Two-step scalable spectral clustering algorithm using landmarks and probability density estimation

Two-step Scalable Spectral Clustering Algorithm Using Landmarks and Probability Density Estimation

Xia Hong, Junbin Gao, Hong Wei, James Xiao and Richard Mitchell

***Abstract*—Spectral clustering is one of the most important clustering approaches, often yielding performance superior to other clustering approaches. However, it is not scalable to large data sets in its original form due to the computational burden of the required large-matrix eigen-decomposition. In this paper, a two-step spectral clustering algorithm is introduced by extending recent advances of scalable spectral clustering based on low-rank affinity matrix using landmarks. In the first step, a scalable spectral clustering algorithm using raw landmark-based affinity matrix is adopted. In the second step, a novel low-rank affinity matrix is learnt via the probability density estimators, constructed from the estimated clusters as derived from the first step. Since the prior information on cluster labels can be utilised in the second step, this learnt affinity matrix reflects intrinsic pairwise data relationships much better. While the proposed more complicated algorithm results in a higher computational cost than the previous landmark-based spectral research, it can be shown that the associated computational cost still scales well with data size. It is demonstrated that the proposed algorithm is capable of achieving far superior performance than other state-of-the-art algorithms for several benchmark multi-class image data sets.**

***Index Terms*—Spectral clustering, probability density estimation, low-rank matrix, landmarks.**

I. Introduction

Clustering is one of the most important research topics in machine learning, computer vision, data mining and various scientific applications [1], [2], [3], [4], [5]. When dealing with empirical data, people often get a first impression on their data by trying to identify groups of "similar behavior". Data clustering is an unsupervised classification paradigm which divides observed data into different subsets (clusters), such that similar objects are allocated to the same subset while dissimilar objects are assigned to different subsets. Most recent advances include: clustering of data with temporal effects [6], 3D point clouds [7] and explainable clustering approaches using neural networks [8].

Spectral clustering techniques [9], [10], [11] have been shown to be among the most effective clustering algorithms, due to their ability to work with nonlinear separable problems [10]. The effectiveness can be explained by considering that the data in the original space are mapped onto a new embedded space where patterns of similar points are built on the basis of pairwise similarity of objects to be grouped. The embedding space is spanned by the eigenvectors of the Laplacian matrix, which is derived from the graph affinity matrix in graph theory. The eigenvalues and eigenvectors of a suitably chosen affinity matrix are calculated to partition the data [12], [13]. Then the K principal eigenvectors are used for clustering the original data, with structural properties of the data set being identified correctly for a block diagonal matrix [10].

Recent developments in sensors, data-storage and data-acquisition devices have made large data-sets widely available. Spectral clustering is a flexible class of clustering algorithms that can produce high-quality clustering on small data sets. Yet it is known that it suffers from a high computational cost due to its computational complexity of $O(N^3)$ for the eigenvectors computation of its graph based affinity matrix, where N is the data size. Hence spectral clustering is not widely viewed as a competitor to classical algorithms such as hierarchical clustering and $k-$means [1] for large-scale data mining problems, and spectral algorithms, in their original form, have limited applicability to large-scale problems.

There is some interesting work on scalable spectral clustering and other associated recent work related to spectral clustering (see Section II). Notably Cai and Chen [14] proposed an affinity matrix using a set of random sampled data points as landmarks, so that the spectral embedding can be efficiently calculated via the landmark-based representation. Recently, the notion of scalable spectral clustering with cosine similarity has been proposed which also leads to computational efficiency by exploiting the properties of low-rank matrices [15].

Of particular interest to this work is the design of a low-rank affinity matrix directly from the data in order to achieve a drastic reduction in computational complexity of spectral clustering while still maintaining high performance. In this paper, a two-step spectral clustering algorithm is introduced, which is inspired by the recent advances of scalable spectral clustering based on low-rank affinity matrices using landmarks [14]. Specifically, the first step is a scalable spectral clustering algorithm using raw landmark-based affinity matrix. Since the effectiveness of spectral clustering depends on the affinity function between each pair of data objects, it is vital to construct a weight matrix that faithfully reflects the similarity information among objects. While the landmark-based affinity matrix is useful in achieving scalable computation, this is constructed in the original high-dimensional data input space.

Xia Hong, Hong Wei and Richard Mitchell are with the School of Mathematical, Physical and Computational Sciences, University of Reading, Reading, RG6 6AY, UK. (x.hong/h.wei/r.j.mitchell@reading.ac.uk)

Junbin Gao is with Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia, (junbin.gao@sydney.edu.au)

James Xiao is an independent researcher, Victoria, BC, Canada, zifan.james.xiao@gmail.com.

Here, the proposed method includes a new second step in which a low-rank affinity matrix is learnt from the probability density estimators, constructed from the estimated clusters given in the first step. Since the prior information on cluster labels can be utilised in the second stage, the affinity matrix in second step should reflect intrinsic pairwise data relationships much better. While the proposed algorithm results in a more complicated algorithm with a slightly higher computational cost than the previous landmark-based spectral research [15], it has been shown that the associated computational cost still scales well with data size. It is demonstrated that the proposed algorithm is capable of achieving far superior performances than other state-of-the-art algorithms for several benchmark multi-class image data sets.

The novel contributions of the paper are as follows:

1) A new scalable spectral clustering algorithm is proposed which is based on low-rank matrices using landmarks and avoids handling the affinity matrix directly.
2) The proposed algorithm is iterative by incorporating the idea of probability density estimators to learn the affinity function in the second step, i.e. the proposed affinity matrix can learn the data class associations from the conventional landmark-based spectral clustering in the first step.
3) The algorithm can also be used for semi-supervised problems in which there exists a limited number of labelled data points.
4) As shown in experiments given here, the proposed algorithm significantly outperforms other state-of-the-art algorithms in terms of clustering performance with higher yet still scalable computational costs.

The paper is organised as follows: Section II reviews the related research in spectral clustering followed by Section III which introduces the basics of spectral clustering and in particular low-rank matrix based scalable spectral clustering. Section IV introduces the proposed two-step spectral clustering algorithm using landmarks and probability density estimation, followed by some discussions to analyse the contributions, rationale and computational complexities of the proposed approach. Section V compares the proposed two-step spectral clustering algorithm with a number of state-of-the-art methods with superior results over a few benchmark data sets. The experimental results of varying parameter settings are included to provide insights. Finally, a computer vision image segmentation example is provided to illustrate the usefulness of the approach.

II. Related Work

In recent decades, considerable efforts have been devoted to expand spectral clustering modelling paradigms and associated optimization algorithms. Some representative works are classified based on their main characteristics.

- *Scalable spectral algorithms*: The Nyström method [16], [17] is an efficient technique to generate low-rank matrix approximation by sampling a subset of the columns of the affinity matrix [18]. A similar technique has been applied in clustering large scale social networks data via a pre-coarsening sampling based on Nyström method [19]. The majority of these algorithms are based on "sampling and approximation", which unavoidably lose information due to data reduction, although researchers work towards improving the performance of such sampling and approximation approaches [20]. Cai and Chen [14] has introduced the idea of landmarks to achieve low rank affinity matrices, by which this work is inspired.
- *Learning affinity:* Bach and Jordan [21] first took a point of view of learning affinity. The works of [22], [23] combine techniques of matrix factorization [22] and sparse coding [23]. The notion of local scaling between each pair of points' affinity was introduced in [24] and subsequently a self-tuning mechanism for scaling parameters is introduced. Ding *et al.* proved the equivalence between spectral clustering and matrix factorization [25]. Spectral clustering with sparse coding has been found to be effective for clustering high-dimensional data [11]. These are effective in finding a suitable sparse representation base on which the affinity matrix is constructed [26]. Along this line, learning affinity has been further explored under the framework of deep learning [27], [28], [29], [30]. Learning affinity is linked with metric learning [31], thus it opens door for utilising metric learning-based clustering.
- *Modified spectral clustering models with dimensionality reduction*: Most recent works are well motivated to address challenges in high-dimensional data sets, and may also be related to learning affinity and scalability. Incorporating dimensional reduction in spectral clustering, these works are combined with other active areas such as multi-view learning, (see [5] and references within). It is therefore necessary to modify the objective functions and develop new optimization methods for extended family of spectral clustering. The one-step multi-view spectral clustering (OMSC) method leads also to scalable computational costs at $O(N)$ [40]. Similarly, Spectral Rotation for One-step Spectral Clustering (SR-OSC) has been introduced with a new composite objective function so that several layers of learning in the new model are jointly optimized [40] with $O(N^2)$.

III. Preliminaries

This section introduces the main concepts of scalable spectral clustering with the aim of achieving computational efficiency for large-sized data. The works of scalable spectral clustering exploiting the properties of low-rank affinity matrices are reviewed in particular.

A. Spectral clustering

Let $\{\boldsymbol{x}_i\}_{i=1}^N$, each $\boldsymbol{x}_i = [x_{i,1}, ..., x_{i,d}]^{\mathrm{T}} \in \Re^d$, be a given data set, and $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^{\mathrm{T}}$. The goal of clustering is to partition these points into K disjoint sets, for some given K. The spectral clustering is based on a weighted undirected graph $G(V, E)$, where each of the vertices in G corresponds to a data point $\boldsymbol{x}_i$, and the weight of each edge $w_{i,j} > 0$ encodes the similarity between a distinctive data pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$.

Algorithm 1 The normalised spectral clustering algorithm of [10].

Require: Similarity matrix $\boldsymbol{W} \in \Re^{N \times N}$, number K of clusters to construct.

1: Compute the normalised weighted matrix $\tilde{\boldsymbol{W}}$ according to (3).
2: Perform the singular value decomposition (SVD) of $\tilde{\boldsymbol{W}}$ as $\tilde{\boldsymbol{W}} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\mathrm{T}}$, where $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_N]$ is the orthonormal matrix, consisting of eigenvectors of $\boldsymbol{L}_{sym}$. $\Lambda = \mathrm{diag}\{\lambda_1, ..., \lambda_N\}$, $\lambda_1 > ... > \lambda_N >= 0$ are eigenvalues.
3: Let $\boldsymbol{U}_K \in \Re^{N \times K}$ be the sub-matrix of $\boldsymbol{U}$ of the first K eigenvectors as $\boldsymbol{U}_k = [\boldsymbol{u}_1, ..., \boldsymbol{u}_K]$.
4: Form the matrix $\boldsymbol{T} = \{t_{i,j}\} \in \Re^{N \times K}$ by normalising the rows to norm one, i.e. to set

$$t_{i,j} = u_{i,j} \Big/ \sqrt{\sum_{j=1}^{K} u_{i,j}^2}. \tag{4}$$

5: **for** $i = 1, ..., N$ **do**
6: Let $\boldsymbol{y}_i \in \Re^K$ be the vector corresponding to the i-th row of $\boldsymbol{T}$.
7: **end for**
8: Cluster the points $\boldsymbol{y}_i$, $i = 1, ..., N$ with the k-means algorithm [1] into clusters $C_1, ..., C_K$.
9: Return: Find clusters $k \in \{1, ..., K\}$ with $\{k, \boldsymbol{y}_i \in C_k\}$ and assign original data points $\boldsymbol{x}_i$ according to clusters index set of $k = 1, ..., K$.

It is often assumed $w_{i,i} = 0$, $\forall i$. Regardless of the similarity function used, a weighted graph G is induced with an affinity matrix $\boldsymbol{W} = [w_{i,j}] \in \Re^{N \times N}$. The degree matrix $\boldsymbol{D}$ of G is calculated as

$$\boldsymbol{D} = \mathrm{diag}(\boldsymbol{W}\boldsymbol{1}), \tag{1}$$

where $\boldsymbol{1}$ denotes a vector of all ones. The graph Laplacian of G is defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}. \tag{2}$$

A normalised version [9] is defined as

$$\boldsymbol{L}_{sym} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \tilde{\boldsymbol{W}}, \tag{3}$$

where $\tilde{\boldsymbol{W}}$ is called normalised affinity matrix. The normalised spectral clustering algorithm of [9] is given as [10], presented in Algorithm 1.

It is known that the spectral clustering suffers from a high computational cost associated with $\boldsymbol{W}$ for large N. The general spectral clustering algorithm needs to construct an affinity matrix, followed by its eigen-decomposition which has computational cost at $O(N^3)$, it is thus of limited use for large data sets without additional treatments.

Research efforts have been devoted to develop fast, approximate algorithms in order to handle large-scale data sets. One line of research is sample based, which allows extrapolation of the complete grouping solution using only a small number of samples, e.g. [14] in which the Nyström method is employed for spectral grouping, and [32] in which the power method was explored. However their accuracy has been criticised in a recent research [33]. Other research includes sparse coding which are based on matrix factorisation techniques [34].

B. Landmark based sparse coding (LSC) [14]

Consider a nonlinear functional mapping $\mathcal{F} : \boldsymbol{x} \in \Re^d \rightarrow \boldsymbol{z} \in \Re^q$, then, from the given data set $\boldsymbol{X}$, $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_N]^{\mathrm{T}} \in \Re^{N \times q}$ can be constructed with $\boldsymbol{z}_i = [z_{i,1}, ..., z_{i,q}]^{\mathrm{T}} \in \Re^q$ obtained via $\mathcal{F}$ from $\boldsymbol{x}_i \in \Re^d$. The low-rank landmark-based affinity matrix has been designed [14] with the form of

$$\boldsymbol{W} = \boldsymbol{Z}\tilde{\boldsymbol{D}}^{-1}\boldsymbol{Z}^{\mathrm{T}}, \tag{5}$$

where $\tilde{\boldsymbol{D}} = \mathrm{diag}\{\sum_i z_{i,1}, ..., \sum_i z_{i,q}\} \in \Re^{q \times q}$, i.e., the diagonal elements of $\tilde{\boldsymbol{D}}$ is the column sum of $\boldsymbol{Z}$.

Definition: Landmarks [35]

It is proposed that $\boldsymbol{Z}$ is based on a set of $q \ll N$ representative landmark points $\boldsymbol{c}_j$, $j = 1, ..., q$ which are randomly selected from the data set or k-means clustering algorithm, so that [35]

$$z_{i,j} = \frac{\mathcal{K}_h(\boldsymbol{x}_i, \boldsymbol{c}_j)}{\sum_{j' \in <i>} \mathcal{K}_h(\boldsymbol{x}_i, \boldsymbol{c}_{j'})}, \tag{6}$$

r where $\mathcal{K}_h(.,.)$ is a kernel function in the form of

$$\mathcal{K}_h(\boldsymbol{x}_i, \boldsymbol{c}_j) = \exp\{-\|\boldsymbol{x}_i - \boldsymbol{c}_j\|^2/(2h^2)\}, \tag{7}$$

where h is a preset kernel width and $< i >$ denotes the set of r nearest landmarks to $\boldsymbol{x}_i$.

The calculation of landmarks is equivalent to computing

$$z_{i,j} = \mathcal{K}_h(\boldsymbol{x}_i, \boldsymbol{c}_j), \tag{8}$$

followed by row sorting, and for any landmarks that are further than the rth landmark, it is set $z_{i,j} = 0$, then each row is normalised by its sum. Hence $\boldsymbol{Z}$ is sparse and row-normalised. It can be verified that the resultant degree matrix $\boldsymbol{D}$ is an identity matrix [14], so that affinity matrix $\boldsymbol{W}$ is automatically normalised.

Denote

$$\boldsymbol{W} = \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^{\mathrm{T}} \tag{9}$$

with $\tilde{\boldsymbol{Z}} = \boldsymbol{Z}\tilde{\boldsymbol{D}}^{-1/2}$. Without directly dealing with $\boldsymbol{W}$, let the thin SVD of $\tilde{\boldsymbol{Z}}$ be $\tilde{\boldsymbol{Z}} = \boldsymbol{U}_{\tilde{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\tilde{\boldsymbol{Z}}}\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}^{\mathrm{T}}$, where $\boldsymbol{U}_{\tilde{\boldsymbol{Z}}} \in \Re^{N \times q}$, and $\boldsymbol{U}_{\tilde{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{U}_{\tilde{\boldsymbol{Z}}} = \boldsymbol{I}$, $\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{V}_{\tilde{\boldsymbol{Z}}} = \boldsymbol{I}$. $\boldsymbol{\Lambda}_{\tilde{\boldsymbol{Z}}} = \mathrm{diag}\{\lambda_1, ..., \lambda_q\}$, $\lambda_1 \geq ...\lambda_q \geq 0$ are singular values. Thus

$$\boldsymbol{U}_{\tilde{\boldsymbol{Z}}} = \tilde{\boldsymbol{Z}}\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\tilde{\boldsymbol{Z}}}^{-1}. \tag{10}$$

Alternatively it can be also shown that the eigen-decomposition of $\tilde{\boldsymbol{Z}}^{\mathrm{T}}\tilde{\boldsymbol{Z}} = \boldsymbol{V}_{\tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{\Lambda}}_{\tilde{\boldsymbol{Z}}}\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}^{\mathrm{T}}$, $\tilde{\boldsymbol{\Lambda}}_{\tilde{\boldsymbol{Z}}} = \mathrm{diag}\{\lambda_1^2, ..., \lambda_q^2\}$, $\lambda_1 \geq ...\lambda_q \geq 0$ are eigenvalues. This

Algorithm 2 Landmark-based sparse coding (LSC) [14].

Require: N data points $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \in \Re^d$, number K of clusters to construct.

1: Produce q landmark points using $k-$means (LSC-K) or random selection (LSC-R).
2: Construct a sparse affinity matrix $\boldsymbol{Z}$, with affinity calculated according to (6).
3: Compute the first K eigenvectors and eigenvalues $\tilde{\boldsymbol{Z}}^{\mathrm{T}}\tilde{\boldsymbol{Z}}$ denoted by $\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}\tilde{\boldsymbol{\Lambda}}_{\tilde{\boldsymbol{Z}}}\boldsymbol{V}_{\tilde{\boldsymbol{Z}}}^{\mathrm{T}}$, where $\tilde{\boldsymbol{Z}} = \boldsymbol{Z}\tilde{\boldsymbol{D}}^{-1/2}$.
4: Compute $\boldsymbol{U}_{\tilde{\boldsymbol{Z}}}$ using (10).
5: Form the matrix $\boldsymbol{T} = \{t_{i,j}\} \in \Re^{N\times K}$ by normalising the rows to norm one, i.e. to set

$$t_{i,j} = u_{i,j} \Big/ \sqrt{\sum_{k=1}^{K} u_{i,j}^2}. \tag{11}$$

6: **for** $i = 1, ..., N$ **do**
7: Let $\boldsymbol{y}_i \in \Re^K$ be the vector corresponding to the i-th row of $\boldsymbol{T}$.
8: **end for**
9: Cluster the points $\boldsymbol{y}_i$, $i = 1, ..., N$ with the $k-$means algorithm into clusters $C_1, ..., C_K$.
10: Return: Find clusters $k \in \{1, ..., K\}$ with $\{k, \boldsymbol{y}_i \in C_k\}$ and assign original data points $\boldsymbol{x}_i$ according to clusters index set of $k = 1, ..., K$.

shows that instead of directly conducting SVD for $\tilde{\boldsymbol{Z}}$, an eigen-decomposition can be applied to the $\tilde{\boldsymbol{Z}}^{\mathrm{T}}\tilde{\boldsymbol{Z}} \in \Re^{q\times q}$, which only has cost of $O(q^3)$. The results are used in calculating $\boldsymbol{U}_{\tilde{\boldsymbol{Z}}}$ according to (10). Further computational savings can be achieved by only computing the first K eigenvectors.

Note that to control computational complexity in LSC, only a small subset of the input data set are used as landmarks, which intermediate a pair of data points for the construction of their affinity function. There is always a question if a limited number of landmarks can sufficiently represent information of a full large data set. A large number of landmarks tend to produce better clustering results, but the computational costs will increase at the rate of $O(q^2N)$. Although k-means clustering, as in LSC-K, may be used for landmark selection for improved clustering, it is still an issue that they may not suit clusters of non-convex geometries and complex shapes.

The steps of LSC is summarised in Algorithm 2. Clearly, computational efficiency of LSC has benefited from using a low-rank affinity matrix with nonzero diagonals.

C. Scalable spectral clustering with cosine similarity [15]

In the following, the recent work [15] is outlined which is based on the idea of construction low rank affinity matrix using cosine similarity. Consider a low rank affinity matrix in the form of [15]

$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} - \boldsymbol{I}. \tag{12}$$

It is assumed that N is large and either $d \ll N$ or $\boldsymbol{X}$ is sparse. Each row of $\boldsymbol{X}$ is normalised. One example of $\boldsymbol{X}$ is the document term matrix. This yields to the cosine similarity-based affinity matrix, which is slightly restrictive but still has important applications such as document clustering. In comparison to (5), the affinity matrix with zero diagonals is enforced in (12).

It is shown that efficiency can be achieved by avoiding computing $\boldsymbol{W}$ explicitly [15]. Specifically for $\boldsymbol{D}$, we have

$$\boldsymbol{D} = \mathrm{diag}\big(\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{1}) - \boldsymbol{1}\big). \tag{13}$$

Moreover, representing $\tilde{\boldsymbol{W}}$ [15]

$$\tilde{\boldsymbol{W}} = \boldsymbol{D}^{-1/2}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{D}^{-1/2} - \boldsymbol{D}^{-1} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\mathrm{T}} - \boldsymbol{D}^{-1}, \tag{14}$$

where $\tilde{\boldsymbol{X}} = \boldsymbol{D}^{-1/2}\boldsymbol{X}$.

The idea in [15] is to use the first K-th left singular vectors of $\tilde{\boldsymbol{X}}$ [15] as approximation of the first K eigenvectors of $\tilde{\boldsymbol{W}}$ under the assumption that $\boldsymbol{D}^{-1} \approx \frac{1}{\beta}\boldsymbol{I}$ for some $\beta > 0$, which is obviously untrue. This means $\boldsymbol{D}^{-1}$ is totally dropped regardless their significance/influence to the spectral clustering results. To have a better result of approximation, $\boldsymbol{D}^{-1}$ should have a low conditional number. For this purpose, an outlier set denoted as C_0 is created which is removed from data points in $\boldsymbol{Z}$, if they correspond to the set of the smallest diagonal elements in $\boldsymbol{D}^{-1}$. Then the left singular vectors of $\tilde{\boldsymbol{X}}$ (size reduced) are calculated via the efficient low-rank SVD algorithm. In this situation, the size of outliers C_0 can be a large number since there are many redundancies in the data. Removing these data points from SVD calculation enhances robustness as well as greatly reduces computation costs. The idea of generating an outlier set C_0 is a good one as it improves numerical stability and robustness. Note that the computation procedure is made simpler in [15] in which the eigenvector of the affinity matrix in spectral clustering was approximated as top-K left singular vectors of $\tilde{\boldsymbol{X}}$.

In our proposed algorithm as follows, it is shown that the computational efficiency does not need be compromised too much if the affinity matrix with zero diagonals is enforced without any approximation as used in [15]. In our proposed method, the term of zeroing diagonals can be fully taken into account efficiently, removing any discrepancy issues.

IV. The proposed two-step LSC algorithm using probability density estimation

In the following, a novel two-step iterative landmark-based spectral clustering algorithm, based on a novel composite LSC (cLSC) affinity matrix, is proposed. Conventional spectral clustering algorithms often start with a given affinity function which defines pairwise relationship of data points in the *input*

space only. In contrast, the affinity function employed in the second step aims to capture the pairwise relationships of data points in *both input and output spaces*, which can help to improve clustering performance.

In Section IV-A, we initially introduce a modified LSC (mLSC) which rectifies the LSC algorithm so that diagonals of affinity matrix are zeros. This mathematical property also happens in the second step.

The main contribution of the proposed two-step method is the new second step in Section IV-B, in which a low-rank affinity matrix is learnt from the probability density estimators from the estimated clusters from mLSC.

A. A modified LSC (mLSC) based on two stages of SVD

The initial step of the proposed algorithm is based on a modification of LSC (see Section III-B [14]) employing two stages of singular value decomposition which can rectify the landmark based affinity matrix so that the convention of an affinity matrix diagonals $w_{i,i} = 0$ is satisfied. A computationally efficient algorithm, referred to as mLSC, is introduced below based on two stages of singular value decomposition.

We form $\tilde{\boldsymbol{Z}} = [\tilde{\boldsymbol{z}}_1, ..., \tilde{\boldsymbol{z}}_N]^{\mathrm{T}}$ as described in LSC in Section III-B [14], except for $\boldsymbol{W}$ which is modified as

$$\boldsymbol{W} = \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^{\mathrm{T}} - \boldsymbol{A}, \tag{15}$$

where $\boldsymbol{A} = \mathrm{diag}\{a_1, ..., a_N\}$, and

$$a_i = \tilde{\boldsymbol{z}}_i^{\mathrm{T}}\tilde{\boldsymbol{z}}_i \tag{16}$$

is a diagonal entry of $\boldsymbol{Z}\tilde{\boldsymbol{D}}^{-1}\boldsymbol{Z}^{\mathrm{T}}$. The modified LSC (mLSC) affinity matrix is no longer normalised and it can be easily verified that associated degree matrix is

$$\boldsymbol{D} = \boldsymbol{I} - \boldsymbol{A} = \mathrm{diag}\{1 - a_1, ..., 1 - a_N\}. \tag{17}$$

The normalised affinity matrix for modified LSC (mLSC) is

$$\tilde{\boldsymbol{W}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{D}^{-1/2} - \boldsymbol{D}^{-1}\boldsymbol{A} = \bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^{\mathrm{T}} - \boldsymbol{A}_{mLSC}, \tag{18}$$

where $\bar{\boldsymbol{Z}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}$, and the ith element of the diagonal matrix $\boldsymbol{A}_{mLSC}$ can be calculated as $a_i/(1 - a_i)$.

In order to exactly compute top-K eigenvectors of $\tilde{\boldsymbol{W}}$ without directly handling $\tilde{\boldsymbol{W}}$, consider initially calculating thin SVD (the first stage) $\bar{\boldsymbol{Z}} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}$, where $\boldsymbol{U}_{\bar{\boldsymbol{Z}}} \in \Re^{N\times q}$ is the left singular vector matrix, and $\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} = \boldsymbol{I}$, $\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}} = \boldsymbol{I}$, $\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}} = \mathrm{diag}\{\lambda_1, ..., \lambda_q\}$, $\lambda_1 > ... > \lambda_q >= 0$ are singular values of $\bar{\boldsymbol{Z}}$.

We have

$$\begin{aligned}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} &= \boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\bar{\boldsymbol{Z}}(\bar{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}) - \boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{A}_{mLSC}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} \\ &= \tilde{\boldsymbol{\Lambda}}_{\bar{\boldsymbol{Z}}} - \boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{A}_{mLSC}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} = \boldsymbol{B}_{mLSC} \in \Re^{q\times q},\end{aligned} \tag{19}$$

where $\tilde{\boldsymbol{\Lambda}}_{\bar{\boldsymbol{Z}}} = \boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}^2$. The computation cost of obtaining $\boldsymbol{B}_{mLSC}$ is $O(Nq^2)$.

The second stage of SVD is for $\boldsymbol{B}_{mLSC} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}$, so that

$$\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}} \tag{20}$$

or

$$\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}} = \boldsymbol{S}_{\boldsymbol{B}}. \tag{21}$$

Let $\boldsymbol{U}_K \in \Re^{N\times k}$ be the sub-matrix of $\boldsymbol{U} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}} \in \Re^{N\times q}$, we have $\boldsymbol{U}_K^{\mathrm{T}}\boldsymbol{U}_K = \boldsymbol{I}$. Spectral clustering will be carried based on normalised eigenvectors matrix $\boldsymbol{U}_K$.

The steps of mLSC is summarized in Algorithm 3.

The computational cost of obtaining $\boldsymbol{U}$ is $O(Nq^2)$. The computational cost of the first SVD is $O(q^3 + q^2N)$, if the fast approach in LSC is adopted [14]. The second SVD costs $O(q^3)$. The total computation cost for this step is higher than that of LSC, but still at the rate $O(N)$ for $q \ll N$, and large N.

B. The proposed spectral clustering algorithm using probability density estimation

In the second step, the proposed algorithm goes beyond using raw landmarks to define data relationships, but this is learnt from the clusters from the first step (mLSC). Specifically in order to improve clustering results, the probability density estimation of the clusters are exploited with the aim of uncovering the pairwise relationship of the data in the output spaces. Note that in the first step of mLSC, there were no cluster labels available so the original affinity matrix has to be defined using input features alone. However the condition changes after spectral clustering algorithm has been applied. Indeed, if there exist clusters, then the spectral clustering algorithm will identify these to a certain degree of confidence. Hence in the second step of the proposed algorithm, the affinity matrix can be adjusted based on the assumption that there exists some prior knowledge about the probability distribution about the clusters. The basic idea behind our proposed approach is to successively apply spectral clustering based on a new low-rank affinity matrix which is iteratively generated as a composite of probability density function estimation and a new set of q landmarks in the second step.

1) Probability density estimation: Consider the unsupervised setting whereby the initial clusters need to be generated at first. Specifically, a set of q representative landmark points $\boldsymbol{c}_j$, $j = 1, ..., q$ is randomly selected from the data set. Then $z_{i,j}$ is formed based on (6). Then Algorithm 3 is initially applied to find K clusters, and this produces K clusters C_k, $k = 1, ..., K$. We label a set of randomly drawn data samples from these clusters C_k as $\tilde{\boldsymbol{x}}_{i,C_k}$, $i = 1, ..., N_q$, where $N_q < q \ll N$ is a predetermined number in order to control computational cost. It is also assumed that N_q is smaller than the number of the data points in each cluster in first step.

After the first step of mLCS is applied (Algorithm 3), the Parzen window probability density function (PDF) estimator [36] for each class (cluster) can be written as

$$\begin{aligned}f_{C_k}(\boldsymbol{x}) = &\frac{1}{N_q(2\pi)^{d/2}\prod\sigma_{k,j}} \\ &\times \sum_{i=1}^{N_q}\exp\{-\frac{1}{2}(\boldsymbol{x} - \tilde{\boldsymbol{x}}_{i,C_k})^{\mathrm{T}}\Sigma_k^{-1}(\boldsymbol{x} - \tilde{\boldsymbol{x}}_{i,C_k})\},\end{aligned} \tag{23}$$

for $k = 1, ..., K$, where $\Sigma_k = \mathrm{diag}\{\sigma_{k,1}^2, ..., \sigma_{k,d}^2\}$, $\sigma_{k,j}$ is called the bandwidth, which can be set using "Scott's rule of

Algorithm 3 The proposed modified landmark-based sparse coding (mLSC).

Require: N data points $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \in \Re^d$, number K of clusters to construct.

Produce q landmark points using random selection.
Construct a sparse affinity matrix $\boldsymbol{Z}$, with affinity calculated according to (6).
Find sparse affinity matrices $\tilde{\boldsymbol{Z}} = \boldsymbol{Z}\bar{\boldsymbol{D}}^{-1/2}$, the degree matrix $\boldsymbol{D}$ using (17) and $\bar{\boldsymbol{Z}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}$.
Compute $\boldsymbol{U}_{\bar{\boldsymbol{Z}}}$ using SVD of $\bar{\boldsymbol{Z}} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}$.
Find $\boldsymbol{A}_{mLSC}$, $\boldsymbol{B}_{mLSC}$ using (18) and (19).
Perform the singular value decomposition (SVD) of $\boldsymbol{B}_{mLSC} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}$
Calculate $\boldsymbol{U} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}}$.
Form the matrix $\boldsymbol{T} = \{t_{i,j}\} \in \Re^{N\times K}$ by normalising the rows to norm one, i.e. to set

$$t_{i,j} = u_{i,j} \Big/ \sqrt{\sum_{k=1}^{K} u_{i,j}^2}. \tag{22}$$

for $i = 1, ..., N$ **do**
  Let $\boldsymbol{y}_i \in \Re^K$ be the vector corresponding to the i-th row of $\boldsymbol{T}$.
end for
Cluster the points $\boldsymbol{y}_i$, $i = 1, ..., N$ with the k-means algorithm into clusters $C_1, ..., C_K$.
Return: Find clusters $k \in \{1, ..., K\}$ with $\{k, \boldsymbol{y}_i \in C_k\}$ and assign original data points $\boldsymbol{x}_i$ according to clusters index set of $k = 1, ..., K$.

thumb" [37] that minimises the mean integrated squared error to true, unknown density, as

$$\sigma_{k,j} \approx S_{k,j} N_q^{-1/(d+4)}, \tag{24}$$

in which $S_{k,j}$ is the standard deviation of samples in the jth feature of cluster k. Note that the "Scott's rule of thumb" bandwidth estimator could provide the fastest way implementing Parzen window PDF estimator of reasonable confidence for low-dimensionality data. However since it originates from a strong assumption about data, it can be unreliable for high-dimensional data sets. Hence we adopt the following heuristics in this work:

$$\sigma_{k,j} = \sigma_j \approx \max\{\frac{1}{d}\sum_{j=1}^{d} S_{k,j} N_q^{-1/(d+4)}, \sigma_{pre}\} \tag{25}$$

which uses a shared variance for all features, and σ_{pre} is a pre-determined parameter that is empirically found for numerical stability.

However, the above PDF formula can still cause numerical problems for high-dimensional data sets due to the fact that $\prod \sigma_{k,j}$ tends to either too large or too small as d increases. Since it is reasonable to assume that this term should be in the same scale for all classes, the following scaled quantity is employed

$$p_{C_k}(\boldsymbol{x}_i) = \frac{1}{N_q}\sum_{i=1}^{N_q} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_{i,C_k})^{\mathrm{T}}\Sigma_k^{-1}(\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_{i,C_k}))\}, \tag{26}$$

for $k = 1, ..., K$. Consider a nonlinear and probabilistic functional mapping $\mathcal{P} : \boldsymbol{x} \in \Re^d \to \boldsymbol{p} \in \Re^K$, so that from the given data set $\boldsymbol{X}$, $\boldsymbol{P} = [\boldsymbol{p}_1, ..., \boldsymbol{p}_N]^{\mathrm{T}}$ can be constructed with $\boldsymbol{p}_i \in \Re^K$ obtained via $\mathcal{P}$ from $\boldsymbol{x}_i \in \Re^d$. We propose that $\boldsymbol{P} = \{p_{i,j}\} \in \Re^{N\times K}$ is based on a set of K normalised Parzen window density estimators

$$p_{i,j} = \frac{p_{C_j}(\boldsymbol{x}_i)}{\sum_{k=1}^{K} p_{C_k}(\boldsymbol{x}_i)}, \tag{27}$$

hence $\boldsymbol{P}$ is row normalised. Define

$$\tilde{\boldsymbol{P}} = \boldsymbol{P}\bar{\boldsymbol{D}}^{-1/2}, \tag{28}$$

where $\bar{\boldsymbol{D}} = \mathrm{diag}\{\sum_i p_{i,1}, ..., \sum_i p_{i,K}\} \in \Re^{K\times K}$. Denote $\tilde{\boldsymbol{P}} = [\tilde{\boldsymbol{p}}_1, ..., \tilde{\boldsymbol{p}}_N]^{\mathrm{T}} \in \Re^{N\times K}$.

The computational cost of constructing $\tilde{\boldsymbol{P}}$ is higher than that of $\tilde{\boldsymbol{Z}}$ in spite of $K \ll q$, since each entry involves N_q terms of Gaussian functions instead of only one term in $\tilde{\boldsymbol{Z}}$. Hence the computational cost is in the order of $O(KN_qN)$ versus $O(qN)$.

2) The proposed low rank composite affinity matrix: In order to improve clustering results, more landmarks are sampled in the Second step from $\boldsymbol{X}$, with the computational cost also controlled via a reasonably low value of q. Our proposed algorithm is based on $\boldsymbol{P}$ which contains information from clustering using landmarks in the previous step, as well as new data samples for current iteration step which builds $\tilde{\boldsymbol{Z}} = [\tilde{\boldsymbol{z}}_1, ..., \tilde{\boldsymbol{z}}_N]^{\mathrm{T}}$ as described in LSC. Our proposed $\boldsymbol{W}$ is a composite of landmarks and current PDFs, defined as:

$$\boldsymbol{W} = \gamma\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^{\mathrm{T}} + (1-\gamma)\tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^{\mathrm{T}} - \bar{\boldsymbol{A}}, \tag{29}$$

where $0 < \gamma < 1$ is a preset parameter. $\bar{\boldsymbol{A}} = \mathrm{diag}\{\bar{a}_1, ..., \bar{a}_N\}$, where

$$\bar{a}_i = \gamma\tilde{\boldsymbol{z}}_i^{\mathrm{T}}\tilde{\boldsymbol{z}}_i + (1-\gamma)\tilde{\boldsymbol{p}}_i^{\mathrm{T}}\tilde{\boldsymbol{p}}_i \tag{30}$$

are diagonal entries of $\boldsymbol{W}$. Similarly the proposed composite LSC (cLSC) affinity matrix is no longer normalised and it can be easily verified that the associated degree matrix is

$$\boldsymbol{D} = \boldsymbol{I} - \bar{\boldsymbol{A}} = \mathrm{diag}\{1 - \bar{a}_1, ..., 1 - \bar{a}_N\}. \tag{31}$$

The normalised affinity matrix for composite LSC (cLSC) is

$$\tilde{\boldsymbol{W}} = \gamma \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{D}^{-1/2} + (1-\gamma)\boldsymbol{D}^{-1/2}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^{\mathrm{T}}\boldsymbol{D}^{-1/2} - \boldsymbol{D}^{-1}\tilde{\boldsymbol{A}} = \gamma\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^{\mathrm{T}} + (1-\gamma)\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^{\mathrm{T}} - \boldsymbol{A}_{cLSC}, \quad (32)$$

where $\bar{\boldsymbol{Z}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}$ and $\bar{\boldsymbol{P}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{P}}$ and the ith element of the diagonal matrix $\boldsymbol{A}_{cLSC}$ can be calculated as $\bar{a}_i/(1-\bar{a}_i)$.

In order to exactly compute top-K eigenvectors of $\tilde{\boldsymbol{W}}$ without directly handling $\tilde{\boldsymbol{W}}$, consider initially calculating thin SVD (the first stage) $\bar{\boldsymbol{Z}} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}$, where $\boldsymbol{U}_{\bar{\boldsymbol{Z}}} \in \Re^{N\times q}$ is the left singular vector matrix, and $\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} = \boldsymbol{I}$, $\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}} = \boldsymbol{I}$, $\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}} = \mathrm{diag}\{\lambda_1, ..., \lambda_q\}$, $\lambda_1 > ... > \lambda_q >= 0$ are singular values of $\bar{\boldsymbol{Z}}$.

We have

$$\begin{aligned}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} &= \gamma\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\bar{\boldsymbol{Z}}(\bar{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}) \\ &+ (1-\gamma)\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\bar{\boldsymbol{P}}(\bar{\boldsymbol{P}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}) - \boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{A}_{cLSC}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} \\ &= \gamma\tilde{\boldsymbol{\Lambda}}_{\bar{\boldsymbol{Z}}} + (1-\gamma)\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\bar{\boldsymbol{P}}(\bar{\boldsymbol{P}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}) - \boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\boldsymbol{A}_{cLSC}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} \\ &= \boldsymbol{B}_{cLSC} \in \Re^{q\times q}, \end{aligned} \quad (33)$$

where $\tilde{\boldsymbol{\Lambda}}_{\bar{\boldsymbol{Z}}} = \boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}^2$. The main computational cost of obtaining $\boldsymbol{B}_{cLSC}$ is $O(Nq^2)$.

Similarly to mLSC, the second stage of SVD in the second step is for $\boldsymbol{B}_{cLSC} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}$, so that

$$\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}} \quad (34)$$

or

$$\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}\tilde{\boldsymbol{W}}\boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}} = \boldsymbol{S}_{\boldsymbol{B}}. \quad (35)$$

Let $\boldsymbol{U}_K \in \Re^{N\times k}$ be the sub-matrix of $\boldsymbol{U} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}} \in \Re^{N\times q}$, we have $\boldsymbol{U}_K^{\mathrm{T}}\boldsymbol{U}_K = \boldsymbol{I}$. The Second step: spectral clustering is carried out based on normalised eigenvector matrix $\boldsymbol{U}_K$, and a new set of clusters C_k is updated.

We are now ready to summarise the steps of cLSC in Algorithm 4. The key differences to other methods are that the proposed new type of affinity matrix is learnt from the landmark-based spectral clustering. In an unsupervised setting, Algorithm 3 is initially applied to obtain the initial clusters, followed by the Parzen window probability density function estimators. However, it can also be employed in the semi-supervised setting since the required probability density function estimators can be easily initialized from a small number of labelled data samples. Specifically, in a semi-supervised setting, the Parzen window probability density function estimators of labelled data points (if provided) can be obtained directly. The computational cost is the double that of conventional spectral clustering, plus that of PDF estimation, both can be controlled as they are much smaller than the data size in large-scale data problems.

Remarks:

Analysis to the proposed affinity function in step 2: The proposed affinity function in Step 2 is very different from that of Step 1 or other conventional spectral clustering algorithms in that it aims to capture the pairwise relationships of data points in output space as well as in input space. To explain this in a simple scenario, consider $\gamma \to 0$, the contribution of PDF estimation function to affinity function (before normalisation) for a distinctive data pair $\{\boldsymbol{x}_i,\boldsymbol{x}_j\}$ can be approximately represented as

$$w_{i,j} \propto \sum_{k=1}^{K} p_{C_k}(\boldsymbol{x}_i)p_{C_j}(\boldsymbol{x}_j) \quad (37)$$

of which the output is high (but low otherwise) if both $\boldsymbol{x}_i$ are $\boldsymbol{x}_j$ has high probability of belonging to any of the same clusters (output space). Hence, the affinity function may be more useful than using input space data features alone as in Step 1, or any predetermined affinity function in unsupervised setting without output information.

Analysis to computation complexities: In comparison to previous work based on landmarks [14], the idea of making the algorithm scalable is similar, i.e, designing the low-rank matrix decomposition of the affinity matrix. The actual affinity matrix is used for analysis only, but not used in actual calculation at all. The computational cost of the proposed algorithm is scalable at $O(N)$, with scaling factors determined by $q \ll N$, $N_q \ll N$ and $K \ll N$ for large N. However, our algorithm has higher computation costs than [14], since spectral based clustering is carried out twice rather than once. There is also PDF estimation between the two steps. Additionally, the proposed algorithm employs more complicated two-stage SVD in each Step in order to ensure our affinity matrix has zero diagonals, coming with additional costs. The effects of the affinity matrix having zero diagonals or not on actual clustering results are difficult if not impossible to analyse, thus is out of scope in this paper.

V. Numerical Experiments

A. Classification of handwritten images

1) Clustering results in comparison with other benchmark algorithms: Four benchmark data sets of handwritten digits, *pendigits*, *usps*, *mnist* and *fashion-mnist* are used for comparing the proposed algorithm with some other state-of-the-art approaches described in Section III. A brief summary of the four data sets is provided in Table I. We used this type of data set since it is generally large with high dimensionality, thus likely to benefit from a scalable algorithm in its application. The data set *pendigits* is a handwritten digit data set of 250 samples from 44 writers, collected as sampled coordinate information of each digit from a tablet. The data set *usps* is a standard handwritten database, and *mnist* is a handwritten digits data set, presented as a fixed-sized image. The *fashion-mnist* data set is proposed as a more challenging replacement data set for *mnist*. It is a data-set comprised of 60,000 small square 28×28 pixel gray scale images of items of 10 types of clothing, such as shoes, T-shirts, dresses, and more. A visualisation comparison between these is shown in Figure 1. Note that these data sets are originally divided into training and test data sets for supervised classification tasks. In this work we merge the two parts for our unsupervised setting. As such, the label information given in the data sets are used only for validation, not training. Each data point is normalised to have the unit norm in the pre-processing step. Note that

Algorithm 4 The proposed two-step spectral clustering based on a novel composite LSC affinity matrix (cLSC).

Require: Data matrix $\boldsymbol{X} \in \Re^{N\times d}$, number K of clusters to construct, number of landmarks q, weighting γ. (Or labels for some data points in $\boldsymbol{X}$ for semi-supervised setting.)

Initialization:
Apply Algorithm 3 to obtain clusters C_k, $k = 1, ..., K$; (For semi-supervised learning using directly existing labelled clusters).
Calculate $\boldsymbol{P}$ and $\tilde{\boldsymbol{P}}$ using (27) and (28).
Produce a new set of q landmark points using random selection, and find sparse affinity matrices $\tilde{\boldsymbol{Z}} = \boldsymbol{Z}\tilde{\boldsymbol{D}}^{-1/2}$ based on the new set of q landmark points.
Find degree matrix $\boldsymbol{D}$ using (31).
Compute the first q eigenvectors and eigenvalues $\bar{\boldsymbol{Z}} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{\Lambda}_{\bar{\boldsymbol{Z}}}\boldsymbol{V}_{\bar{\boldsymbol{Z}}}^{\mathrm{T}}$, where $\bar{\boldsymbol{Z}} = \boldsymbol{D}^{-1/2}\tilde{\boldsymbol{Z}}$.
Find $\boldsymbol{A}_{cLSC}$, $\boldsymbol{B}_{cLSC}$ using (30) and (33).
Perform the singular value decomposition (SVD) of $\boldsymbol{B}_{mLSC} = \boldsymbol{U}_{\boldsymbol{B}}\boldsymbol{S}_{\boldsymbol{B}}\boldsymbol{U}_{\boldsymbol{B}}^{\mathrm{T}}$.
Calculate $\boldsymbol{U} = \boldsymbol{U}_{\bar{\boldsymbol{Z}}}\boldsymbol{U}_{\boldsymbol{B}}$.
Form the matrix $\boldsymbol{T} = \{t_{i,j}\} \in \Re^{N\times K}$ by normalising the rows to norm one, i.e. to set

$$t_{i,j} = u_{i,j} \Big/ \sqrt{\sum_{k=1}^{K} u_{i,j}^2}. \tag{36}$$

for $i = 1, ..., N$ **do**
  Let $\boldsymbol{y}_i \in \Re^K$ be the vector corresponding to the i-th row of $\boldsymbol{T}$.
end for
Cluster the points $\boldsymbol{y}_i$, $i = 1, ..., N$ with the k-means algorithm [1] into clusters $C_1, ..., C_K$.
Return: Find clusters $k \in \{1, ..., K\}$ with $\{k, \boldsymbol{y}_i \in C_k\}$ and assign original data points $\boldsymbol{x}_i$ according to clusters index set of $k = 1, ..., K$.

(a) (b)

Fig. 1. 10 sample images per class are shown as visual comparison between (a) *mnist* and (b) *fashion-mnist*; Each data-set has 70000 images.

for validation of all the comparable algorithms, the predicted cluster labels need to be mapped into those provided by the database via the well known Kuhn-Munkres algorithm [38] which is also used in [14].

Table II outlines the clustering accuracy, which is the percentage of data points that are correctly clustered, of applying the proposed method during the second step with $\gamma = 0.001$. The results of a number of comparative methods are listed in comparison, which demonstrates the proposed method's superior results of classification accuracy. Further explanation for experiments in Table II is given below:

1) Original NJW. This is named the same way as [15] which is based on the normalised spectral clustering algorithm of [10]. We used Gaussian functions as the affinity functions, where the width is set by trial and error for the best result.
2) Scalable NJW. This is Algorithm [15]. We set the default fraction of outliers as 0.01, and the clustering classification is carried out on non-outlier parts of the data set.
3) Nyström. We followed [14] to choose [39] as a comparison, which is a MATLAB implementation with

TABLE I
A SUMMARY OF FOUR IMAGE DATA SETS.

Data sets	Number of data size (N)	Number of features(d)	Number of classes (K)
pendigits	10992	16	10
usps	9298	256	10
mnist	70000	784	10
fashion-mnist	70000	784	10

TABLE II
CLUSTERING ACCURACY IN PERCENTAGE % (MEAN± STANDARD DEVIATION).

Data sets	Original NJW	Scalable NJW	Nyström	LSC-K	LSC-R	The Proposed cLSC
pendigits	74.8	73.6	70.8 ± 4.1	82.9 ± 3.4	81.4 ± 5.2	95.9 ± 0.4
usps	67.5	67.9	65.3 ± 2.9	71.9 ± 3.6	72.7 ± 2.8	90.6 ± 0.3
mnist	*72.46*[1]	52.8	54.3 ± 2.5	72.6 ± 5.7	72.0 ± 5.2	88.9 ± 0.3
fashion-mnist	NA	56.0 ± 0.1	55.9 ± 1.9	57.6 ± 1.8	57.3 ± 2.1	74.5 ± 0.4

[1]cited from [14] for reference only since our calculation shows out of memory for *mnist* and *fashion-mnist*.

TABLE III
RECORDED RUNNING TIME OF THE PROPOSED CLSC ALGORITHM (SECONDS) .

Data sets	No. of Land marks ($\times 10^2$)							
	3	4	5	6	7	8	9	10
pendigits	2.37	2.45	3.19	3.67	4.23	4.61	4.96	6.27
usps	4.18	4.97	5.40	5.90	6.61	7.58	8.50	9.18
mnist	89.39	98.59	104.57	107.63	122.96	127.78	139.61	149.59
fashion-mnist	85.80	91.75	100.84	111.83	119.62	127.87	132.57	136.20

orthogonalisation, available online http://alumni.cs.ucsb.edu/~wychen. The hyper-parameter is set by trial and error for the best result too.

4) LSC-K and LSC-R. The two landmark algorithms [14] are also available online http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html. Table II reports the results of landmark $q = 1000$, $r = 6$. The kernel width h for landmark is set as the mean distance between two data points in the data set.
5) The proposed cLSC with $\gamma = 0.001$. The results uses the same parameters $q = 1000$, $r = 6$ for a fair comparison. The results are recorded in which step 2 has $N_q = 250$. The kernel width h for landmark is also set as the mean distance between two data points in the data set.

Note that in Table II, since the results for Nyström and landmark based algorithms (LSC-K, LSC- and the proposed algorithm) are subject to random effects, the related experiments are repeated 20 times, with mean and standard deviation being reported. To demonstrate the computational costs of the proposed algorithm, Table III shows a set of recorded running time with $N_q = 250$, obtained from MATLAB 2018a on a desktop, with specification of Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz, 16.0 GB RAM, 64-bit operating system.

2) Cluster results with respect to landmark size q: The proposed algorithm is applied over the three data sets by varying the parameters, and the clustering results are plotted in Figures 2–5, showing the clustering accuracy versus number of landmarks with $\gamma = 0.001$ of the first step (initialisation step) and second step for the three data sets with $r = 6$. The kernel width h for landmark is set as the mean distance between two data points in the data set. These results clearly demonstrate that improvements in terms of both mean and variance are significant by using the PDF-estimation based affinity matrix.

3) Clustering results with respect to choice of γ: The proposed algorithm is applied to the three data sets by varying γ in the range $0 < \gamma < 1$. The data experiments are based on fixing landmark sizes to three typical values of $q = 100, 200, 500$ respectively. The PDF estimation sample size is set as $N_q = q$ for convenience. The kernel width h for landmark is also set as the mean distance between two data points in the data set, and $r = 6$. Figures 6-9 show the clustering accuracy for the first step (initialization step) and second step. Note that the first step does not use γ, so the plot is flat with random effects due to random landmarks.

- The percentage contribution of PDFs used in the second stage affinity matrix construction is quantified as $(1-\gamma)$. This can be interpreted as the information obtained in the first stage spectral clustering has been used as a prior for the second stage in order to improve the clustering results by the proposed algorithm. The remaining γ percentage contribution to affinity matrix is due to using more landmarks. If γ is close to one, it means that both first and second stages are based on two sets of independent landmarks totally. If γ is close to zero, then the second stage completely uses PDF results from the first stage to build its affinity matrix (without using the new set of landmarks), followed by its use in spectral clustering in second stage.

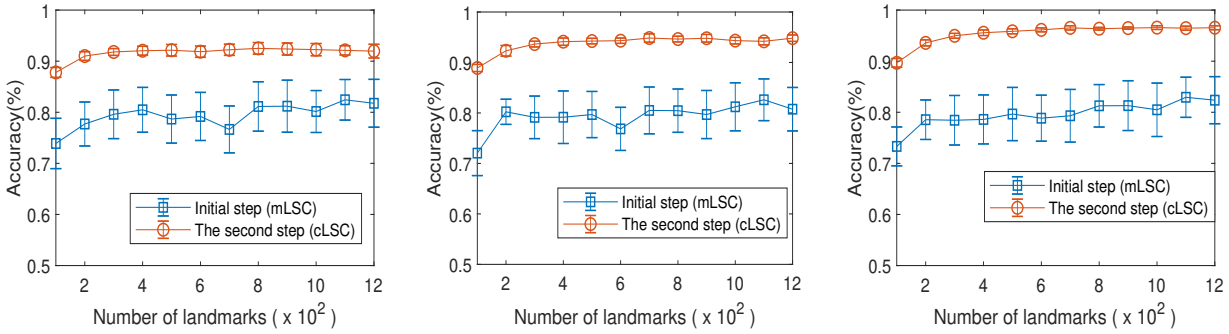

Fig. 2. Clustering accuracy versus number of landmarks for *pendigits* data set; (a) $N_q = 50$; (b) $N_q = 100$; and (c) $N_q = 250$; The results show that it is better to have more landmarks to cover the input data space, as well having sufficient samples in PDF estimation N_q.

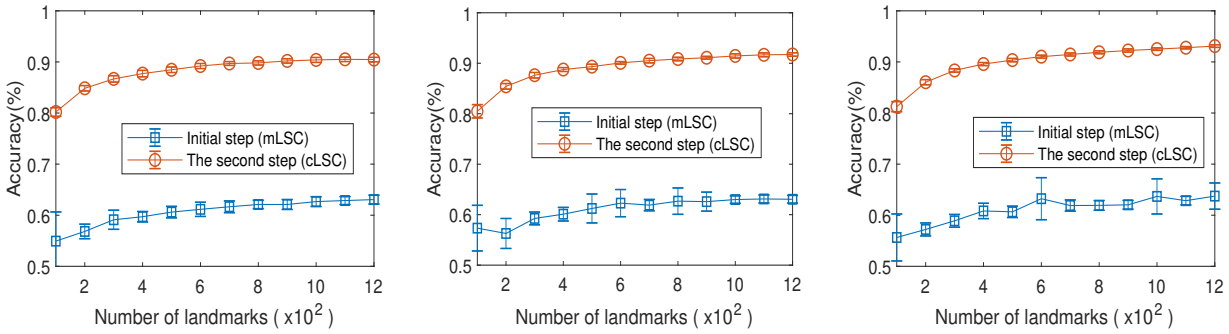

Fig. 3. Clustering accuracy versus number of landmarks for *usps* data set; (a) $N_q = 50$; (b) $N_q = 100$; and (c) $N_q = 250$. The results show that it is better to have more landmarks to cover the input data space, as well as having sufficient samples in PDF estimation N_q.

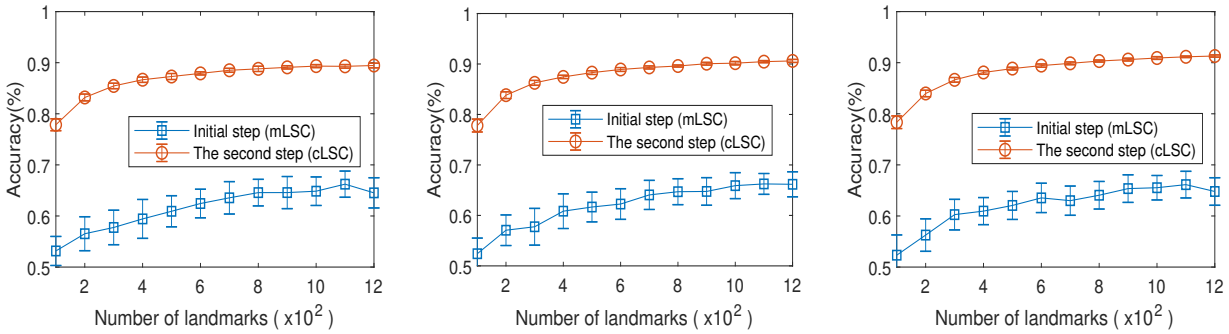

Fig. 4. Clustering accuracy versus number of landmarks for *mnist* data set; (a) $N_q = 50$; (b) $N_q = 100$; and (c) $N_q = 250$. The results show that it is better to have more landmarks to cover the input data space, as well as having sufficient samples in PDF estimation N_q.

- As shown in Figures 6– 9, it can be seen that the second stage always improve clustering results even with γ close to one, since there are twice as many landmarks than the first step. However, it is best to set γ close to zero, in which case the PDFs of the first stage are used almost completely in constructing the affinity matrix.
- The performance is quite stable for a range of small values of γ, e.g. $\gamma < 0.1$, except for the most challenging *fashion-mnist* data set. The results have clearly shown that PDF-based affinity matrices is superior to the raw landmark-based one, in both higher mean accuracy and smaller standard deviation.

4) Semi-supervised experiments: The above two-step spectral clustering algorithm is based on unsupervised scenario where no prior information on clusters are given. During the second step clustering there is prior information which are the estimation of clusters from the first step, i.e. the class labels are obtained. In the semi-supervised setting, it is assumed that we have a small number of labelled data points, also denoted as N_q. These can simply be used to construct PDF (26) directly, followed by the final affinity matrix construction in a single step. We perform experiments using a range of N_q

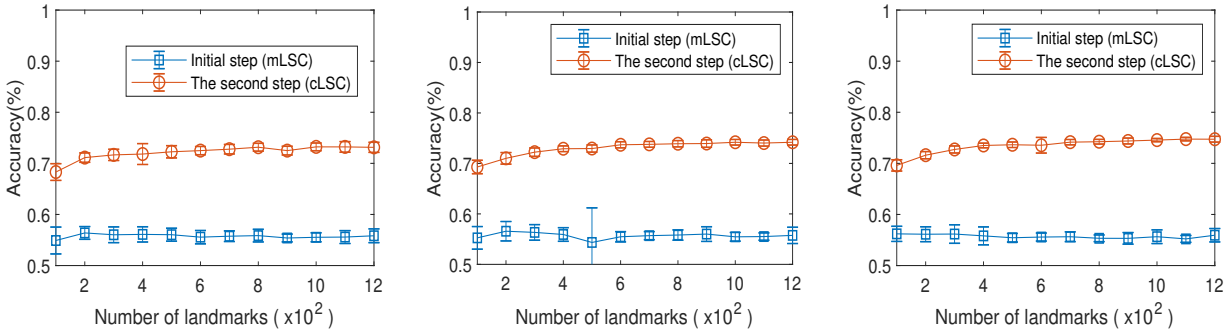

Fig. 5. Clustering accuracy versus number of landmarks for *fashion-mnist* data set; (a) $N_q = 50$; (b) $N_q = 100$; and (c) $N_q = 250$. The results show that it is better to have more landmarks to cover the input data space, as well as having sufficient samples in PDF estimation N_q.

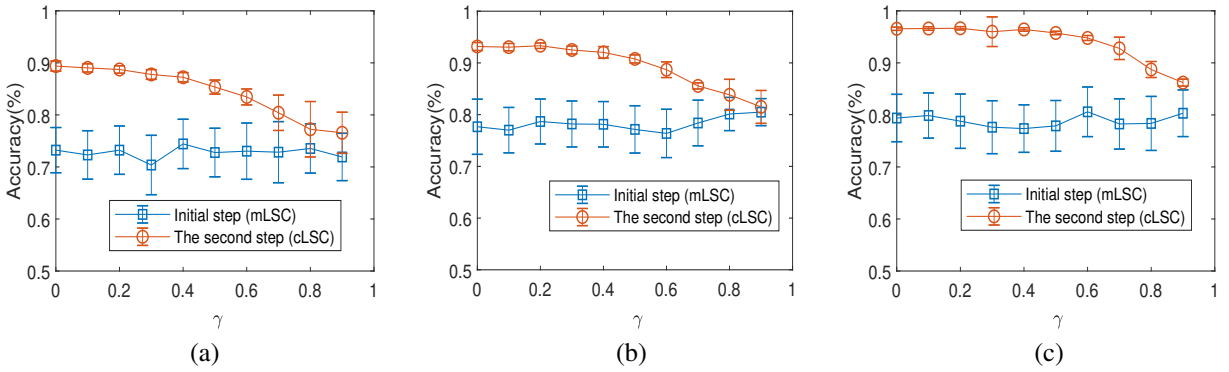

Fig. 6. Clustering accuracy versus value of γ for *pendigits* data set; (a) $q = N_q = 100$; (b) $q = N_q = 200$; and (c) $q = N_q = 500$; It can be seen that the second stage improves clustering results mostly when γ is close to zero.

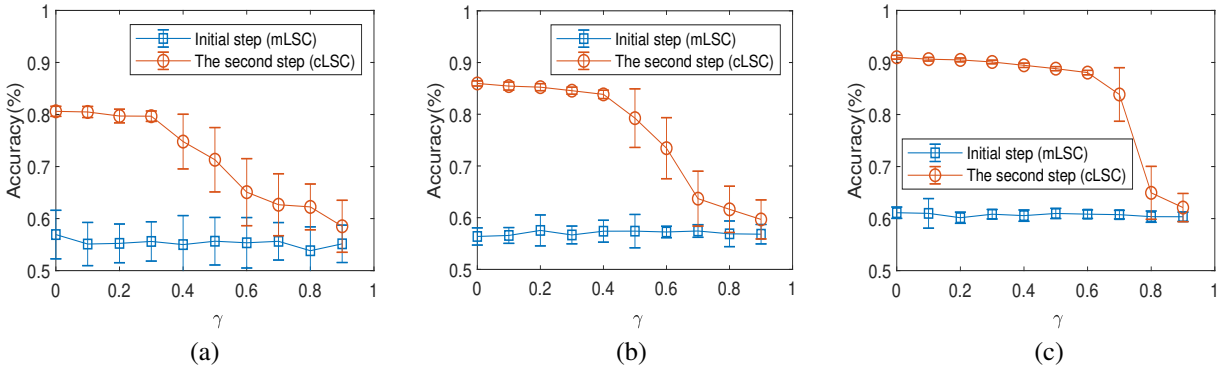

Fig. 7. Clustering accuracy versus value of γ for *usps* data set; (a) $q = N_q = 100$; (b) $q = N_q = 200$; and (c) $q = N_q = 500$; It can be seen that the second stage improves clustering results mostly when γ is close to zero.

to denote the number of labelled data samples in each class in a semi-supervised setting. In the semi-supervised case, the true labels as given in the three data sets are randomly sampled and quantities of (26) are constructed directly from sampled labelled data points using provided in data sets, which are directly used to establish the affinity matrix. Note that the spectral clustering algorithm is carried out only once based on the composite affinity matrix of Equation (29). Figure 11 has compared the clustering accuracy's of the semi-supervised with its unsupervised counterparts, in which the landmark size is set as $q = 500$, $r = 6$ and $\gamma = 0.001$. The results have shown that the proposed algorithm is comparable to semi-supervised counterparts, even though this is an unsupervised approach and do not have training labels for these parameter settings. Note that the landmarks numbers q and N_q are restricted to be quite small in comparison to data size since the main aim of our proposed algorithm is scalable clustering.

B. A medical image segmentation application

Image segmentation is an important step towards image understanding and interpretation. The staining method [41]

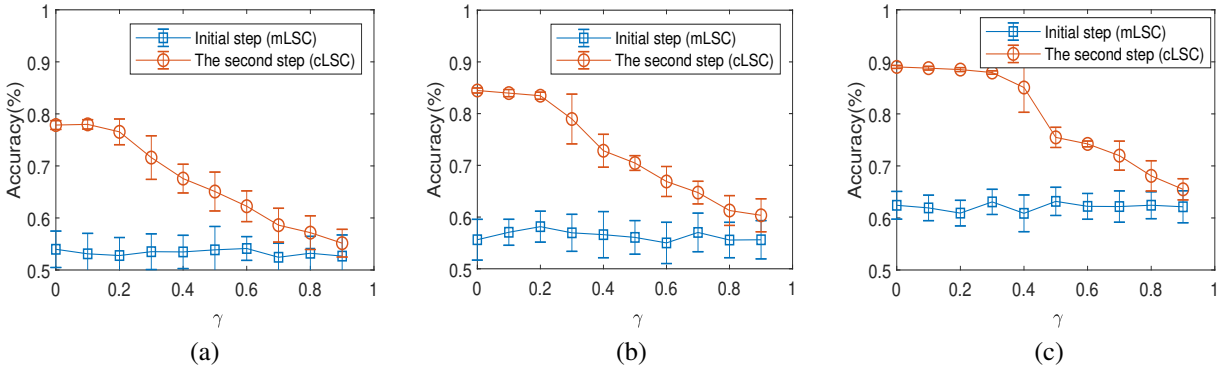

Fig. 8. Clustering accuracy versus value of γ for *mnist* data set; (a) $q = N_q = 100$; (b) $q = N_q = 200$; and (c) $q = N_q = 500$; It can be seen that the second stage improves clustering results mostly when γ is close to zero.

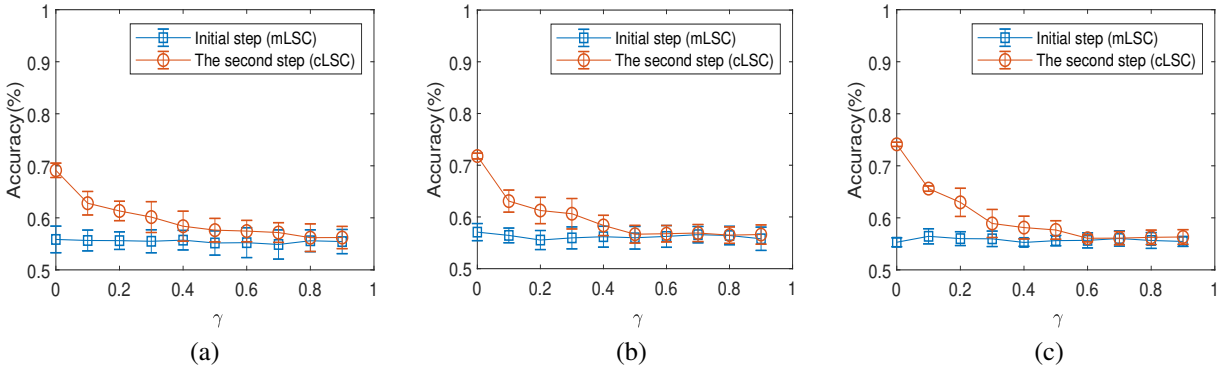

Fig. 9. Clustering accuracy versus value of γ for *mnist* data set; (a) $q = N_q = 100$; (b) $q = N_q = 200$; and (c) $q = N_q = 500$; It can be seen that the second stage improves clustering results mostly when γ is close to zero.

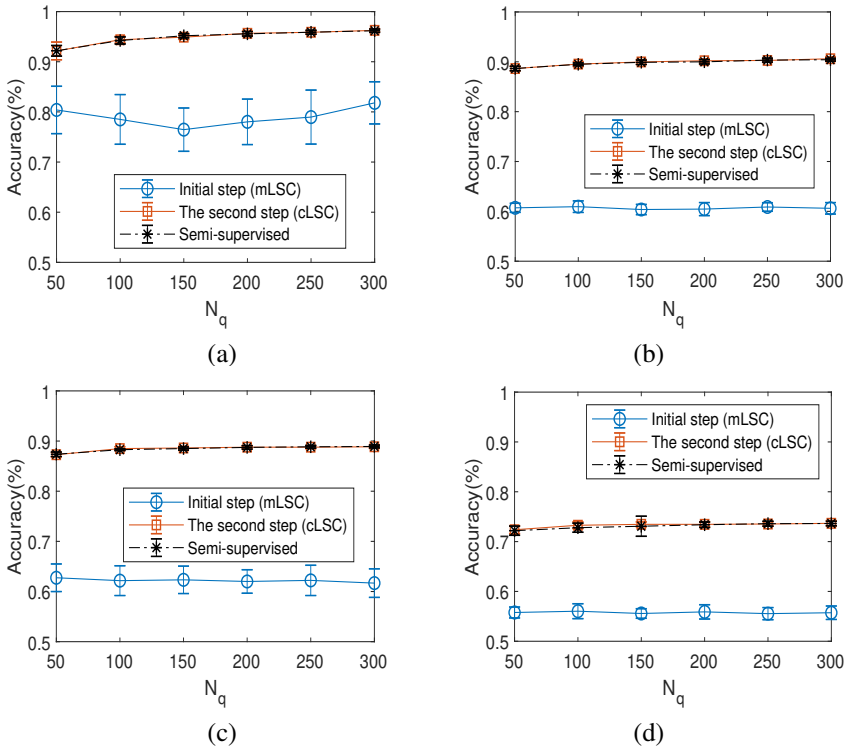

Fig. 10. Clustering accuracy versus value of N_q of Semi-supervised approaches for three data sets; (a) *pendigits*; (b) *usps*; (c) *mnist*; and (d) *fashion-mnist*; The results have shown that the proposed algorithm is comparable to semi-supervised counterpart, in spite of being an unsupervised approach.

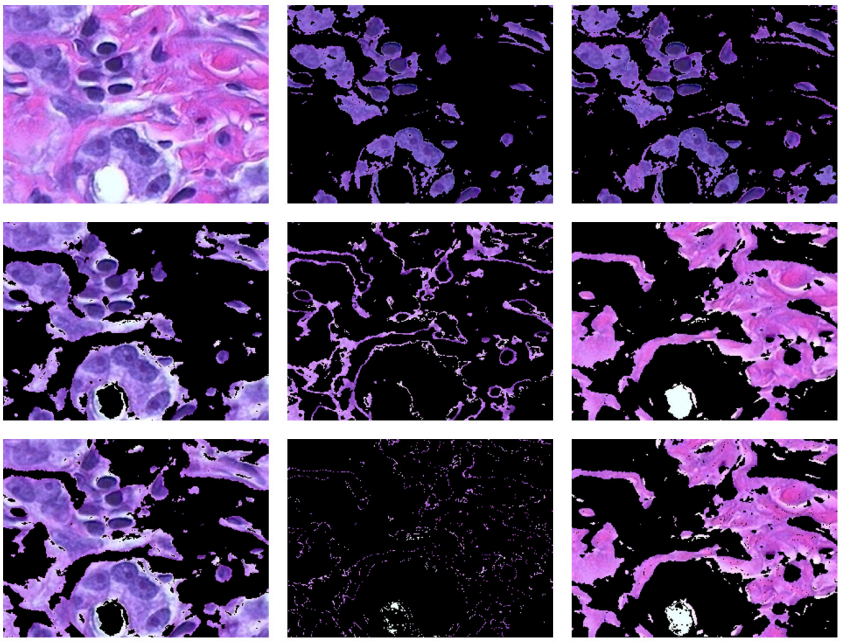

Fig. 11. Segmentation results of a tissue image stained with hemotoxylin and eosin (H & E); Top row: The original image and Segmented blue Nuclei based on step 1 and Step 2 respectively, after a standard threshold method; Middle row: segmented three clusters from step 1; left image, after thresholding, yields the top middle segmented image; Bottom row: segmented three clusters from step 2; left image after thresholding, yield top right segmented image.

helps pathologists distinguish different tissue types. The image in the left-top corner of Figure 11 shows the tissues stained with hemotoxyline and eosin (H & E). We will use this as a demonstration of applying the proposed method for segmentation based on image colors in an automated fashion. The aim is to segment the cell nuclei which is in dark blue color based on colour features [42]. In making use of colour information, various colour spaces, such as RGB [43], HSV [44], L*a*b* [45], etc. were attempted. Different from the RGB colour space, in which colour information is mixed with red (R), green (G), and blue (B) channels, HSV and L*a*b* colour spaces separate the colour information from the brightness/intensity of an image. In such a way, the colour information can be dealt with separately in case only colour information is interesting in segmentation.

The original image (size 227×303) is as shown in the top-left corner of Figure 9, which was converted from RGB color space to CIE L*a*b* (CIELAB) and HSV space respectively. CIE L*a*b* (CIELAB) is a color space specified by the International Commission on Illumination. It describes all the colors visible to the human eye and was created to serve as a device-independent model to be used as a reference. In L*a*b* color space, chromaticity-layer 'a*' indicating where color falls along the red-green axis, and chromaticity-layer 'b*' indicating where the color falls along the blue-yellow axis. Alternatively, HSV (hue, saturation, value) is an alternative representation of the RGB color model, designed by computer graphics researchers to more closely align with the way human vision perceives color-making attributes. The HSV representation models the way paints of different colors mix together, hue 'H' is the color portion of the model, saturation 'S' describes the amount of gray in a particular color. The value dimension 'V' resembles the mixture of those paints with varying amounts of black or white paint. From these understanding, we choose 'a*', 'b*' values from CIE L*a*b* space, and 'H' from HSV to form a three dimensional feature since they can capture color information in the original image.

We obtain the input data matrix $\boldsymbol{X}$ by vectorizing two converted images as $N = 227 \times 303 = 68781$. Our $d = 3$ input features are from the two converted images, based on 'a*', 'b*' values from CIE L*a*b* space, and 'H' from HSV since these features are most relevant for the sake of generating initial clusters which are separated in color space. The number of clusters is preset as $K = 3$. $N_q = q = 500$, $\gamma = 0.001$, $r = 6$ are predetermined. The segmentation process and results are plotted in Figure 11 and as explained as follows. The proposed algorithm produces three clusters from step 1 (Figure 11 middle row). Three clusters are finally obtained from step 2 as shown (Figure 11 bottom row). Similarly, for easy visual comparison, the Kuhn-Munkres algorithm [38] has been applied so that the clusters from the two steps match each other.

Since 'L*' layer in the converted CIE L*a*b* (CIELAB) image contains the brightness values of each color, these are finally used to extract the brightness values of the pixels in the blues clusters (the two images below the original images). By applying a standard threshold method to these clusters, we obtain the dark blue pixels and return the final segmented blue Nuclei. Note that since blueness amongst three clusters can be quantified, hence the total process,

between clustering and segmentation, can be fully automatic. Alternatively, human intervention can be included if this is preferable. For comparison the segmentation based on clusters both steps 1 and 2 are shown as the top middle and top right images respectively. Note that in medical image segmentation applications, often there is no labelled information which makes it difficult to quantify the results, however this example can still demonstrate how to segment colors in an automated fashion using the proposed two-step spectral clustering algorithm using landmarks and probability density estimation, which clearly matches with human vision. Future research will explore other useful computer vision applications.

VI. Conclusions

This paper has investigated a novel two-step scalable spectral clustering algorithm which includes an additional step based on a new affinity matrix, defined based on class probability density using the estimated clusters. The proposed algorithm follows the idea of landmark-based low-rank affinity matrices to control the computational costs so the algorithm scales well with data size. The second stage makes use of an improved affinity function incorporating cluster labels, which become available within the algorithm. The class PDF-based affinity function is conceptually different from the affinity function in the first step, which is able to associate data samples that is close in output space than purely from the input space as the original spectral clustering. It is demonstrated that the proposed algorithm is capable of achieving far superior performances than other state-of-the-art algorithms for several benchmark multi-class image data sets. Various experiments have been carried out over a range of parameter settings in order to gain insights. The algorithm has a higher computational cost than the previous landmark-based spectral clustering research, but still scales well with data size. Finally, an example is provided to demonstrate the working process and usefulness of the proposed algorithm in color based medical image segmentation that can help pathologists to distinguish different tissue types. Though out of the scope of this paper, future work would investigate combining other active research areas such as HPC and machine learning production - this would be highly relevant to scalable computational methods.

References

[1] S. Haykin, *Neural Networks and Learning Machines*. Pearson Education Inc., 2009.

[2] J. Shi and J. Malik, "normalised cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[3] D. Pandove, S. Goel, and R. Rani, "Systematic review of clustering high-dimensional and large datasets," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 2, pp. 1-68, 2018.

[4] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 2010.

[5] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei and P. Zhu, " One-Step Multi-View Spectral Clustering, " *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, No.10, pp. 2022–2034, 2019.

[6] L. Bonfils, A. Samé, L. Oukhellou, " Dynamic clustering and modeling of temporal data subject to common regressive effects," *Neurocomputing*, Vol. 500, pp. 217-230, 2022.

[7] Y. Xu, S. Arai, D. Liu, F. Lin, K. Kosuge, "FPCC: Fast point cloud clustering-based instance segmentation for industrial bin-picking," *Neurocomputing*, Vol 494,pp. 255-268, 2022.

[8] J. Kauffmann, M. Esders, L. Ruff, G. Montavon, W. Samek and K. -R. Müller, " From Clustering to Cluster Explanations via Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022, In Press.

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

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 849–856,2002

[11] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1477–1486, 2014.

[12] F. R. Chung, *Spectral Graph Theory*, Ser. CBMS Regional Conference Series in Mathematics. American Mathematical Society, vol. 92, 1997.

[13] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *Proceedings of the seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 975–982, 1999

[14] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.

[15] G. Chen, "Scalable spectral clustering with cosine similarity," in *Proceedings of 24th International Conference on Pattern Recognition*, Beijing, China, Aug. 2018.

[16] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proceedings of NIPS*, pp. 682–688, 2001.

[17] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, Feb 2004.

[18] A. Choromanska, T. Jebara, H. Kim, M. Mohan, and C. Monteleoni, "Fast spectral clustering via the nyström method," in *Algorithmic Learning Theory*, ser. Lecture Notes in Computer Science, S. Jain, R. Munos, F. Stephan, and T. Zeugmann, Eds., vol. 8139. Springer, Berlin, Heidelberg, pp. 367–381. 2013.

[19] Y. Kang, B. Yu, W. Wang, and D. Meng, "Spectral clustering for large-scale social networks via a pre-coarsening sampling based nyström method," in *Advances in Knowledge Discovery and Data Mining (PAKDD)*, ser. Lecture Notes in Computer Science, T. Cao, E. P. Lim, Z. H. Zhou, T. B. Ho, D. Cheung, and H. Motoda, Eds., vol. 9078. Springer, Cham, pp. 106–118. 2015

[20] M. Mohan and C. Monteleoni, "Exploiting sparsity to improve the accuracy of nyström-based large-scale spectral clustering," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 9–16, 2017.

[21] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, vol. 7, pp. 1963–2001, 2006.

[22] Shulin Wang, Fang Chen, and Jianwen Fang, "Spectral clustering of high-dimensional data via nonnegative matrix factorization," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2015,

[23] S. Wu, X. Feng, and W. Zhou, "Spectral clustering of high-dimensional data exploiting sparse representation vectors," *Neurocomputing*, vol. 135, no. 7, pp. 229–239, 2014.

[24] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," NIPS'04: Proceedings of the 17th International Conference on Neural Information Processing Systems, pp. 1601–1608,2004

[25] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering read more: https://epubs.siam.org/doi/abs/10.1137/1.9781611972757.70," in *Proceedings of the SIAM International Conference on Data Mining*, 2005.

[26] Y. Zhao, Y. Yuan, F. Nie, and Q. Wang, "Spectral clustering based on iterative optimization for large-scale and high-dimensional data," *Neurocomputing*, vol. 318, no. 11, pp. 227–235, 2018.

[27] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. Sydney, Australia: PMLR, pp. 1985–1994, 06–11 Aug 2017,

[28] U. Shaham, K. Stanton, H. Li, R. Basri, B. Nadler, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," in *Proceedings of the International Conference on Learning Representation*, 2018.

[29] J. Wang, A. Hilton, and J. Jiang, "Spectral analysis network for deep representation learning and image clustering," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1540–1545, 06–11 Aug 2017,

[30] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep spectral clustering using dual autoencoder network," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4066–4075, 2019.

[31] Y. Li, J. Chen, Y. Zhao, and H. Lu, "Adaptive affinity matrix for unsupervised metric learning," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2016.

[32] C. Boutsidis, A. Gittens, and P. Kambadur, "Spectral clustering via the power method - provably," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, pp. 40–48, 2015.

[33] S. Wang, A. Gittens, and M. W. Mahoney, "Scalable kernel K-means clustering with Nyström approximation: Relative-error bounds," *Journal of Machine Learning Research*, vol. 20, pp. 1–49, 2019.

[34] T. Li, *Data Clustering: Algorithms and Applications*, ch. Nonnegative Matrix Factorizations for Clustering: A Survey, pp. 1–28. Chapman and Hall/CRC, 2018.

[35] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of 27th International Conference on Machine Learning (IMCL)*, Haifa, Israel, pp. 679–686, 2010.

[36] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[37] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley, 1992.

[38] L. L. Lovasz and M. D. Plummer, *Matching theory*, Amsterdam ; New York : North-Holland : Elsevier Science Publishers B.V. ; New York, N.Y. : Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co, 1986.

[39] W. Y. Chen, Y. Song, H. Bai, C. J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2010.

[40] X. Zhu, Y. Zhu, and W. Zheng, "Spectral rotation for deep one-step clustering" *Pattern Recognition*, vol. 105, 107175, 2020.

[41] J. K. Presnell, M. P. Schreibman, and G. L. Humason, *Humason's Animal tissue techniques*, 5th ed. Baltimore: Johns Hopkins University Press, 1997.

[42] F. Garcia-Lamont, J. Cervantes, A. Lopez, and L. Rodriguez, "Segmentation of images by color features: A survey," *Neurocomputing*, vol. 292, pp. 1–27, 2018.

[43] A. R. F. Araujo and D. C. Costa, "Local adaptive receptive field self-organizing map for image color segmentation," *Image and Vision Computing*, vol. 27, no. 9, pp. 1229–1239, 2009.

[44] K. Deb, S. J. Kang, and K. H. Jo, "Statistical characteristics in HSI color model and position histogram based vehicle license plate detection," *Intelligent Service Robotics*, vol. 2, no. 3, pp. 173–186, 2009.

[45] R. Huang, N. Sang, D. Luo, and Q. Tang, "Image segmentation via coherent clustering in Lab color space," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 891–902, 2011.