. \tag{15}$$

Setting $\mathbf{Y} = \mathbf{D}^{1/2} \tilde{\mathbf{X}}$, the optimization of NAssoc (3) can be expressed as a trace maximization problem [6]:

$$\max_{Y} \text{tr}(\mathbf{Y}^T \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \mathbf{Y}), \tag{16}$$

and that of NCut as

$$\min_{Y} \text{tr}(\mathbf{Y}^T \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{Y}). \tag{17}$$

Since $\mathbf{W} = \mathbf{D} - \mathbf{L}$, the ratio association can be reformulated as a normalized cut problem:

$$\begin{aligned} &\overbrace{\text{tr}(\mathbf{Y}^T \mathbf{D}^{-1/2} (\mathbf{D} - \mathbf{L}) \mathbf{D}^{-1/2} \mathbf{Y})}^{(16)} \\ &= k - \underbrace{\text{tr}(\mathbf{Y}^T \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{Y})}_{(17)}. \end{aligned} \tag{18}$$

Thus, optimizing (4) is equivalent to optimizing (3). Considering $\mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1}$ as a kernel matrix $\mathbf{K}$, the association problem (16) can be formulated as the weighted kernel k-means (WKKM) problem [4], [6] formulated in (13):

$$\max_{Y} \text{tr}(\mathbf{Y}^T \mathbf{D}^{1/2} \mathbf{K} \mathbf{D}^{1/2} \mathbf{Y}). \tag{19}$$

III. The Multilevel Diffusion Model-Based Clustering

We introduce a multilevel clustering method that leverages the diffusion principles presented in Section II-D to enhance the robustness and accuracy of the results produced by the multilevel weighted kernel k-means algorithm. At the coarsest level, spectral clustering of the diffusion Laplacian is employed, while at the subsequent levels of the uncoarsening phase, a new formulation of the weighted kernel of Section II-E is introduced. The key components of our Multilevel Diffusion Clustering (MDC) algorithm are illustrated in Figure 1.

A. The multilevel approach

Multilevel graph partitioning methods reduce partitioning times by progressively decreasing the size of the original graph G_0. Vertices and edges are collapsed into smaller instances $G_1, G_2, \ldots, G_m$, preserving the original structure with $|\mathcal{V}_0| > |\mathcal{V}_1| > \cdots > |\mathcal{V}_m|$. After coarsening to the final level G_m, a balanced partition P_m is computed and then projected onto finer graph instances $G_{m-1}, G_{m-2}, \ldots, G_1, G_0$ until it is projected onto the original graph G_0 with the complete set of edges and

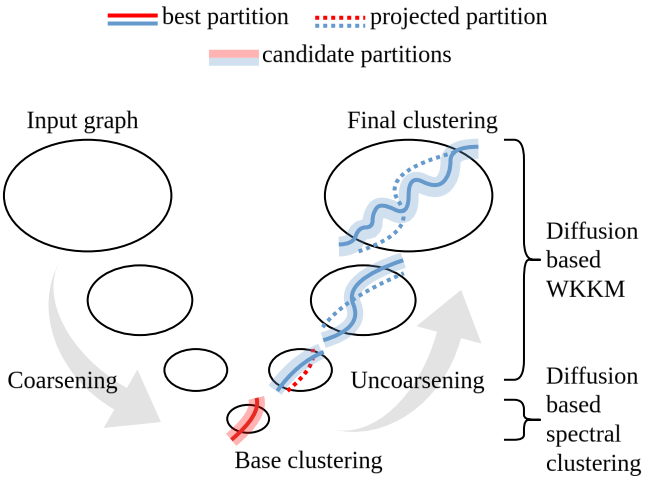


Fig. 1: Overview of the multilevel diffusion clustering (MDC). We incorporate diffusion principles at both the coarsest level with spectral clustering of $\mathbf{L}_{\beta_m}$ and at subsequent levels with a new kernel formulation.

nodes. During the uncoarsening phase, intermediate partitions $P_{m-1}, P_{m-2}, \ldots, P_1, P_0$ are refined to minimize cuts. The multilevel framework used in MDC is similar to that of METIS [16], [17], with differences in the partitioning and refinement strategies, as described in Subsections III-B and III-C.

1) Coarsening phase: A coarser graph G_{i+1} is constructed by merging nodes in G_i into supernodes in G_{i+1}. The edges emerging from a supernode are assigned weights that are the sum of the weights of the original collapsed edges [18]. Heavy edge matching is an efficient method for generating coarser graphs [17]. It involves randomly visiting vertices, selecting the adjacent node with the highest edge weight, and then merging these nodes into a single supernode. The size of the coarsest graph is set according to the default METIS [16] criterion defined as $|\mathcal{V}_m| = \max(|\mathcal{V}_0|/40 * \log_2 k, 20 * k)$.

2) Uncoarsening phase: Once the partition P_m is computed on the coarsest graph G_m, it is successively projected back through each intermediate level, up to the original graph G_0. Initially, each vertex in the finer graph G_{i-1} is assigned to a unique cluster, as G_{i-1} is composed of distinct subsets of the vertices from the coarser graph G_i. With the increasing number of vertices, it becomes essential to re-evaluate and adjust these cluster assignments on the finer graph G_{i-1}, ensuring that the partitioning remains optimal as the graph increases in size. To this end, a modified version of the WKKM algorithm, discussed in detail in Subsection III-C, leverages the information from the partition projected from the coarser level.

B. Diffusion model-based spectral clustering of the coarsest graph

At the coarsest level m, the graph Laplacian matrix of the reduced graph is constructed, and a Rayleigh quotient optimization problem is solved to compute the eigenvectors associated with the k-smallest eigenvalues [19]. We incorporate the approach of diffusion model-based spectral clustering

(ADMSC) [8] into the formulation of the Laplacian at the coarsest level of our method:

$$\mathbf{L}_{\beta_m} = \mathbf{I} - \mathbf{D}^{-\beta_m/2}\mathbf{W}\mathbf{D}^{-\beta_m/2}. \tag{20}$$

ADMSC introduces a power factor, referred to as β_m at this coarsest level, which adjusts the Laplacian matrix to account for network heterogeneity, thereby influencing the resulting clusters. We utilize the STAG [20] library to compute the eigenvectors associated with the k-smallest eigenvalues of $\mathbf{L}_{\beta_m}$, and apply k-means on them to obtain k discrete clusters. The reduced size of this eigenproblem at the coarsest level allows for multiple runs with different values of $\beta_m \in [0, 2]$, following the approach of [8]. Varying the β_m value generates multiple candidate partitions, from which we select the partition with the lowest NCut. This ensures a high-quality initial partition for the subsequent uncoarsening phases, where the graph is progressively refined. Note that eigenvector computations are only performed at this coarsest level due to the small size of the graph at this stage.

C. Diffusion-based WKKM clustering of the intermediate graphs

In the uncoarsening levels, we leverage the effectiveness of diffusion model-based spectral clustering and the equivalence between spectral analysis and WKKM to obtain high quality partitions. A fundamental linear algebra result [21], [22] states that trace maximization problems in the form of (19), with a relaxation allowing $\mathbf{Y} = \tilde{\mathbf{U}}\mathbf{Q}$ where $\mathbf{Q}$ is an arbitrary orthogonal $k \times k$ matrix, are solved by selecting $\tilde{\mathbf{U}}$ as the $n \times k$ matrix containing eigenvectors associated to the k-smallest eigenvalues of $\mathbf{D}^{1/2}\mathbf{K}\mathbf{D}^{1/2}$ as columns. This establishes an equivalence between spectral clustering and kernel k-means, thus allowing the substitution of the computationally intensive task of computing graph Laplacian eigenvectors by the computationally cheaper WKKM method.

Our MDC algorithm introduces the adjustable parameter from (6), extending the equivalence between WKKM and diffusion spectral clustering. Specifically, clustering using the symmetrized diffusion matrix (9) is achieved by configuring the kernel $\mathbf{K}$ to

$$\begin{aligned} \mathbf{D}^{1/2}\mathbf{K}\mathbf{D}^{1/2} &= \mathbf{\Delta} - \mathbf{D}^{-\beta/2}\mathbf{W}\mathbf{D}^{-\beta/2} \\ \mathbf{K} &= \mathbf{D}^{-1/2}(\mathbf{\Delta} - \mathbf{D}^{-\beta/2}\mathbf{W}\mathbf{D}^{-\beta/2})\mathbf{D}^{-1/2} \\ \mathbf{K} &= \mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2} - \mathbf{D}^{-\alpha}\mathbf{W}\mathbf{D}^{-\alpha}, \end{aligned} \tag{21}$$

where $\alpha = \frac{1}{2}(1+\beta)$, and β is the parameter that influences the diffusion process. The trace maximization of (16) is performed as an equivalent spectral clustering method for the diffusion Laplacian described in (9). The parameter β allows for multiple runs with different values at each level of the uncoarsening process. As in Section III-B, by systematically varying β and evaluating the resulting candidate partitions, the optimal values of $\beta \in [0, 2]$ that yield the smallest NCut are identified. The best partition is then retained to be used as the initial partition for the next uncoarsening level.

From a diffusion perspective, introducing β as a power factor enables the adjustment of the diffusion matrix $\mathbf{\Gamma}$ by weighting the transition matrix $\mathbf{\Omega}$ according to the degree matrix. By tuning β in (9), the diffusion process can emphasize either short-range or long-range connections. Higher β values reduce the influence of high-degree nodes, enhancing local clustering and emphasizing short-range connections. Lower β values increase the influence of high-degree nodes, capturing larger structures and emphasizing long-range connections [8], [23]. This adjustable parameter allows the algorithm to be fine-tuned to the specific properties of the graph and ensures optimal graph cuts across a range of different graphical structures.

Moreover, (21) incorporates the diffusion characteristics directly into every level of the uncoarsening phase of the multilevel approach described in Section III-A, aligning the kernel matrix $\mathbf{K}$ closely with the properties of the diffusion Laplacians of the graphs $G_{m-1}, G_{m-2}, \ldots, G_1, G_0$. The flexibility introduced by the parameter β reduces the cut edges across all partitions $P_{m-1}, P_{m-2}, \ldots, P_1, P_0$, maintaining the computational benefits of WKKM while improving partitions quality. From a computational perspective, it necessitates the calculation of the distance from point $\mathbf{x}_i$ to the centroid of the cluster π_c, as expressed in (13). This requires determining the individual entries of the kernel matrix $\mathbf{K}$, defined element-wise as

$$\mathbf{K}_{ij} = d_i^{-\beta} - d_i^{-\alpha} w_{ij} d_i^{-\alpha}. \tag{22}$$

Note that $\mathbf{K}_{ii}$ in (13) is a constant and can be excluded from the optimization process. Consequently, the first term $d_i^{-\beta}$ does not affect the distance computation and can also be omitted.

D. A multilevel diffusion clustering (MDC) algorithm

This section details the two core algorithms employed in our methodology: the spectral clustering of the coarsest graph through the diffusion Laplacian, and the multilevel diffusion clustering of the subsequent graphs. Diffusion principles are introduced via the perturbation parameters β_m for the first algorithm and β for the second. For different values of β_m, Algorithm 1 performs spectral clustering on the coarsest graph G_m. It computes the graph Laplacian $\mathbf{L}_{\beta_m}$ (line 3) and performs eigenvalue decomposition to obtain the eigenvectors $\tilde{\mathbf{U}}$ corresponding to the k-smallest eigenvalues (lines 4 and 5). These eigenvectors are then clustered with k-means to produce k discrete partitions (line 6), and the ones with the smallest NCut are returned (lines 7 – 10).

Algorithm 2 applies diffusion clustering to the subsequent graphs G_l. It computes the diffusion weighted kernel $\mathbf{K}$ (line 3), and performs WKKM (lines 5 – 9). The objective is to iteratively improve the partitions resulting from the coarser graph G_{l+1} for different values of β, and to project the partitions that minimize NCut at each level l to the next level $l-1$ (lines 10 – 13). In both algorithms, the β iteration loop is parallelizable within the same level, as the iterations operate independently of each other.

E. Computational complexity

The main computationally intensive tasks of Algorithm 1 lie in lines 4 and lines 6, involving the spectral decomposition and the k-means clustering. STAG [20] computes eigenpairs

Algorithm 1 Diffusion spectral clustering of the coarsest graph

Input: k : number of clusters,
$\mathbf{W}^{(m)}$: weighted adj. matrix of the graph G_m
Output: $\pi_1^{(m)}, \ldots, \pi_k^{(m)}$: partitioning of G_m
1: Compute d_i of each vertex $v_i \in \mathcal{V}_m$ // acc. to (2)
2: **for** $\beta_m = 0, \Delta\beta_m, \ldots, 2$ **do**
3: Compute the graph Laplacian $\mathbf{L}_{\beta_m}$ // acc. to (20)
4: Perform eigenvalue decomposition: $\mathbf{L}_{\beta_m} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$
5: Extract $\tilde{\mathbf{U}}$: the k columns of $\mathbf{U}$ corresponding to the k-smallest eigenvalues
6: Apply k-means on the rows of $\tilde{\mathbf{U}}$
7: Compute current NCut // acc. to (4)
8: **if** Current NCut $<$ previous NCut **then**
9: Update clusters $\pi_1^{(m)}, \ldots, \pi_k^{(m)}$
10: **return** Clusters $\pi_1^{(m)}, \ldots, \pi_k^{(m)}$

Algorithm 2 Multilevel diffusion based clustering of the subsequent graphs

Input: k : number of clusters,
$\mathbf{W}^{(l)}$: weighted adj. matrix of the graph G_l with $l \in \{m-1, m-2, \ldots, 1, 0\}$
$\pi_1^{(l+1)}, \ldots, \pi_k^{(l+1)}$: partitioning of G_{l+1}
Output: $\pi_1^{(l)}, \ldots, \pi_k^{(l)}$: partitioning of G_l
1: Compute d_i of each vertex $v_i \in G_l$ // acc. to (2)
2: **for** $\beta = 0, \Delta\beta, \ldots, 2$ **do**
3: Compute the kernel $\mathbf{K}$ // acc. to (22)
4: Initialize the k clusters: $\pi_1^{(l)}, \ldots, \pi_k^{(l)}$
5: **repeat**
6: **for** each point $\mathbf{x}$ **do**
7: Find the new cluster index as $j^*(\mathbf{x}) = \text{argmin}_j \left\{ ||\phi(\mathbf{x}) - \mathbf{m}_j||^2 \right\}$ // acc. to (13)
8: Update the cluster as $\pi_j^{(l)} = \{\mathbf{x} : j^*(\mathbf{x}) = j\}$
9: **until** Convergence of objective // acc. to (16)
10: Compute current NCut // acc. to (4)
11: **if** Current NCut $<$ previous NCut **then**
12: Update clusters $\pi_1^{(l)}, \ldots, \pi_k^{(l)}$
13: **return** Clusters $\pi_1^{(l)}, \ldots, \pi_k^{(l)}$

using the Lanczos method [24], typically performing $O(d_m n_m^2)$ operations [22], where d_m is the average number of non-zero elements in a row of the Laplacian $\mathbf{L}_{\beta_m}$, and n_m is the number of vertices in the coarsest graph G_m. The k-means algorithm has a complexity of $O(\tau_m k^2 n_m)$, where k is the number of clusters, and τ_m is the number of iterations. Since these operations are computed v times for different β_m, the total complexity of Algorithm 1 is $O(v(dn^2 + \tau k^2 n)_m)$.

Algorithm 2's bottlenecks are the kernel computation in line 3 and the distance evaluation in lines 5 – 9. each with a complexity of $O(n_l^2)$ when the kernel $\mathbf{K}$ is dense, where n_l is the number of vertices in the intermediate graph G_l. For v values of β and τ_l iterations, the total complexity of Algorithm 2 is $O(v(dn + \tau n^2)_l)$. Thus, MDC's overall complexity is $O(v(dn^2 + \tau k^2 n)_m + v(dn + \tau n^2)_l)$, which simplifies to $O(v\tau n^2)$ given that $k \ll n_m < n_l < n = |\mathcal{V}|$, where τ is the maximum number of iterations at the finest level $l = 0$. In comparison, Graclus has an asymptotic complexity of $O(\tau n^2)$, which is of the same order as MDC's complexity. Directly applying the Lanczos method to compute the eigenvectors of the input graph Laplacian results in a complexity of $O(dn^2)$, where d is the average number of non-zero elements per row. For dense matrices, where $d \to n$, this increases to $O(n^3)$.

IV. Numerical results

We demonstrate in this section the effectiveness of the multilevel diffusion based spectral graph clustering framework, as summarized in Algorithms 1 and 2. In Subsection IV-A, we describe the setup of the numerical experiments and the methods considered in our comparative studies. Results on synthetic data are presented in Subsection IV-B and on real-world graphs in Subsection IV-C.

A. Experimental setup

We report results about the quality of the cut in terms of normalized cut (NCut) (4) and modularity Q which describes the number of edges belonging to clusters, minus the anticipated number in an equivalent graph with a random distribution of edges [25]. Accuracy is measured in terms of normalized mutual information, NMI $\in [0, 1]$, and variation of information [26], VI $\in [0, \log n]$. Both metrics are invariant to the permutations of the label values, ensuring that the absolute values of the labels do not affect the scores. NMI values close to 1 and VI values near 0 indicate better alignment with the true labels [26]. We utilize Julia (1.7.2) notebooks to interface with our C++ implementation, The diffusion parameter for the Laplacian $\mathbf{L}_{\beta_m}$ (20) and for the kernel $\mathbf{K}$ (21) at the subsequent levels varies within the range $[0, 2]$, with increments of 0.1.

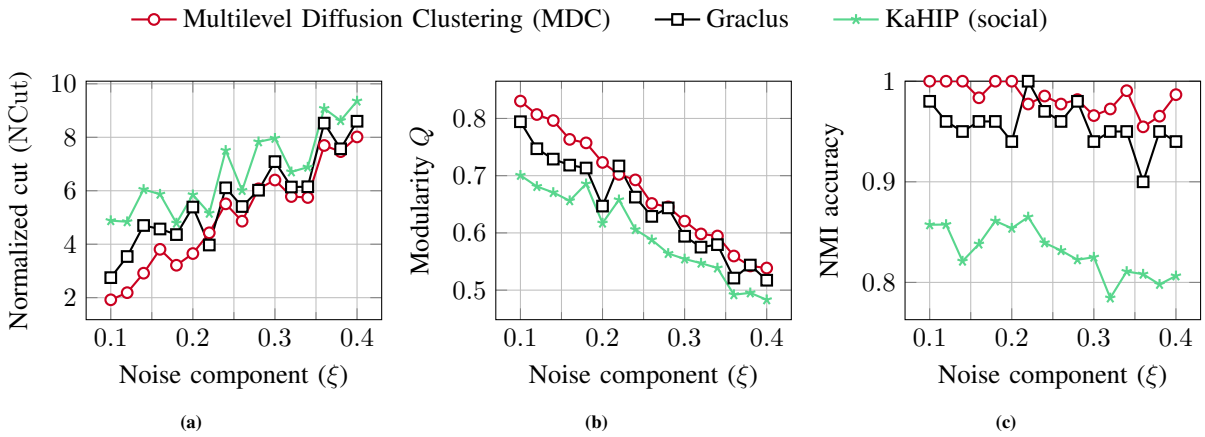
Our method is compared against two state-of-the-art multilevel clustering frameworks[1]:

1) Graclus [6]: Optimization of weighted graph clustering objectives with a weighted kernel k-means, using Kernighan-Lin [7] for the initial clustering. Our algorithm builds upon the code release of this method.
2) KaFFPa - Karlsruhe Fast Flow Partitioner [27]: A multilevel algorithm within KaHIP that uses techniques such as max-flow min-cut, local improvement and multi-try Fidducia-Mattheyses [28]. We use KaFFPa "social" [29] for optimal partitioning of irregular graphs.

B. Experiments with synthetic data

We consider a popular benchmark set of artificial datasets to demonstrate the behavior of our MDC algorithm across scenarios of varying difficulty. The Lancichinetti–Fortunato–Radicchi (LFR) datasets [30] are based on a stochastic block model where node degrees and community sizes follow power-law distributions. A parameter ξ, referred to as noise component, controls the fraction of a node's neighbors that are outside its community. Following the approach of [31], we select ξ

[1]The Graclus code is available at: https://www.cs.utexas.edu/~dml/Software/graclus.html, and the KaHIP code at: https://github.com/KaHIP/KaHIP.



(a) (b) (c)

Fig. 2: Clustering the LFR datasets for an increasing noise component ξ. (a) NCut values. (b) Modularity values. (c) NMI accuracy values.

Test case	Normalized cut (NCut)			Normalized mutual information (NMI)			Variation of information (VI)		
	MDC	Graclus	KaHIP	MDC	Graclus	KaHIP	MDC	Graclus	KaHIP
Binaryalphadigs	**12.381**	12.820	13.592	**0.613**	0.603	0.595	**2.736**	2.832	2.906
mfeatzernike	**0.848**	1.037	0.966	**0.715**	0.693	0.700	**1.226**	1.414	1.379
Ecoli	**1.559**	1.676	1.860	**0.540**	0.493	0.507	**1.650**	1.830	1.794
har	**0.160**	0.286	0.305	**0.691**	0.648	0.615	1.789	**1.242**	1.378
indianpines	**0.183**	0.294	0.294	**0.462**	0.388	0.357	**1.767**	2.159	2.286
MNIST	**0.795**	0.914	0.823	0.749	0.699	**0.789**	1.134	1.380	**0.970**
Fashion MNIST	**0.576**	0.755	0.705	0.598	0.582	**0.608**	**1.804**	1.923	1.806
Japanese Vowels	**0.510**	0.638	0.583	0.675	0.639	**0.721**	1.345	1.568	**1.220**

TABLE I. Classification results for the image datasets of Section IV-C. The value of the best method achieved is in bold font.

in the range $[0.1, 0.4]$. This selection produces graphs with increasingly noisy clusters as the value of ξ increases. The number of clusters k in this benchmark ranges from 17 to 21. The NCut value of the final partition increases for an increasing noise component ξ, as illustrated in Figure 2a. MDC yields a strictly lower NCut compared to Graclus in 14 out of the 16 graphs, and consistently outperforms KaHIP. On average, MDC achieves a NCut that is 14% lower than that of Graclus and 28% lower than that of KaHIP. We then present in Figure 2b the values of modularity Q achieved by each method. MDC reports a strictly higher value of Q for 13/16 cases, and is again consistently better than KaHIP. On average, MDC results in 5% higher modularity values compared to Graclus, and 13% higher than KaHIP. Last, the NMI values for the three methods are presented in Figure 2c. Our method achieves a NMI equal or close to 1 for a noise component $\xi \leq 0.2$, indicating a high level of accuracy between the produced clusters and the true community structure. The NMI of the partitions computed by our algorithm surpasses that of Graclus in 15/16 cases and consistently exceeds that of KaHIP across all cases. On average, the NMI of MDC is 3% higher compared to Graclus and 19% higher compared to KaHIP.

C. Clustering real-world instances

We consider a subset of the ML_Graph group [32] from the University of Florida sparse matrix collection [33] for our real world experiments.[2] MDC reports the lowest values of NCut for all the graphs considered. With respect to classification accuracy, MDC is the best method in 5/8 cases in terms of NMI and VI. KaHIP achieved the best NMI in 3/8 cases and the highest VI in 2/8 cases. Graclus is the best method in 1/8 cases in terms of VI.

V. Conclusion & Outlook

In this work we have introduced diffusion principles in a multilevel spectral graph clustering framework. The initial partitioning at the coarsest and sparsest level is achieved by an eigenvector analysis of the diffusion Laplacian. Subsequently, at each uncoarsening phase, a diffusion weighted kernel k-means is employed, avoiding the need for the computationally expensive eigenspectrum computations and refines the projected cluster assignment from the previous level. Our experiments with synthetic and real-world data demonstrate the potential of the MDC algorithm in both minimizing graph cut metrics and maximizing classification accuracy. The source code and timing results will be presented in future work, following the exploitation of the parallelization potential that this multilevel method offers. We intend to further explore the spectral properties of graphs that benefit from diffusion clustering and extend the equivalence between WKKM and spectral clustering to modularity optimization [34].

[2]The ML_Graph group is available at https://sparse.tamu.edu/ML_Graph. Details for the datasets of Table I are offered therein.

ACKNOWLEDGMENT

This work was financially supported by the joint DFG - 470857344 and SNSF - 204817 project. We would also like to acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). NHR@FAU funding is provided by federal and Bavarian state authorities and by the German Research Foundation (DFG) - 440719683.

REFERENCES

[1] U. von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[2] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," Neural Computation, vol. 10, no. 5, pp. 1299–1319, 07 1998.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[4] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 551–556.

[5] ——, "A Unified View of Kernel k-means, Spectral Clustering and Graph Cuts," University of Texas Dept. of Computer Science, Tech. Rep. TR-04-25, 2005. [Online]. Available: http://www.cs.utexas.edu/ftp/pub/techreports/tr04-25.pdf

[6] ——, "Weighted graph cuts without eigenvectors a multilevel approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 11, pp. 1944–1957, 2007.

[7] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," The Bell System Technical Journal, vol. 49, no. 2, pp. 291–307, 1970.

[8] K. Inoue, W. Li, and H. Kurata, "Diffusion model based spectral clustering for protein-protein interaction networks," PLOS ONE, vol. 5, no. 9, pp. 1–10, 09 2010.

[9] D. Pasadakis, M. Bollhöfer, and O. Schenk, "Sparse quadratic approximation for graph learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 9, pp. 11 256–11 269, 2023.

[10] Yu and Shi, "Multiclass spectral clustering," in Proceedings Ninth IEEE International Conference on Computer Vision, 2003, pp. 313–319 vol.1.

[11] F. R. Chung, Spectral graph theory. American Mathematical Soc., 1997, vol. 92.

[12] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," in Proceedings of the 18th International Conference on Neural Information Processing Systems, ser. NIPS'05. Cambridge, MA, USA: MIT Press, 2005, p. 955–962.

[13] F. Bach and M. Jordan, "Learning spectral clustering," in Advances in Neural Information Processing Systems, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16. MIT Press, 2003.

[14] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," The Annals of Statistics, vol. 36, no. 3, pp. 1171 – 1220, 2008.

[15] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.

[16] G. Karypis and V. Kumar, METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, September 1998.

[17] ——, "A fast and high quality multilevel scheme for partitioning irregular graphs," SIAM Journal on Scientific Computing, vol. 20, no. 1, pp. 359–392, 1998.

[18] W. Che, Z. Liu, Y. Wang, and J. Liu, "Merit: multi-level graph embedding refinement framework for large-scale graph," Complex & Intelligent Systems, vol. 10, no. 1, p. 1303–1318, Aug. 2023.

[19] R.-C. Li, "Rayleigh quotient based optimization methods for eigenvalue problems," in Matrix Functions and Matrix Equations. World Scientific, 2015, pp. 76–108.

[20] P. Macgregor and H. Sun, "Spectral toolkit of algorithms for graphs: Technical report (1)," CoRR, vol. abs/2304.03170, 2023.

[21] H. Lütkepohl, Handbook of matrices. John Wiley & Sons, 1997.

[22] G. H. Golub and C. F. Van Loan, Matrix computations. JHU press, 2013.

[23] B. Kozma, M. B. Hastings, and G. Korniss, "Diffusion processes on power-law small-world networks," Phys. Rev. Lett., vol. 95, p. 018701, Jun 2005.

[24] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," Journal of Research of the National Bureau of Standards, vol. 45, no. 4, p. 255, Oct. 1950.

[25] M. E. J. Newman, "Modularity and community structure in networks," Proceedings of the National Academy of Sciences, vol. 103, no. 23, pp. 8577–8582, 2006.

[26] M. Meilă, "Comparing clusterings—an information based distance," Journal of Multivariate Analysis, vol. 98, no. 5, pp. 873–895, 2007.

[27] P. Sanders and C. Schulz, "Think locally, act globally: Highly balanced graph partitioning," in Experimental Algorithms, V. Bonifaci, C. Demetrescu, and A. Marchetti-Spaccamela, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 164–175.

[28] C. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions," in 19th Design Automation Conference, 1982, pp. 175–181.

[29] H. Meyerhenke, P. Sanders, and C. Schulz, "Partitioning (hierarchically clustered) complex networks via size-constrained graph clustering," Journal of Heuristics, vol. 22, no. 5, p. 759–782, oct 2016.

[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Phys. Rev. E, vol. 78, p. 046110, Oct 2008.

[31] K. Fountoulakis, D. Wang, and S. Yang, "p-norm flow diffusion for local graph clustering," in Proceedings of the 37th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3222–3232.

[32] D. Pasadakis, C. L. Alappat, O. Schenk, and G. Wellein, "Multiway p-spectral graph cuts on Grassmann manifolds," Mach. Learn., vol. 111, no. 2, p. 791–829, feb 2022.

[33] T. A. Davis and Y. Hu, "The university of Florida sparse matrix collection," ACM Trans. Math. Softw., vol. 38, no. 1, dec 2011.

[34] D. Floros, N. Pitsianis, and X. Sun, "The Fiedler connection to the parametrized modularity optimization for community detection," 2023. [Online]. Available: https://arxiv.org/abs/2310.14359