
Spectral Clustering with Smooth Tiny Clusters

Hengrui Wang
 wanghr1230@pku.edu.cn
 Yuanpei College
 Peking University

Yubo Zhang
 zhangyubo18@pku.edu.cn
 School of electronic engineering and computer science
 Peking University

Mingzhi Chen
 1800017712@pku.edu.cn
 School of electronic engineering and computer science
 Peking University

Tong Yang
 *yangtongemail@gmail.com
 School of electronic engineering and computer science
 Peking University

Abstract

Spectral clustering is one of the most prominent clustering approaches. The distance-based similarity is the most widely used method for spectral clustering. However, people have already noticed that this is not suitable for multi-scale data, as the distance varies a lot for clusters with different densities. State of the art (ROSC and CAST) addresses this limitation by taking the reachability similarity of objects into account. However, we observe that in real-world scenarios, data in the same cluster tend to present in a smooth manner, and previous algorithms never take this into account.

Based on this observation, we propose a novel clustering algorithm, which considers the smoothness of data for the first time. We first divide objects into a great many tiny clusters. Our key idea is to cluster tiny clusters, whose centers constitute smooth graphs. Theoretical analysis and experimental results show that our clustering algorithm significantly outperforms state of the art. Although in this paper, we singly focus on multi-scale situations, the idea of data smoothness can certainly be extended to any clustering algorithms.

1 Introduction

1.1 Background and Motivation

Spectral clustering is a new and hot topic in recent years [1], and many researchers have focus on improving spectral cluster's performance. Spectral clustering is a set of algorithms which use graph partitioning methods to cluster objects. These set of algorithms have been widely used in many tasks, such as text mining [2] and network analysis [3–5] and even medical image analysis [6–8]. Typically, spectral clustering algorithms use a distance-based similarity matrix to indicate the similarity among objects, and this works well in many cases. But this considers only the feature similarity, which is quite straightforward.

*Corresponding author: yangtongemail@gmail.com

Researchers have found that spectral methods can perform really poor when used on multi-scale data, whose clusters various a lot in sizes and densities. Because the standard of similarity varies a lot, for sparse clusters, distances among objects are always much larger than distances among objects from dense clusters. Typically, existing solutions either scale the similarity matrix or apply the power iteration technique to derive pseudo-eigenvectors with rich cluster separation information. Among all the methods focusing on multi-scale data problem, the ROSC [9] and CAST [10] algorithms are state of the art. ROCS and CAST handle multi-scale data by extracting separation information from pseudo-eigenvectors and tuning the given traditional distance-based similarity matrix to a coefficient matrix, considering reachability similarity.

1.2 Our Proposed Solution

In this paper, we propose a new algorithm, namely Smooth-Clustering. Smooth-Clustering is the first algorithm that considers the smoothness of data for clustering problems. As mentioned above, spectral clustering is actually a graph partition problem, which means that the distribution of objects tend to be smooth, and this is quite consistent with real-world scenarios [11]. If there exists a sudden change among different objects, they belong to different groups with high probability. We call this *data smoothness* in the rest of our paper.

In fact, data smoothness is quite difficult to define because under extremely small scale, all data will seems discrete. To address this problem, first, we divide objects into a great many tiny clusters. Specifically, we use traditional cluster algorithms to cluster extremely close objects into one group and treated them as a tiny cluster. In the rest of the clustering process, we will use the center of the tiny cluster as an object. In this way, no objects will be extremely close. Second, we cluster tiny clusters, whose centers constitute smooth graphs. Specifically, we add an extra penalty term to embedding smoothness information when getting the coefficient matrix.

In this way, we can take data smoothness into account when dealing with the multi-scale data problem. For objects from different clusters, they may happen to be close or even reachable to each other, under which situation no previous algorithms can work well. However, for objects from different algorithms, their feature distribution will always vary a lot and will be in a discrete manner. When considering data smoothness, this problem can be easily fixed.

In this paper, we present a concrete spectral cluster algorithm, which considers feature similarity, reachability similarity, and data smoothness, focusing on multi-scale data. But we notice that the performance of most of the existing spectral clustering method is always affected by randomness. However, when clustering, we can hardly judge the performance of one clustering by any metrics for the lack of labels. Therefore, *our idea of data smoothness can be definitely applied to any spectral cluster algorithms to achieve better performance.*

Our main contributions are as follows:

- This is the first work that considers smoothness of tiny clusters in spectral clustering.
- We propose a new algorithm embedding smoothness information into spectral clustering algorithms.
- We derived mathematical proofs to show our algorithm can work well for multi-scale data.
- We conduct extensive experiments to show the effectiveness of our algorithm by comparing its performance with more than 10 previous clustering algorithms. The experimental results show that our algorithm provides high performance on different datasets. We also conduct experiments on synthetic datasets to show that our algorithm is efficient when data smoothness is related to clustering result. These experimental results will be provided in the full version of this paper.
- We will show the experimental results of applying the idea of data smoothness to many more clustering algorithms.

2 Related Work

Spectral clustering is a hot and new topic. Assuming we have a set of n objects $\mathcal{X} = \{x_1, \dots, x_n\}$, and a similarity matrix S to indicate the similarity among objects. One of the most widely used sim-

ilarity matrix is the traditional Gaussian kernel based similarity $S_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$. Spectral clustering turns objects and similarity matrix into a weighted graph $G = (X, S)$, transforming \mathcal{X} to the set of vertices and entries of S to weights of corresponding edges. The clustering problem is then transformed into a graph partition problem. We need to partition the graph G into different groups and to optimize the quality [12] of the partitioning.

A great number of researches has been conducted to focus on different aspects of spectral clustering. Some studies focus on spectral clustering for different data characteristic [13–19], on computational efficiency and speed [20–27], and on the theoretical side [28–34]. Some researchers even combine deep learning with spectral clustering [35–37].

Although spectral clustering has achieved great success, spectral clustering’s effectiveness suffering from degradation when dealing with noisy [38] or multi-scale data [39]. To figure this out, a number of methods have been proposed [40–42, 38, 43]. The state of arts spectral clustering algorithms for multi-scale data are ROSC and CAST. Details of these algorithms will be discussed in the next section.

2.1 Scaling the Similar Matrix

One of the typical methods for handling multi-scale data in spectral clustering is scaling the similar matrix S [44, 45]. Self-tuning spectral clustering (ZP method) [40] uses the Gaussian kernel based similarity $S_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ to $S_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right)$, where x is the feature vector of an object, and σ is a scaling parameter which is always difficult to set. Instead of a hyper-parameter σ , ZP introduces a local scaling parameter σ_i for each object x_i . The local scaling parameter is set as the distance from x_i to its l -th nearest neighbor (l is a hyper-parameter). Under this design, objects in a sparse cluster will have large σ and dense cluster’s σ will be relatively small.

2.2 PI Technique for Obtaining Pseudo-eigenvectors for Cluster Seperation Information

Spectral clustering algorithms always try to find eigenvectors with the richest cluster separation information [46, 47]. For most spectral clustering algorithms, the smallest eigenvectors are usually treated as the most informative ones. But in fact, researchers have already find that some of these smallest eigenvectors might contain salient noise in the data and might not contain good cluster separation information [48]. Under this circumstances, researches focusing on how to select eigenvectors with the richest cluster separation information have been raised.

The most widely used methods to generate pseudo-eigenvectors is the power iteration (PI) technique. The PI method is used to derive the dominant eigenvector of a matrix and can thus using to enhance spectral clustering.

Given a matrix W , PI will first generate a random vector $v_0 \neq 0$ and iterates as below:

$$v_{t+1} = \frac{Wv_t}{\|Wv_t\|_1}, \quad t \geq 0$$

For simplicity, we assume that W has eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_n$ with corresponding eigenvectors e_1, e_2, \dots, e_n . We express v_0 as

$$v_0 = c_1 e_1 + c_2 e_2 + \dots + c_n e_n$$

with parameters c_1, \dots, c_n . Let $R = \prod_{i=0}^{t-1} \|Wv_i\|_1$, we have,

$$\begin{aligned} v_t &= W^t v_0 / R = (c_1 W^t e_1 + c_2 W^t e_2 + \dots + c_n W^t e_n) / R \\ &= (c_1 \lambda_1^t e_1 + c_2 \lambda_2^t e_2 + \dots + c_n \lambda_n^t e_n) / R \\ &= \frac{c_1 \lambda_1^t}{R} \left[e_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^t e_2 + \dots + \frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1} \right)^t e_n \right] \end{aligned}$$

We can see that v_t is thus a linear combination of the eigenvectors. It is obviously that when $c_1 \neq 0$, v_t converges to the scaled dominant eigenvector e_1 .

By truncating the iterative PI process, we can obtain an intermediate pseudo-eigenvector, an eigenvectors' linear combination, and thus containing rich cluster separation information. With PI technique, researchers have proposed the Power Iteration Clustering (PIC) method [48]. However, each object has only one feature value corresponding to the lone pseudo-eigenvector in PIC, and this is far from enough if we have a large clusters number as the cluster collision problem [49] may happen. Under this circumstance, the PIC- k method has been further proposed to fix this issue, which directly runs PI multiple times to generate more pseudo-eigenvectors for clustering.

The most weakness of the PIC- k method is that the pseudo-eigenvectors could be similar because they are not strictly orthogonal, which will leads to redundancy. In order to reduce this redundancy, Deflation-based Power Iteration Clustering (DPIC) method [50] has been proposed to use Schur complement to derive orthogonal pseudo-eigenvectors. Besides, the PIC based methods always ignore lesser but necessary eigenvectors because dominant eigenvectors are assigned larger weights during the PI process. To figure this out, the Diverse Power Iteration Embedding (DPIE) method [51] has been proposed. In this algorithm, When we generate a new pseudo-eigenvector, previously pseudo-eigenvectors' information will be removed from the new one.

3 Algorithm

In this section, we will first define TKNN graph and grouping effect. After that, we introduce ROSC and CAST in detail. Finally, we present our algorithm and prove that similar to ROSC and CAST, the matrix Z in our algorithm also has grouping effect and it can further take data smoothness into account when clustering. It has been proved in previous work that if the coefficient matrix has grouping effect, it can capture the high correlation among objects.

3.1 Transitive K Nearest Neighbor (TKNN) Graph

TKNN graph is always used to regularize the coefficient matrix Z . The main purpose of TKNN graph is to capture the high correlations between objects. We hope that even for objects belong to the same cluster but located at distant far ends of the cluster, their correlations can still be captured.

Definition 1. (Mutual K -nearest neighbors) [9] Let $N_K(x)$ be the set of K nearest neighbors of an object x . Two objects x_i and x_j are said to be mutual K -nearest neighbors of each other, denoted by $x_i \sim x_j$, iff $x_i \in N_K(x_j)$ and $x_j \in N_K(x_i)$.

Definition 2. (Reachability) [9] Two objects x_i and x_j are said to be reachable from each other if there exists a sequence of $h \geq 2$ objects $\{x_i = x_{a_1}, \dots, x_{a_h} = x_j\}$ such that $x_{a_r} \sim x_{a_{r+1}}$ for $1 \leq r < h$.

Definition 3. (Transitive K -nearest neighbor (TKNN) graph) [9] Given a set of objects $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, the TKNN graph $\mathcal{G}_K = (\mathcal{X}, \mathcal{E})$ is an undirected graph where \mathcal{X} is the set of vertices and \mathcal{E} is the set of edges. Specifically, the edge $(x_i, x_j) \in \mathcal{E}$ iff x_i and x_j are reachable from each other. We represent the TKNN graph by an $n \times n$ reachability matrix \mathcal{W} whose (i, j) -entry $\mathcal{W}_{ij} = 1$ if $(x_i, x_j) \in \mathcal{E}$; 0 otherwise.

3.2 Grouping Effect

For an object x_p , we use z_p to denote the p -th column of matrix Z . We know from previous works [52–54] that if Z has grouping effect, then using Z for spectral clustering will have an excellent performance.

Definition 4. (Grouping Effect) [9] Given a set of objects $X = \{x_1, x_2, \dots, x_n\}$, let w_q be the q -th column of \mathcal{W} . Further, let $x_i \rightarrow x_j$ denote the condition: (1) $x_i^T x_j \rightarrow 1$ and (2) $\|w_i - w_j\|_2 \rightarrow 0$. A matrix Z is said to have grouping effect if

$$x_i \rightarrow x_j \Rightarrow |Z_{ip} - Z_{jp}| \rightarrow 0, \forall 1 \leq p \leq n$$

3.3 ROSC

The main idea of ROSC is to modify the traditional distance-based similarity matrix S to a coefficient matrix Z and then perform spectral clustering based on Z . The similarity between two objects can

be treated as how much common characteristics they share. If two objects share a lot common characteristics, they are more likely to be represented by each other. Therefore, it is quite nature to determine a coefficient matrix Z by

$$X = XZ \quad (1)$$

To construct Z , ROSC first applies PI multiple times to generate p pseudo-eigenvectors, which form $X \in \mathbb{R}^{p \times n}$. The q -th column of the matrix X is regarded as the feature vector x_q of an object x_q . The pseudo-eigenvectors can be treated as low dimensional embeddings of objects, which contains rich separation information. We know from previous works that the generated p pseudo-eigenvectors are always noisy so that ROSC represents each object by others with

$$X = XZ + O \quad (2)$$

where $O \in \mathbb{R}^{p \times n}$ is the matrix used to represent the noise in the pseudo-eigenvectors.

ROSC derived the matrix Z by optimizing the objective function

$$\min_Z \|X - XZ\|_F^2 + \alpha_1 \|Z\|_F^2 + \alpha_2 \|Z - \mathcal{W}\|_F^2 \quad (3)$$

where the first term is to reduce the noise O , the second term is the Frobenius norm of Z acting as a penalty term to guarantee the sparsity of Z and the third term regularizes Z by the TKNN graph, which means if two objects are reachable to each other, it will be treated as similar. $\alpha_1 > 0, \alpha_2 \geq 0$ are two factors using to indicate the relative weights of these three terms. We can easily derive the optimal solution to this problem and this has been proved to have grouping effect so that ROSC can work in multi-scale data:

$$Z^* = (X^T X + \alpha_1 I + \alpha_2 I)^{-1} (X^T X + \alpha_2 \mathcal{W}) \quad (4)$$

3.4 CAST

ROSC only express the high correlation among objects from the same cluster. On the other hand, suppressing the correlation among objects from different clusters is also important for correct clustering. CAST studied methods to consider both factors. The key idea of CAST is applying trace Lasso [55] to regularize the coefficient matrix. Researchers have proved that when regularized by trace lasso, the coefficient matrix exhibits "sparsity". Sparsity is the desired property that entries in the matrix corresponding to inter-cluster object pairs should be 0 or very small, thus suppressing the correlation among objects from different clusters. The optimization problem of CAST is as below:

$$\min_z \frac{1}{2} \|x - Xz\|_2^2 + \alpha_1 \|X \text{Diag}(z)\|_* + \frac{\alpha_2}{2} \|z - w\|_2^2 \quad (5)$$

where the second term of equation 5 is:

$$\|X \text{Diag}(z)\|_* = \sum_{q=1}^n \|x_q\|_2 |z_q| = \sum_{q=1}^n |z_q| = \|z\|_1 \quad (6)$$

3.5 Our Algorithm

Although ROSC and CAST use coefficient matrix Z for spectral clustering, they ignore that in most cases, data always presents in a smooth manner. In other words, some objects may be reachable to each other but with sudden direction changes, in this case they are likely to belong to different clusters.

So in our algorithm, we transform the problem into:

$$\min_Z \|X - XZ\|_F^2 + \alpha_1 \|Z\|_F^2 + \alpha_2 \|Z - \mathcal{W}\|_F^2 + \alpha_3 \|Z - \mathcal{W} * \mathcal{W} + \alpha_4 * \mathcal{W}\|_F^2 \quad (7)$$

All the notions and the first three terms above are exactly the same with those in ROSC. The forth term is a penalty term on sudden direction changes. $\mathcal{W} * \mathcal{W}$ is the second reachable matrix for objects, indicating the path numbers between objects. Intuitively, if there are many paths between two objects, then they are likely to be smooth paths between objects. As we hope this can work for multi-scale data, so a hyper-parameter α_4 is used here to indicate the standard of "many paths".

We can easily derive the optimal solution, Z^* , to the optimization problem:

$$Z^* = (X^T X + \alpha_1 I + \alpha_2 I + \alpha_3 I)^{-1} (X^T X + (\alpha_2 - \alpha_3 * \alpha_4) \mathcal{W} + \alpha_3 \mathcal{W} * \mathcal{W}) \quad (8)$$

LEMMA 1. Given a set of objects \mathcal{X} , the matrix $X \in \mathbb{R}^{p \times n}$, whose rows are pseudo-eigenvectors, the reachability matrix \mathcal{W} , then we have the the optimal soutuon Z^* to Equation 6

$$Z_{ip}^* = \frac{x_i^T (x_p - X z_p^*) + (\alpha_2 - \alpha_3 * \alpha_4) \mathcal{W}_{ip} + \alpha_3 \mathcal{W} \mathcal{W}_{ip}}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \forall 1 \leq i, p \leq n, \quad (9)$$

where $\mathcal{W} \mathcal{W} = \mathcal{W} * \mathcal{W}$

PROOF

For $1 \leq p \leq n$, let

$$J(z_p) = \|x_p - X z_p\|_2^2 + \alpha_1 \|z_p\|_2^2 + \alpha_2 \|z_p - w_p\|_2^2 + \alpha_3 \|z_p - w w_p + \alpha_4 * w_p\|_2^2. \quad (10)$$

As Z^* is the optimal solution to Equation 6, we have

$$\left. \frac{\partial J}{\partial Z_{ip}} \right|_{z_p = z_p^*} = 0, \quad \forall 1 \leq i \leq n. \quad (11)$$

Hence

$$-2x_i^T (x_p - X z_p^*) + 2\alpha_1 Z_{ip}^* + 2\alpha_2 (Z_{ip}^* - \mathcal{W}_{ip}) + 2\alpha_3 (Z_{ip}^* - \mathcal{W} \mathcal{W}_{ip} + \alpha_4 \mathcal{W}_{ip}) = 0 \quad (12)$$

, which induces Equation 7.

LEMMA 2. $\forall 1 \leq i, j, p \leq n$

$$|Z_{ip}^* - Z_{jp}^*| \leq \frac{c \sqrt{2(1-r)} + |\alpha_2 - \alpha_3 * \alpha_4| |\mathcal{W}_{ip} - \mathcal{W}_{jp}| + \alpha_3 |\mathcal{W} \mathcal{W}_{ip} - \mathcal{W} \mathcal{W}_{jp}|}{\alpha_1 + \alpha_2 + \alpha_3} \quad (13)$$

where $c = \sqrt{1 + (\alpha_2 + \alpha_3 * \alpha_4) \|w_p\|_2^2 + \alpha_3 \|w w_p\|_2^2}$ and $r = x_i^T x_j$

PROOF

From Equation 7, we have

$$Z_{ip}^* - Z_{jp}^* = \frac{(x_i^T - x_j^T) (x_p - X z_p^*) + (\alpha_2 - \alpha_3 * \alpha_4) (\mathcal{W}_{ip} - \mathcal{W}_{jp}) + \alpha_3 (\mathcal{W} \mathcal{W}_{ip} - \mathcal{W} \mathcal{W}_{jp})}{\alpha_1 + \alpha_2 + \alpha_3} \quad (14)$$

That implies

$$\begin{aligned} |Z_{ip}^* - Z_{jp}^*| &\leq \frac{|(x_i^T - x_j^T) (x_p - X z_p^*)| + |\alpha_2 - \alpha_3 * \alpha_4| |\mathcal{W}_{ip} - \mathcal{W}_{jp}| + \alpha_3 |\mathcal{W} \mathcal{W}_{ip} - \mathcal{W} \mathcal{W}_{jp}|}{\alpha_1 + \alpha_2 + \alpha_3} \\ &\leq \frac{\|x_i - x_j\|_2 \|x_p - X z_p^*\|_2 + |\alpha_2 - \alpha_3 * \alpha_4| |\mathcal{W}_{ip} - \mathcal{W}_{jp}| + \alpha_3 |\mathcal{W} \mathcal{W}_{ip} - \mathcal{W} \mathcal{W}_{jp}|}{\alpha_1 + \alpha_2 + \alpha_3} \end{aligned}$$

As the column vectors of X are normalized vectors (i.e., $x_q^T x_q = 1, \forall 1 \leq q \leq n$), we have $\|x_i - x_j\|_2 = \sqrt{2(1-r)}$, where $r = x_i^T x_j$. As Z^* is the optimal solution of Equation 5, we have

$$\begin{aligned} J(z_p^*) &= \|x_p - X z_p^*\|_2^2 + \alpha_1 \|z_p^*\|_2^2 + \alpha_2 \|z_p^* - w_p\|_2^2 + \alpha_3 \|z_p^* - w w_p + \alpha_4 * w_p\|_2^2 \\ &\leq J(0) = \|x_p\|_2^2 + (\alpha_2 + \alpha_3 * \alpha_4) \|w_p\|_2^2 + \alpha_3 \|w w_p\|_2^2 = 1 + (\alpha_2 + \alpha_3 * \alpha_4) \|w_p\|_2^2 + \alpha_3 \|w w_p\|_2^2 \end{aligned}$$

Hence, $\|x_p - X z_p^*\|_2 \leq \sqrt{1 + (\alpha_2 + \alpha_3 * \alpha_4) \|w_p\|_2^2 + \alpha_3 \|w w_p\|_2^2} = c$.

Then we get

$$|Z_{ip}^* - Z_{jp}^*| \leq \frac{c\sqrt{2(1-r)} + |\alpha_2 - \alpha_3 * \alpha_4| |W_{ip} - W_{jp}| + \alpha_3 |WW_{ip} - WW_{jp}|}{\alpha_1 + \alpha_2 + \alpha_3}$$

LEMMA 3. Matrix Z^* in our algorithm has grouping effect.

PROOF Given two objects x_i and x_j such that $x_i \rightarrow x_j$, we have (1) $x_i^T x_j \rightarrow 1$ and (2) $\|w_i - w_j\|_2 \rightarrow 0$. These imply $r = x_i^T x_j \rightarrow 1$ and $|W_{ip} - W_{jp}| \rightarrow 0$, what's more, $|WW_{ip} - WW_{jp}| \leq \|w_i - w_j\|_2 * \|w_p\|_2 \rightarrow 0$. Hence, the three terms of the numerator of Equation 11 are close to 0. Therefore, $|Z_{ip}^* - Z_{jp}^*| \rightarrow 0$ and thus Z^* has grouping effect.

We show that our algorithm also improves the performance of spectral clustering on multi-scale data. While traditional approaches focus on feature similarity and state of the art use integrated similarity of both feature and reachability, our approach further considers data smoothness without destroying the grouping effect. Two distant objects x_i and x_j belong to a same cluster may not share a strong feature similarity or even strong reachability similarity, but present a sudden change in some elements of the feature vector. This leads to a small r , and traditional approaches will likely separate them into different clusters. What's more, ROSC and CAST may also poorly performed here because such a sudden change may still preserve reachability similarity. On the contrary, our algorithm considers the smoothness of the objects' present manner and thus keeping them in the same cluster.

4 Conclusion

In this paper, we studied the performance of considering data smoothness for spectral clustering on data with various sizes and densities. We reviewed existing spectral methods in handling multi-scale data, such as ROSC and CAST. In particular, we found that both ROSC and CAST are not suitable when data is presented in a significant smooth manner. We thus proposed our algorithm, which uses a penalty term with angle information to take data smoothness into account. We mathematically proved that the matrix Z constructed by our algorithm has grouping effect. We conducted extensive experiments to evaluate our algorithm's performance and compared it against other competitors using both synthetic and real datasets. Our experimental results showed that our algorithm performed very well against its competitors over all the datasets, especially on our synthetic datasets. It is thus robust when applied to multi-scale data of different properties.

References

- [1] Pavel Kolev and Kurt Mehlhorn. A note on spectral clustering. In *ESA*, 2016.
- [2] Inderjit Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 05 2001.
- [3] Xiang Li, Ben Kao, Zhaochun Ren, and Dawei Yin. Spectral clustering in heterogeneous information networks. 04 2019.
- [4] Brahim Aamer, Hatim Chergui, Nouamane Chergui, K. Tourki, M. Benjillali, C. Verikoukis, and M. Debbah. Self-tuning spectral clustering for adaptive tracking areas design in 5g ultra-dense networks. *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–5, 2019.
- [5] Austin R. Benson, D. Gleich, and J. Leskovec. Tensor spectral clustering for partitioning higher-order network structures. *Proceedings of the ... SIAM International Conference on Data Mining. SIAM International Conference on Data Mining*, 2015:118–126, 2015.
- [6] T. Schultz and G. Kindlmann. Open-box spectral clustering: Applications to medical image analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19:2100–2108, 2013.
- [7] K. Xia, Xiaoqing Gu, and Yudong Zhang. Oriented grouping-constrained spectral clustering for medical imaging segmentation. *Multimedia Systems*, 26:27–36, 2019.

- [8] Chia-Tung Kuo, Peter B. Walker, O. Carmichael, and I. Davidson. Spectral clustering for medical imaging. *2014 IEEE International Conference on Data Mining*, pages 887–892, 2014.
- [9] Xiang Li, Ben Kao, Siqiang Luo, and Martin Ester. Rosc: Robust spectral clustering on multi-scale data. pages 157–166, 04 2018.
- [10] Xiang Li, Ben Kao, Caihua Shan, Dawei Yin, and Martin Ester. Cast: A correlation-based adaptive spectral clustering algorithm on multi-scale data, 06 2020.
- [11] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *Scholarpedia*, 3:5187, 2001.
- [12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 22(08):888–905, aug 2000.
- [13] Ling Huang, Donghui Yan, Michael I. Jordan, and Nina Taft. Spectral clustering with perturbed data. In *Proceedings of the 21st International Conference on Neural Information Processing Systems, NIPS’08*, page 705–712, Red Hook, NY, USA, 2008. Curran Associates Inc.
- [14] Ulrike Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, 36, 05 2008.
- [15] X. Zhu, C. C. Loy, and S. Gong. Constructing robust affinity graphs for spectral clustering. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1450–1457, 2014.
- [16] M. Afzalan and F. Jazizadeh. An automated spectral clustering for multi-scale data. *Neurocomputing*, 347:94–108, 2019.
- [17] Yang Zhao, Y. Yuan, F. Nie, and Q. Wang. Spectral clustering based on iterative optimization for large-scale and high-dimensional data. *Neurocomputing*, 318:227–235, 2018.
- [18] Y. Li, F. Nie, Heng Huang, and J. Huang. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, 2015.
- [19] R. Couillet and Florent Benaych-Georges. Kernel spectral clustering of large dimensional data. *arXiv: Statistics Theory*, 2015.
- [20] Xinlei Chen and Deng Cai. Large scale spectral clustering with landmark-based representation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI’11*, page 313–318. AAAI Press, 2011.
- [21] David Cheng, Ravindran Kannan, Santosh Vempala, and Grant Wang. On a recursive spectral algorithm for clustering from pairwise similarities. 07 2003.
- [22] Jane Cullum and Ralph Willoughby. Lanczos algorithms for large symmetric eigenvalue computations. vol. i: Theory. *Classics in Applied Mathematics*, I, 01 2002.
- [23] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.
- [24] Alex Gittens, Prabhanjan Kambadur, and Christos Boutsidis. Approximate spectral clustering via randomized sketching. 11 2013.
- [25] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09*, page 907–916, New York, NY, USA, 2009. Association for Computing Machinery.
- [26] Nicolas Tremblay, Gilles Puy, Pierre Borgnat, Rémi Gribonval, and Pierre Vanderghenst. Accelerated spectral clustering using graph filtering of random signals. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4094–4098, 2016.

- [27] Longcheng Zhai, Bin Wu, and Qiuyue Li. Ksc: A fast and simple spectral clustering algorithm. *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pages 381–387, 2019.
- [28] S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.
- [29] M Meila and Jue Shi. A random walks view of spectral segmentation. *aistats. AI and Statistics (AISTATS)*, 02 2001.
- [30] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. *Adv Neural Inf Process Syst*, 18, 07 2005.
- [31] Matthäus Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern. Guarantees for spectral clustering with fairness constraints. In *ICML*, 2019.
- [32] Y. Zhang and K. Rohe. Understanding regularized spectral clustering via graph conductance. *ArXiv*, abs/1806.01468, 2018.
- [33] Yang Wang and Lihua Wu. Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural networks : the official journal of the International Neural Network Society*, 103:1–8, 2018.
- [34] T. Bühler and M. Hein. Spectral clustering based on the graph p-laplacian. In *ICML ’09*, 2009.
- [35] Uri Shoham, K. Stanton, Haochao Li, B. Nadler, R. Basri, and Y. Kluger. Spectralnet: Spectral clustering using deep neural networks. *ArXiv*, abs/1801.01587, 2018.
- [36] X. Yang, Cheng Deng, Feng Zheng, Junchi Yan, and W. Liu. Deep spectral clustering using dual autoencoder network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4061–4070, 2019.
- [37] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. Suykens. Multiclass semisupervised learning based upon kernel spectral clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 26:720–733, 2015.
- [38] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
- [39] Boaz Nadler and Meirav Galun. Fundamental limitations of spectral clustering. volume 19, pages 1017–1024, 01 2006.
- [40] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. volume 17, 01 2004.
- [41] Aleksandar Bojchevski, Yves Matkovic, and Stephan Günnemann. Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17, page 737–746, New York, NY, USA, 2017. Association for Computing Machinery.
- [42] Carlos D. Correa and Peter Lindstrom. Locally-scaled spectral clustering using empty region graphs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’12, page 1330–1338, New York, NY, USA, 2012. Association for Computing Machinery.
- [43] Christopher R. John, D. Watson, M. R. Barnes, C. Pitzalis, and M. Lewis. Spectrum: fast density-aware spectral clustering for single and multi-omic data. *Bioinformatics*, 2020.
- [44] Hongjie Jia, Shifei Ding, and Mingjing Du. Self-tuning p-spectral clustering based on shared nearest neighbors. *Cognitive Computation*, 7:622–632, 2015.
- [45] M. Beauchemin. A density-based similarity matrix construction for spectral clustering. *Neurocomputing*, 151:835–844, 2015.

- [46] S. Y. Charles J. Alpert. Spectral partitioning: The more eigenvectors, the better. In *32nd Design Automation Conference*, pages 195–200, 1995.
- [47] Wei Ye, Sebastian Goebel, Claudia Plant, and Christian Böhm. Fuse: Full spectral clustering. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1985–1994, New York, NY, USA, 2016. Association for Computing Machinery.
- [48] Frank Lin and William Cohen. Power iteration clustering. pages 655–662, 08 2010.
- [49] F. Lin. Scalable methods for graph-based unsupervised and semi-supervised learning. 2012.
- [50] Anh Pham The, N. Thang, L. Vinh, Y. Lee, and S. Lee. Deflation-based power iteration clustering. *Applied Intelligence*, 39:367–385, 2012.
- [51] H. Huang, Shinjae Yoo, D. Yu, and H. Qin. Diverse power iteration embeddings and its applications. *2014 IEEE International Conference on Data Mining*, pages 200–209, 2014.
- [52] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3834–3841, 2014.
- [53] C. Lu, J. Feng, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *2013 IEEE International Conference on Computer Vision*, pages 1345–1352, 2013.
- [54] Lu Canyi, Min Hai, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. volume 7578, pages 347–360, 10 2012.
- [55] Edouard Grave, Guillaume Obozinski, and Francis Bach. Trace lasso: a trace norm regularization for correlated designs. *Advances in Neural Information Processing Systems*, 09 2011.