Fast kernel spectral clustering based on incomplete Cholesky factorization for large scale data analysis

Mihály Novák[a,*], Carlos Alzate[b,c], Rocco Langone[c], Johan A.K. Suykens[c]

[a] *Institute for Nuclear Research of the Hungarian Academy of Sciences, MTA ATOMKI, 18/c Bem tér, H-4026 Debrecen, Hungary*
[b] *Smarter Cities Technology Centre IBM Research, Ireland Damastown Industrial Estate Mulhuddart, Dublin 15, Ireland*
[c] *Department of Electrical Engineering, ESAT-STADIUS, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*

Abstract

A fast spectral clustering algorithm, suitable for large scale data clustering problems, is presented in this work. The algorithm is an improved version of the sparse multiway spectral clustering algorithm, formulated as weighted kernel principal component (PCA) analysis within primal-dual least-squares support vector machine (LS-SVM) framework. Sparsity is introduced by the combination of reduced set method with the incomplete Cholesky factorization based low rank approximation of the kernel matrix. The out-of-sample capability of the clustering model and the possibility of model selection provide the opportunity to obtain the optimal model parameters through classical training-validation phases and the obtained optimal model can be easily extended to unseen data points. However, the computational complexity of the original sparse algorithm is quadratic in the training set size that prevented the application of the method to large scale data analysis problems. This quadratic computational complexity is reduced to linear in the proposed algorithm while attractive properties such as the simple way of out-of-sample extension, possibility of model selection through training and validation are conserved. Furthermore, a new, soft cluster membership encoding-decoding scheme and corresponding model selection criterion are also proposed. These modifications make the algorithm capable to solve much larger size of clustering problems than the original one with a refined model selection procedure. The significantly reduced computational complexity, the improved accuracy and sparsity of the presented algorithm compared to the original version are illustrated by experimental results obtained on difficult toy as well as real life problems such as image segmentation.

Keywords: spectral clustering, sparse model, large-scale data, kernel methods, LS-SVM

*Corresponding author
Email address: M.Novak@atomki.mta.hu (Mihály Novák)

1. Introduction

Clustering algorithms comprise unsupervised, explanatory data mining techniques with the common goal of grouping objects that are similar according to some similarity measure. Beyond the classical clustering algorithms such as K-means, a set of methods have been developed that make use of some eigenvectors of a normalized affinity matrix derived from the data available. These spectral clustering algorithms are known to perform well even in cases when classical methods fail and have been successfully applied to data analysis, image processing, pattern recognition and to many other machine learning related problems.

Classical formulation of spectral clustering algorithms [1, 2, 3, 4, 5, 6] starts from a graph partitioning problem that is NP-hard due to the discrete constraint on the cluster indicators. By relaxing the discrete constraint the problem is transformed to an eigenvalue problem involving a normalized affinity matrix formed from pairwise similarities between the data points in hand. The discrete cluster indicators are then inferred from some selected eigenvectors that give the optimal solution to the relaxed problem.

Traditional spectral clustering models are typically based on the full set of data due to the lack of clear extension to unseen data points. As a consequence, the memory usage and computation time of these algorithms grow quickly by increasing data sizes due to the increasing size of the involved eigenvalue problem. A possible solution to overcome these problems is offered by the spectral clustering algorithms [7, 8] based on the Nyström method [9]. A clustering model is trained on a subset of data points. The training phase involves the approximation of the implicit eigenfunctions by means of Nyström method. The cluster membership of out-of-sample data points can then be determined by using this approximation. However, the underlying clustering model is not known and the parameter selection is done in a heuristic way.

Unlike the classical spectral clustering algorithms, that are formulated on graph theoretical bases, a clustering algorithm is proposed in [10, 11] as a weighted form of kernel principal component analysis [12, 13] derived within the least-squares support vector machine (LS-SVM) formalism [14] by using a primal-dual optimization framework. This kernel spectral clustering (KSC) algorithm provides a straightforward way to extend the clustering model, trained on a subset of data point, to unseen points through simple projection to some eigenvectors obtained during the training phase. Furthermore, a model selection criterion can be derived by making use of the known structure of these eigenvectors in the ideal case. The optimal model parameters such as the number of clusters or kernel parameters can be determined through the maximization of this model selection criterion. Therefore, this kernel spectral clustering model can be trained on a subset of data, validated and extended to out-of-sample points similarly to supervised methods in spite of the unsupervised nature of the clustering problem.

The out-of-sample extension capability of the KSC model directly provides the possibility of sparse model construction by training the model on a selected

subset of the full data set. It has been shown in [11] that the KSC model outperforms the Nyström approximation based clustering algorithm [8] both in terms of accuracy and computation time even in case of random training set selection. Moreover, the computation time and the memory requirement of the original KSC model can be significantly reduced by using a clever training set selection that can notably increase the sparsity of the clustering model while improving the accuracy. Such sparse versions of the original KSC model were proposed based on the combination of reduced set method [15, 16] with the incomplete Cholesky decomposition (ICD) based low rank approximation of the kernel matrix in [17] and by using a Rényi entropy maximization based training set selection in [18].

The sparse version of the KSC model, based on the combination of ICD and reduced set method, was shown to be able to perfectly solve problems based on a small reduced set size that are very hard even for spectral clustering algorithms due to the properties of the corresponding spectrum [17] (see Sections 3.2.3 and 5.1). However, the computational complexity of this algorithm is quadratic in the training set size that prevented the application to large scale problems.

An improved version of the ICD based sparse KSC algorithm is presented in this paper. The quadratic computational complexity in the training set size is reduced to linear while all the attractive properties such as the simple way of out-of-sample extension, possibility of model selection through training and validation are conserved. A new, soft cluster membership encoding-decoding scheme and a corresponding new model selection criterion is also proposed both for the original non-sparse KSC algorithm and for the presented sparse version. These modifications make the algorithm capable to solve much larger size of clustering problems than the original one.

The paper is organized as follows: spectral clustering problem and classical algorithms are introduced in Section 2. The original non-sparse KSC model is reviewed in Section 3. Important properties of some eigenvectors and the corresponding score variables, that play an essential role in the spectral clustering interpretation of the weighted kernel PCA and in the introduced model selection criterion, are also discussed. In Section 4, first the original ICD based sparse KSC model and algorithm is reviewed by highlighting the computational bottlenecks then a new algorithm is proposed. The significantly improved computational time, sparsity, and accuracy of the proposed algorithm over the original version is illustrated through some computational experiments in Section 5.

2. Spectral clustering

Given a set of data points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d$ together with some similarity measure $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ between pairs of data points $\mathbf{x}_i, \mathbf{x}_j$ the goal of clustering is to partition the input data set such that points in the same subset are similar and points in different subsets are dissimilar. An intuitive form of representing the data with the given similarity measure is using a similarity graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices, representing the data points in the graph, and $\mathcal{E}$ is the set of edges between the vertices weighted by the

similarity between the corresponding data points. If the similarity measure is symmetric then the G graph is undirected and a symmetric similarity matrix S can be constructed from $\mathcal{E}$ such that $S \in \mathbb{R}_{\geq 0}^{N \times N}, [S]_{ij} = s_{ij} = s(\mathbf{x}_i, \mathbf{x}_j) = s(\mathbf{x}_j, \mathbf{x}_i), \forall i, j \in \{1, ..., N\}$. The degree of a vertex d_i, that represents the sum of weights to this node, i.e. sum of similarities between data point $\mathbf{x}_i$ and all the data points in $\mathcal{D}$, is defined as $d_i = \sum_{j=1}^{N} s(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j=1}^{N} [S]_{ij}$. The degree matrix D, associated to the given similarity graph G, is a diagonal matrix with $[d_1, ..., d_N]$ diagonal elements.

The above clustering problem can now be reformulated based on the similarity graph as a graph partition problem [2] [6]. The goal is to find a partition of the vertices $\mathcal{V}$ into $\mathcal{K}$ disjoint subsets $\{A_1, ..., A_\mathcal{K}\}, \mathcal{V} = \mathcal{A}_1 \cup, ..., \cup \mathcal{A}_\mathcal{K}$, and $\mathcal{A}_k \cap \mathcal{A}_l = \emptyset, \forall k, l \in \{1, ..., \mathcal{K}\}, k \neq l$ such that the edges between vertices belonging to the same subset have a high weight while the edges between vertices from different subsets have a low weight. Formulating this would lead to the so-called *mincut* problem [19]. However, *mincut* can provide unbalanced partitions. The *mincut* objective can be modified to overcome this problem by involving the size of the subsets $\mathcal{A}_k$ into the *mincut* objective. The size of a subset can be measured either by using the cardinality of the subset $|\mathcal{A}_k|$ or by summing up the weights of all edges attached to the vertices that belong to the subset $vol(\mathcal{A}_k) = \sum_{i \in \mathcal{A}_k} d_i$. The former leads to the *RatioCut* [4] while the latter results in the *Ncut* objective [1]. Only *Ncut* graph partition will be discussed in the following.

Based on the above discussion, the *Ncut* objective [1] can be written as

$$\text{Ncut}\{A_1, ..., A_\mathcal{K}\} = \sum_{k=1}^{\mathcal{K}} \sum_{i \in \mathcal{A}_k, j \notin \mathcal{A}_k} \frac{[S]_{ij}}{\text{vol}(\mathcal{A}_k)} \tag{1}$$

One can introduce $\mathbf{f}_k \in \{0, 1\}^N, k = 1, ..., \mathcal{K}$ binary cluster indicator vectors, such that the i-th component of $\mathbf{f}_k$ is 1 if the i-th data point belongs to the k-th cluster $i \in \mathcal{A}_k$ and zero otherwise. The cluster indicator matrix $F \in \{0, 1\}^{N \times \mathcal{K}}$ is formed by these mutually orthogonal $\mathbf{f}_k$ cluster indicator vectors. Defining $\mathbf{h}_k = D^{\frac{1}{2}} \mathbf{f}_k / \|D^{\frac{1}{2}} \mathbf{f}_k\|_2, k = 1, ..., \mathcal{K}$ vectors and $H = [\mathbf{h}_1, ..., \mathbf{h}_\mathcal{K}]$ orthonormal matrix, the *Ncut* objective given by Eq. (1) can be written as

$$\begin{aligned} \text{Ncut}\{A_1, ..., A_\mathcal{K}\} &= \sum_{k=1}^{\mathcal{K}} \left(1 - \sum_{i \in \mathcal{A}_k, j \in \mathcal{A}_k} \frac{[S]_{ij}}{\text{vol}(\mathcal{A}_k)} \right) \\ &= \mathcal{K} - Tr(H^T D^{-\frac{1}{2}} S D^{-\frac{1}{2}} H) \end{aligned} \tag{2}$$

The *Ncut* minimization problem then can be reformulated as the following trace maximization

$$\begin{aligned} \underset{H}{\text{argmin}}\, \text{Ncut}\{A_1, ..., A_\mathcal{K}\} \quad &= \quad \underset{H}{\text{argmax}}\, Tr(H^T L_{sym} H) \\ \text{such that} \qquad & \qquad H^T H = I_\mathcal{K} \end{aligned} \tag{3}$$

by introducing $L_{sym} = D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$ (symmetric) normalized graph Laplacian [6]. This optimization problem is NP-complete due to the discrete constraint on H [1]. However, relaxing this discrete constraint by allowing H to be a real valued matrix $\tilde{H} \in \mathbb{R}^{N \times \mathcal{K}}$ yields [20]

$$\begin{aligned} \underset{\tilde{H}}{\operatorname{argmin}} \operatorname{Ncut}\{A_1, ..., A_{\mathcal{K}}\} &= \underset{\tilde{H}}{\operatorname{argmax}} Tr(\tilde{H}^T L_{sym} \tilde{H}) \\ \text{such that} \qquad & \tilde{H}^T \tilde{H} = I_{\mathcal{K}} \end{aligned} \tag{4}$$

According to the Ky Fan theorem[21], a solution of this relaxed trace maximization problem can be given by taking $\boldsymbol{\alpha}_k, k = 1, ..., \mathcal{K}$ eigenvectors, associated to the $\mathcal{K}$ highest eigenvalues λ_k, of the $L_{sym} = D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$ matrix as columns of $\tilde{H}$. Hence, a continuous solution of the *Ncut* partitioning problem can be obtained by the solution of the $L_{sym}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$ eigenvalue problem.

Multiplying both side from the left by $D^{-\frac{1}{2}}$ and introducing $\boldsymbol{\beta} = D^{-\frac{1}{2}}\boldsymbol{\alpha}$ vectors, the above eigenvalue problem can be transformed to $L_{rw}\boldsymbol{\beta} = \lambda\boldsymbol{\beta}$ where $L_{rw} = D^{-1}S$ is the so-called *random walk* normalized graph Laplacian [6]. Its name expresses the strong relation of this normalized graph Laplacian to a Markov random walk on a graph[22] [5]. L_{rw} is a row stochastic matrix with ij-th entries representing the probability of moving from the i-th vertex to the j-th in one step of the random walk. Therefore, L_{rw} can be interpreted as a stochastic transition matrix and plays a key role in the Markov random walk interpretation of clustering. Within this representation, the *Ncut* minimization corresponds to looking for a partition of the graph such that the random walk, described by L_{rw} as transition matrix, remains most of the time on vertices that belong to the same subset of the partition and there are only a few jumps between vertices that belong to different subsets of the partition [5] [6].

The last step of the clustering, either by solving the eigenvalue problem including the symmetric L_{sym} matrix or the generalized eigenvalue problem derived from the non-symmetric L_{rw} matrix, is to convert the continuous relaxed solutions $\boldsymbol{\alpha}^{(k)}, k = 1, ..., \mathcal{K}$ or $\boldsymbol{\beta}^{(k)}, k = 1, ..., \mathcal{K}$ to discrete cluster indicators. This is usually done by applying simple K-means on the N, $\mathcal{K}$-dimensional points formed by the corresponding components of the obtained $\mathcal{K}$ leading eigenvectors. The latter including L_{rw} gives the spectral clustering algorithm of *Shi and Malik* [1] while the former involving L_{sym} leads to the *NJW* [3] algorithm.

3. Kernel spectral clustering through weighted kernel PCA

A multiway kernel spectral clustering algorithm with out-of-sample extension capability was introduced in [11] based on primal-dual least-square support vector machine (LS-SMV) [14, 13] formulation of weighted kernel PCA. The algorithm will be reviewed in this section together with some important properties of the solutions and the corresponding score variables that are essential for the kernel spectral clustering interpretation of the weighted kernel PCA. Furthermore, a new cluster membership encoding-decoding scheme and a model selection criterion will be proposed based on some of these properties.

3.1. LS-SVM formulation of weighted kernel PCA

Given an input data set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d$ together with the corresponding weights $\mathcal{V} = \{v_l\}_{l=1}^N, v_l \in \mathbb{R}^+$, the goal of weighted kernel PCA is to find the vectors $\mathbf{w}^{(k)}, k = 1, \ldots, \mathcal{K} < N$ such that the weighted variance of the projections of the weight centered feature map onto these $\mathbf{w}^{(k)}$ vectors is maximal. Following the LS-SVM formulation of the unweighted version of the kernel PCA problem, given in [13], the weighted kernel PCA optimization can be formulated as follows:

$$\max_{\mathbf{w}^{(k)}, b^{(k)}} \sum_{k=1}^{\mathcal{K}} \sum_{l=1}^{N} v_l [0 - ({\mathbf{w}^{(k)}}^T \varphi(\mathbf{x}_l) + b^{(k)})]^2 \tag{5}$$

where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is the non-linear mapping of the d dimensional input data to a n_h high-dimensional feature space and $b^{(k)} \in \mathbb{R}, k = 1, \ldots, \mathcal{K}$ are bias terms [1]. This maximization problem can be written as

$$\max_{\mathbf{w}^{(k)}, \mathbf{e}^{(k)}, b^{(k)}} \sum_{k=1}^{\mathcal{K}} {\mathbf{e}^{(k)}}^T V \mathbf{e}^{(k)} \tag{6}$$

by introducing $V \in \mathbb{R}^{N \times N}, [V]_{ii} = v_i, i = 1, ..., N$ diagonal weight matrix, $\Phi \in \mathbb{R}^{N \times n_h}$ feature map matrix as

$$\Phi = \begin{bmatrix} \varphi^T(\mathbf{x}_1) \\ \vdots \\ \varphi^T(\mathbf{x}_N) \end{bmatrix} \tag{7}$$

and $\mathbf{e}^{(k)} \in \mathbb{R}^N, k = 1, \ldots, \mathcal{K}$ error vectors as

$$\begin{aligned} \mathbf{e}^{(k)} &= \Phi \mathbf{w}^{(k)} + b^{(k)} \mathbb{1}_N \\ \mathbf{e}^{(k)} &= \begin{bmatrix} e_1^{(k)} \\ \vdots \\ e_N^{(k)} \end{bmatrix} = \begin{bmatrix} {\mathbf{w}^{(k)}}^T \varphi(\mathbf{x}_1) + b^{(k)} \\ \vdots \\ {\mathbf{w}^{(k)}}^T \varphi(\mathbf{x}_N) + b^{(k)} \end{bmatrix} \end{aligned} \tag{8}$$

The corresponding primal optimization problem can be formulated as [11]

$$\begin{aligned} \max_{\mathbf{w}^{(k)}, \mathbf{e}^{(k)}, b^{(k)}} J(\mathbf{w}, \mathbf{e}, b) &= \frac{1}{2} \sum_{k=1}^{\mathcal{K}} \gamma^{(k)} {\mathbf{e}^{(k)}}^T V \mathbf{e}^{(k)} \\ &\quad - \frac{1}{2} \sum_{k=1}^{\mathcal{K}} {\mathbf{w}^{(k)}}^T \mathbf{w}^{(k)} \\ \textit{such that} \quad \mathbf{e}^{(k)} &= \Phi \mathbf{w}^{(k)} + b^{(k)} \mathbb{1}_N, \quad k = 1, ..., \mathcal{K} \end{aligned} \tag{9}$$

[1] It can be shown, that using the bias term approach leads to the same weight centered kernel matrix as the explicit weighted centering of the feature map (see A).

where $\gamma^{(k)}$ are real positive constants. The Lagrangian of this constrained optimization problem is

$$\begin{aligned}\mathcal{L}(\mathbf{w}^{(k)}, \mathbf{e}^{(k)}, b^{(k)}; \boldsymbol{\beta}^{(k)}) = \frac{1}{2}\sum_{k=1}^{\mathcal{K}} \gamma^{(k)} \mathbf{e}^{(k)^T} V \mathbf{e}^{(k)} \\ - \frac{1}{2}\sum_{k=1}^{\mathcal{K}} \mathbf{w}^{(k)^T}\mathbf{w}^{(k)} - \sum_{k=1}^{\mathcal{K}} \boldsymbol{\beta}^{(k)^T}(\mathbf{e}^{(k)} - \Phi \mathbf{w}^{(k)} - b^{(k)}\mathbb{1}_N)\end{aligned} \tag{10}$$

with $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^N, k = 1, ..., \mathcal{K}$ Lagrange multiplier vectors. The Karush-Kuhn-Tucker (KKT) optimality conditions yield

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}^{(k)}} = 0 \quad \rightarrow \quad \mathbf{w}^{(k)} = \Phi^T \boldsymbol{\beta}^{(k)} \\ \dfrac{\partial \mathcal{L}}{\partial \mathbf{e}^{(k)}} = 0 \quad \rightarrow \quad \boldsymbol{\beta}^{(k)} = \gamma^{(k)} V \mathbf{e}^{(k)} \\ \dfrac{\partial \mathcal{L}}{\partial b^{(k)}} = 0 \quad \rightarrow \quad \mathbb{1}_N^T \boldsymbol{\beta}^{(k)} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}^{(k)}} = 0 \quad \rightarrow \quad \mathbf{e}^{(k)} - \Phi \mathbf{w}^{(k)} - b^{(k)}\mathbb{1}_N = 0 \end{cases} \tag{11}$$

for $k = 1, ..., \mathcal{K}$. Eliminating the primal variables $\mathbf{w}^{(k)}$ and $\mathbf{e}^{(k)}$ by substituting the first and second ($\mathbf{e}^{(k)} = V^{-1}\boldsymbol{\beta}^{(k)}/\gamma^{(k)}$) equations into the fourth results in

$$\frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} - \Phi\Phi^T\boldsymbol{\beta}^{(k)} - \mathbb{1}_N b^{(k)} = 0, \quad k = 1, ..., \mathcal{K} \tag{12}$$

Then applying the third KKT condition gives the expression for the last primal variable $b^{(k)}$

$$\begin{aligned} \mathbb{1}_N^T\boldsymbol{\beta}^{(k)} = \gamma^{(k)}\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)} + \gamma^{(k)}\mathbb{1}_N^T V \mathbb{1}_N b^{(k)} = 0 \rightarrow \\ \mathbb{1}_N^T V \mathbb{1}_N b^{(k)} = -\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)} \rightarrow \\ b^{(k)} = -\frac{\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)}}{\mathbb{1}_N^T V \mathbb{1}_N}\end{aligned} \tag{13}$$

for $k = 1, ..., \mathcal{K}$. Substituting this back into Eq. (12) gives the following equations for the stationary points of the Lagrangian expressed in terms of the dual variables $\boldsymbol{\beta}^{(k)}$

$$\begin{aligned} \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} - \Phi\Phi^T\boldsymbol{\beta}^{(k)} + \frac{\mathbb{1}_N\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)}}{\mathbb{1}_N^T V\mathbb{1}_N} = 0 \rightarrow \\ \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} - \left[I_N - \frac{\mathbb{1}_N\mathbb{1}_N^T V}{\mathbb{1}_N^T V\mathbb{1}_N}\right]\Phi\Phi^T\boldsymbol{\beta}^{(k)} = 0\end{aligned} \tag{14}$$

$k = 1, ..., \mathcal{K}$, where I_N is the identity matrix. Multiplying both sides by the diagonal weight matrix V from the left and substituting $\lambda^{(k)} = 1/\gamma^{(k)}$ the solution can be expressed in terms of the dual variables $\boldsymbol{\beta}^{(k)}$ as the following eigenvalue problem involving the dual variables as eigenvectors

$$VM_v\Omega\boldsymbol{\beta}^{(k)} = \lambda^{(k)}\boldsymbol{\beta}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{15}$$

where

$$M_v = I_N - \frac{\mathbb{1}_N\mathbb{1}_N^T V}{\mathbb{1}_N^T V\mathbb{1}_N} \tag{16}$$

is the weighted centering matrix and $\Omega = \Phi\Phi^T$ is the kernel matrix with ij-th entry $[\Omega]_{ij} = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, ..., N$ and $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive definite kernel.

It should be noted, that in general Eq. (9) is a nonconvex problem, therefore the KKT optimality conditions are necessary but not sufficient [23] [11]. However, taking the stationary points of the Lagrangian i.e. $\boldsymbol{\beta}$ eigenvectors of the $VM_v\Omega$ matrix as the pool of candidate solutions, the objective, given by Eq. (6), can be maximized by selecting the $\mathcal{K}$ leading eigenvectors since

$$\begin{aligned}\sum_{k=1}^{\mathcal{K}} \mathbf{e}^{(k)^T} V\mathbf{e}^{(k)} &= \sum_{k=1}^{\mathcal{K}} \frac{\boldsymbol{\beta}^{(k)^T} V^{-1}}{\gamma^{(k)}} V \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} \\ &= \sum_{k=1}^{\mathcal{K}} \frac{1}{\gamma^{(k)^2}} \boldsymbol{\beta}^{(k)^T} V^{-1}\boldsymbol{\beta}^{(k)} = \sum_{k=1}^{\mathcal{K}} \lambda^{(k)^2}\end{aligned} \tag{17}$$

where the V^{-1}-normality (i.e. $\boldsymbol{\beta}^{(k)^T} V^{-1}\boldsymbol{\beta}^{(k)} = 1$) of the $\boldsymbol{\beta}^{(k)}$ eigenvectors of the $VM_v\Omega$ matrix was assumed.

The k-th projection of the l-th mapped input data $\varphi(\mathbf{x}_l)$ can be written as

$$\begin{aligned}z_l^{(k)} =& z^{(k)}(\mathbf{x}_l) = \mathbf{w}^{(k)^T}\varphi(\mathbf{x}_l) + b^{(k)} = \boldsymbol{\beta}^{(k)^T}\Phi\varphi(\mathbf{x}_l) + b^{(k)} = \\ & \sum_{i=1}^{N} \beta_i^{(k)} \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_l) + b^{(k)} = \sum_{i=1}^{N} \beta_i^{(k)} K(\mathbf{x}_i, \mathbf{x}_l) + b^{(k)}\end{aligned} \tag{18}$$

by using the first KKT condition given by Eqs. (11) and the kernel trick. The k-th score variable $\mathbf{z}^{(k)} \in \mathbb{R}^{N\times 1}$ is

$$\begin{aligned}\mathbf{z}^{(k)} =& \Phi\mathbf{w}^{(k)} + \mathbb{1}_N b^{(k)} = \Phi\Phi^T\boldsymbol{\beta}^{(k)} - \frac{\mathbb{1}_N\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)}}{\mathbb{1}_N^T V\mathbb{1}_N} = \\ & \left[I_N - \frac{\mathbb{1}_N\mathbb{1}_N^T V}{\mathbb{1}_N^T V\mathbb{1}_N}\right]\Phi\Phi^T\boldsymbol{\beta}^{(k)} = M_v\Omega\boldsymbol{\beta}^{(k)} = \lambda^{(k)}V^{-1}\boldsymbol{\beta}^{(k)}\end{aligned} \tag{19}$$

The weighted kernel PCA primal optimization problem Eqs. (9) transforms to an eigenproblem in the dual weight space Eq. (15) involving the dual variables $\boldsymbol{\beta}$ as eigenvectors of the weight centered weighted kernel matrix $VM_v\Omega$. The

dual solutions of the weighted kernel PCA problem can be obtained by taking the $\mathcal{K}$ leading eigenvectors of the $VM_v\Omega$ matrix. The effect of multiplying the kernel matrix Ω by the weighted centering matrix M_v is that the weighted mean will be removed from each columns of the kernel matrix. Note, that M_v becomes the centering matrix by choosing the weight matrix being equal to the identity matrix $V = I_N$ and Eq. (15) corresponds to the dual form of the unweighted kernel PCA problem discussed in [13].

According to section 2, taking the kernel matrix as a similarity matrix and choosing the weight matrix equal to the inverse degree matrix $V = D^{-1}$, Eq. (15) becomes

$$D^{-1}M_D\Omega\boldsymbol{\beta}^{(k)} = \lambda^{(k)}\boldsymbol{\beta}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{20}$$

This is similar to the random walk spectral clustering eigenproblem based on the random walk normalized Laplacian L_{rw} discussed in section 2. However, now the weight centered kernel matrix $M_D\Omega$ plays the role of the similarity matrix in the eigenproblem [11] where

$$M_D = I_N - \frac{\mathbb{1}_N\mathbb{1}_N^T D^{-1}}{\mathbb{1}_N^T D^{-1}\mathbb{1}_N} \tag{21}$$

Furthermore, Eq. (20) can be transformed to

$$D^{-\frac{1}{2}}M_D\Omega M_D^T D^{-\frac{1}{2}}\boldsymbol{\alpha}^{(k)} = \lambda^{(k)}\boldsymbol{\alpha}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{22}$$

by using the identity given by Eq. (66) then multiplying both sides by $D^{1/2}$ from the left and substituting $\boldsymbol{\alpha} = D^{1/2}\boldsymbol{\beta}$. The resulted eigenproblem is similar to the symmetric spectral clustering eigenproblem based on the symmetric normalized Laplacian L_{sym} discussed in section 2 with the difference that now the symmetric weight centered kernel matrix $M_D\Omega M_D^T$ plays the role of the similarity matrix in the eigenproblem.

Therefore, a close relation can be shown between the spectral clustering algorithms, both involving the random walk L_{rw} and the symmetric normalized Laplacian L_{sym}, and the weighted kernel PCA by choosing the weight matrix as $V = D^{-1}$ [10]. However, some properties of the $\mathcal{K}$ leading eigenvectors of Eqs. (20) and (22) will be different than the corresponding eigenvectors of L_{rw} and L_{sym} due to the weighted centering of the kernel matrix. These properties will be discussed in the next section.

3.2. Properties of the eigenvectors and the score variables

Some properties of the eigenvectors of $D^{-1}M_D\Omega$ matrix and the corresponding score variables will be discussed in the following. These properties are essential for the spectral clustering interpretation of the weighted kernel PCA and for the formulation of the clustering algorithm.

Let assume that there are $\mathcal{K}$ well formed clusters in the N input data and the corresponding perfect partition of this input data set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d$ is $\Delta = \{C_1, ..., C_\mathcal{K}\}$. $i \in C_k$ will be used as a shorthand for $\mathbf{x}_i \in C_k$. Furthermore, it is assumed that this structure of the input data is well represented by the

corresponding similarity graph and the kernel matrix i.e. the similarity graph consists of $\mathcal{K}$ disconnected components that results in corresponding kernel matrix $[\Omega]_{ij} \neq 0 \iff i,j \in C_k \ \forall k = 1,...,\mathcal{K},\ i,j = 1...N$. This case, when the clusters are well formed, is often referred as ideal case [3].

3.2.1. Piecewise constant eigenvectors

In the ideal case, the algebraic multiplicity of the eigenvalue 1 of the $D^{-1}\Omega$ matrix is equal to the number of disconnected components in the corresponding similarity graph. It was shown in [5], that these $\mathcal{K}$ leading eigenvectors of the $D^{-1}\Omega$ matrix are *piecewise constant* with respect to the partition $\Delta = \{C_1,...,C_\mathcal{K}\}$ when the corresponding $\mathcal{K}$ clusters are well formed due to the *block-stochastic* property of the $D^{-1}\Omega$ matrix. A vector $\mathbf{x} \in \mathbb{R}^N$ is *piecewise constant* relative to a partition $\Delta = \{C_1,...,C_\mathcal{K}\}$ of the $i = 1,...,N$ indices if and only if $x_i = x_j,\ \forall i,j \in C_k,\ k = 1,...,\mathcal{K}$. A matrix P, indexed by $i = 1,...,N$, is block-stochastic with respect to a partition $\Delta = \{C_1,...,C_\mathcal{K}\}$ of the indices if $P_{ik} = \sum_{j \in C_k} P_{ij}$ are constant $\forall\ i \in C_{k'},\ \forall k,k' = 1,...,\mathcal{K}$ and the $R \in \mathbb{R}^{\mathcal{K}\times\mathcal{K}}$ matrix with $P_{k'k}$ entries are non-singular [5][2].

Although the $D^{-1}M_D\Omega$ matrix is not block-stochastic, one can show that it has $\mathcal{K}-1$ eigenvalues equal to 1 and the corresponding eigenvectors are piecewise constant on the underlying perfect partition of the input data.

The i,j-th elements of the $D^{-1}M_D\Omega$ matrix can be written as

$$[D^{-1}M_D\Omega]_{ij} = \frac{1}{d_i}\left\{[\Omega]_{ij} - \frac{\sum_{l=1}^N \frac{1}{d_l}[\Omega]_{lj}}{\sum_{l=1}^N \frac{1}{d_l}}\right\} \tag{23}$$

and the sum of the i-th row is

$$\begin{aligned}\sum_{j=1}^N [D^{-1}M_D\Omega]_{ij} &= \frac{\sum_{j=1}^N [\Omega]_{ij}}{d_i} - \frac{1}{d_i\epsilon}\sum_{l=1}^N \frac{\sum_{j=1}^N [\Omega]_{lj}}{d_l} \\ &= 1 - \frac{N}{d_i\epsilon}\end{aligned} \tag{24}$$

where $\epsilon = \sum_{i=1}^N 1/d_i$. It shows that unlike $D^{-1}\Omega$ the $D^{-1}M_D\Omega$ matrix is not block-stochastic since $d_i,\ i = 1,...,N$ are usually different for the different rows. However in the ideal case, if $\boldsymbol{\beta}^{(l)} \in \mathbb{R}^N$ is an eigenvector of the $D^{-1}M_D\Omega$ matrix with associated eigenvalue $\lambda^{(l)} = 1$, then it is also an eigenvector of the $D^{-1}\Omega$ matrix with the same eigenvalue. In order to see this, one first need to show that the $b^{(l)}$ bias terms, associated to $\boldsymbol{\beta}^{(l)}$ eigenvectors with $\lambda^{(l)} = 1$, are zero in the ideal case. This can be shown by using the KKT conditions given in

[2]Note, that in the ideal case $D^{-1}\Omega$ is block diagonal that is a block-stochastic matrix with $R = I_\mathcal{K}$.

Eqs. (11) and taking into account that now $V = D^{-1}$

$$\mathbb{1}_N b^{(l)} = \mathbf{e}^{(l)} - \Phi \mathbf{w}^{(l)} = \frac{D\boldsymbol{\beta}^{(l)}}{\gamma^{(l)}} - \Phi\Phi^T\boldsymbol{\beta}^{(l)} = [\lambda^{(l)} D - \Omega]\boldsymbol{\beta}^{(l)} \tag{25}$$

Multiplying both side by $\mathbb{1}_N^T/N$ from the left yields

$$\begin{aligned} b^{(l)} &= \frac{1}{N}(\lambda^{(l)} - 1)[d_1, \ldots, d_N]\boldsymbol{\beta}^{(l)} = \frac{1}{N}(\lambda^{(l)} - 1)\sum_{i=1}^{N} d_i \beta_i^{(l)} \\ &= \frac{1}{N}(\lambda^{(l)} - 1)\mathbb{1}_N^T D\boldsymbol{\beta}^{(l)}, l = 1, \ldots, \mathcal{K} - 1 \end{aligned} \tag{26}$$

That gives zero since $\lambda^{(l)} = 1$. Substituting the fourth KKT condition into the second results in

$$\boldsymbol{\beta}^{(l)} = D^{-1}\mathbf{e}^{(l)} = D^{-1}\Phi\Phi^T\boldsymbol{\beta}^{(l)} = D^{-1}\Omega\boldsymbol{\beta}^{(l)} \tag{27}$$

by using $1/\gamma^{(l)} = \lambda^{(l)} = 1$ and $b^{(l)} = 0$. Thus in the ideal case, if $\boldsymbol{\beta}^{(l)} \in \mathbb{R}^N$ is an eigenvector of the $D^{-1}M_D\Omega$ matrix with associated eigenvalue $\lambda^{(l)} = 1$, then it is also an eigenvector of the $D^{-1}\Omega$ matrix with the same eigenvalue.

According to [5], the eigenvectors of the $D^{-1}\Omega$ matrix, with associated eigenvalue equal to 1, are piecewise constants on the clusters in the ideal case i.e. $\beta_i^{(l)} = \beta_j^{(l)} = \xi_k^{(l)}, \forall i, j \in C_k, k \in \{1, \ldots, \mathcal{K}\}$. This can also be seen by writing $D^{-1}\Omega\boldsymbol{\beta}^{(l)} = \boldsymbol{\beta}^{(l)}$ for the components of $\boldsymbol{\beta}^{(l)}$

$$\frac{1}{d_i}\sum_{j=1}^{N}[\Omega]_{ij}\beta_j^{(l)} = \beta_i^{(l)}, \quad \forall i = 1, \ldots, N \tag{28}$$

and assuming that the i-th input data belongs to the k-th cluster

$$\sum_{j \in C_k}[\Omega]_{ij}\beta_j^{(l)} - d_i\beta_i^{(l)} = \sum_{j \in C_k}[\Omega]_{ij}\beta_j^{(l)} - \sum_{j \in C_k}[\Omega]_{ij}\beta_i^{(l)} = 0 \tag{29}$$

This equation must hold for $\forall i \in C_k, k \in 1, \ldots, \mathcal{K}$ that indicates $\beta_i^{(l)} = \beta_j^{(l)} = \xi_k^{(l)}, \forall i, j \in C_k, k \in \{1, \ldots, \mathcal{K}\}$.

Moreover, the centered property of these piecewise constant eigenvectors $\sum_{i=1}^{N}\beta_i^{(l)} = \sum_{k=1}^{\mathcal{K}} n_k\xi_k^{(l)} = 0$ shows that $\mathcal{K} - 1$ out of the $\mathcal{K}$, $\xi_k^{(l)}$ constants, that form the given eigenvector, can be chosen independently. Hence, the eigenspace of the $D^{-1}M_D\Omega$ matrix associated to $\lambda = 1$ eigenvalue is a $\mathcal{K} - 1$ dimensional space spanned by the corresponding $\mathcal{K} - 1$, D-orthogonal eigenvectors. Thus these $\mathcal{K} - 1$ eigenvectors are centered, piecewise constant on the underlying clusters. Therefore each of the $\mathcal{K}$ well formed clusters are represented as single points in the $\mathcal{K} - 1$ dimensional eigenspace of the $D^{-1}M_D\Omega$ matrix associated to $\lambda = 1$ eigenvalue.

3.2.2. Cluster membership is coded in signs

It has been shown above, that $D^{-1}M_D\Omega$ matrix has $\mathcal{K}-1$ eigenvectors with eigenvalue equal to 1 if the underlying data set consists of $\mathcal{K}$ well formed clusters and these $\mathcal{K}-1$ leading eigenvectors are piecewise constant on the perfect partition of the input data. The eigenvectors of the $D^{-1}M_D\Omega$ matrix are mutually D-orthogonal where D is the diagonal degree matrix with positive diagonal entries as it was shown in section 3.1. According to the third KKT condition Eqs. (11), these eigenvectors are centered. The signs of the constants, that form the $\mathcal{K}-1$ leading eigenvectors, satisfy the following two criteria in order to ensure that the eigenvectors fulfil the centered and mutually D-orthogonal properties: (i) at least one out of the $\mathcal{K}$ constants, that form the individual eigenvectors, has different sign than the others; (ii) there are not two vectors among these $\mathcal{K}-1$ eigenvectors that have the same signs in each case of their corresponding constants. Therefore, each of the $\mathcal{K}$ well formed clusters are represented as $\mathcal{K}$ single points in the $\mathcal{K}-1$ dimensional eigenspace of $D^{-1}M_D\Omega$ matrix associated to $\lambda=1$ eigenvalue and these $\mathcal{K}$ points, that correspond to different clusters, are located in different orthant of this $\mathcal{K}-1$ dimensional eigen-subspace.

3.2.3. Some results of perturbation analysis

The ideal case, when the perfect partition of the input data set into $\mathcal{K}$ disjoint subsets appears in the similarity graph as $\mathcal{K}$ disconnected subgraphs, has been considered so far. The corresponding similarity matrix Ω is a block diagonal matrix in this ideal case. It has been shown above, that each of the $\mathcal{K}$ disconnected components are represented as single points in the $\mathcal{K}-1$ dimensional eigenspace of $D^{-1}M_D\Omega$ matrix associated to $\lambda=1$ eigenvalue due to the fact that these eigenvectors are piecewise constant on the $\mathcal{K}$ disconnected components. Therefore, the piecewise constant property of these $\mathcal{K}-1$ leading eigenvectors is crucial in spectral clustering since it makes possible to transform the original clustering problem given in the input space to a trivial clustering problem in the $\mathcal{K}-1$ dimensional eigen-subspace of $D^{-1}M_D\Omega$ matrix associated to $\lambda=1$ eigenvalue. Then the question is, how this piecewise constant property of the $\mathcal{K}-1$ leading eigenvectors change in the non-ideal case when there are edges between the $\mathcal{K}$ subgraphs of the ideal case resulting into a non-block diagonal similarity matrix.

Perturbation analysis can provide an answer in the nearly ideal case, when there still are distinct clusters in the data set but the similarities between data points that belong to different clusters are not necessarily zero but small compared to the within cluster similarities. The difference between the nearly ideal and ideal case can be considered as a small perturbation. According to the Davis-Kahan theorem, if a symmetric matrix A is modified by an additive perturbation H, the distance between the eigenspaces corresponding to the first k eigenvalues of A and the first k eigenvalues of the perturbed matrix $\tilde{A}=A+H$ is bounded by $\|H\|_F/\delta_k$, where $\|\cdot\|_F$ denotes the Frobenius norm, $\delta_k=\lambda_k-\lambda_{k+1}$ is the additive eigengap and λ_k is the k-th eigenvalue of A if they are in descend-

ing order[3]. It means, that the larger the gap between λ_k and λ_{k+1} is, the closer the k leading eigenvectors of the perturbed matrix $\tilde{A}$ and the original matrix A are.

The consequence of the Davis-Kahan theorem is that if the perturbation is small and the eigengap between the $\mathcal{K}-1$-th and $\mathcal{K}$-th ordered eigenvalues of the $D^{-1}M_D\Omega$ matrix is large enough, the first $\mathcal{K}-1$ leading eigenvectors will be very close to the corresponding eigenvectors of the ideal case. Therefore, the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix will be approximately piecewise constant on the clusters even if the between cluster similarities are not necessarily zero (but small compared to the within cluster similarities) and/or the eigengap is large enough. The $\mathcal{K}$ clusters will be represented in the $\mathcal{K}-1$ dimensional space, spanned by these $\mathcal{K}-1$ leading eigenvectors, as $\mathcal{K}$ clouds around the $\mathcal{K}$ corresponding points of the ideal case. The closer to the ideal case we are, the smaller the spreading of the individual clouds is.

3.2.4. Properties of the score variables

Some important properties of the score variables will be discussed in the following. The ideal case, when the similarity matrix Ω has a block diagonal form, will be discussed first then the deviations caused by perturbations will also be addressed as in case of the eigenvectors. It was shown above, that the $D^{-1}M_D\Omega$ matrix has $\mathcal{K}-1$ eigenvectors with eigenvalue equal to 1 when the similarity graph consists of $\mathcal{K}$ disconnected components and these eigenvectors are piecewise constants on the disconnected components.

Property 1. *Every entry in the k-th score variable has the same sign as the corresponding entry in the k-th eigenvector ($k = 1, \ldots, \mathcal{K}-1$).*

It can be seen from Eq. (19) by substituting $V = D^{-1}$ and taking into account that $\lambda^{(k)} > 0, \forall k \in \{1, \ldots, \mathcal{K}-1\}$ and D is a diagonal matrix with positive diagonal entries.

Property 2. $\mathbf{z}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ *score variables are D^{-1}-orthogonal to the constant vector $\mathbb{1}_N$ i.e. the weighted mean is removed from the score variables.*

According to Eq. (19), the weighted mean of the score variables can be written as

$$\frac{1}{\epsilon}\mathbb{1}_N^T D^{-1}\mathbf{z}^{(k)} = \frac{\lambda^{(k)}}{\epsilon}\underbrace{\mathbb{1}_N^T\boldsymbol{\beta}^{(k)}}_{0} = 0, \forall k \in \{1, \ldots, \mathcal{K}-1\} \tag{30}$$

by choosing the weight matrix as $V = D^{-1}$ and using the centered property of the $\boldsymbol{\beta}$ eigenvectors i.e. the third KKT condition from Eqs. (11).

[3]Although $D^{-1}M_D\Omega$ is not a symmetric matrix, it can always be transformed to the corresponding symmetric problem and vice versa (see Eq. (22)).

Property 3. $\mathbf{z}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ *score variables are mutually* D^{-1}*-orthogonal.*

According to Eq. (19)

$$\begin{aligned}
\mathbf{z}^{(i)^T} D^{-1} \mathbf{z}^{(j)} &= [\lambda^{(i)} D\boldsymbol{\beta}^{(i)}]^T D^{-1} [\lambda^{(j)} D\boldsymbol{\beta}^{(j)}] \\
&= \lambda^{(i)}\lambda^{(j)} \boldsymbol{\beta}^{(i)^T} \underbrace{DD^{-1}}_{I_N} D\boldsymbol{\beta}^{(j)} \\
&= \lambda^{(i)}\lambda^{(j)} \underbrace{\boldsymbol{\beta}^{(i)^T} D\boldsymbol{\beta}^{(j)}}_{=0 \; if \; i \neq j} \\
&= 0 \quad if \; i \neq j \; \forall \, i, j = 1 ... \mathcal{K}-1
\end{aligned} \tag{31}$$

where the mutual D-orthogonality of the $\boldsymbol{\beta}$ eigenvectors of the $D^{-1}M_D\Omega$ was used.

Property 4. *The* $\mathcal{K}-1$ *dimensional score points, that correspond to input data points from the same cluster, are collinear in the* $\mathcal{K}-1$ *dimensional score space spanned by the score variables* $z^{(k)}, k = 1, \ldots, \mathcal{K}-1$.

Let $\mathbf{z}_i^* \in \mathbb{R}^{1\times(\mathcal{K}-1)}$ denote the i-th row of the $N \times (\mathcal{K}-1)$ dimensional score matrix. The k-th component of $\mathbf{z}_i^*$ can be given by Eq. (18). In the ideal case, the similarity matrix is block diagonal and the $\mathcal{K}-1$ leading eigenvectors $\boldsymbol{\beta}$ of the $D^{-1}M_D\Omega$ matrix are piecewise constants on the underlying $\mathcal{K}$ clusters i.e. $K(\mathbf{x}_i, \mathbf{x}_j) \neq 0 \iff i, j \in C_p, p \in \{1, \ldots, \mathcal{K}\}$ and $\beta_i^{(k)} = \beta_j^{(k)} = \xi_p^{(k)}, \forall i, j \in C_p, p \in \{1, \ldots, \mathcal{K}\}, k \in \{1, \ldots, \mathcal{K}-1\}$. Assuming, that the i-th input data point $\mathbf{x}_i$ belongs to the p-th cluster i.e. $i \in C_p, p \in \{1, \ldots, \mathcal{K}\}$, the k-th component of the corresponding $\mathbf{z}_i^*$ can be written as

$$\begin{aligned}
z_i^{*(k)} &= \sum_{j=1}^{N} \beta_j^{(k)} K(\mathbf{x}_j, \mathbf{x}_i) + b^{(k)} = \sum_{j \in C_p} \beta_j^{(k)} K(\mathbf{x}_j, \mathbf{x}_i) \\
&\quad + \sum_{j \notin C_p} \beta_j^{(k)} K(\mathbf{x}_j, \mathbf{x}_i) + b^{(k)} \\
&= \xi_p^{(k)} \underbrace{\sum_{j \in C_p} K(\mathbf{x}_j, \mathbf{x}_i)}_{=d_i} + \sum_{j \notin C_p} \beta_j^{(k)} \underbrace{K(\mathbf{x}_j, \mathbf{x}_i)}_{=0 \; \forall j \notin C_p \; i \in C_p} + b^{(k)} \\
&= d_i \xi_p^{(k)} + b^{(k)}, \\
&\forall i \in C_p, p \in \{1, \ldots, \mathcal{K}\}, \forall k \in \{1, \ldots, \mathcal{K}-1\}
\end{aligned} \tag{32}$$

and

$$\begin{aligned}
\mathbf{z}_i^* =& d_i [\xi_p^{(1)}, \ldots, \xi_p^{(\mathcal{K}-1)}] + [b^{(1)}, \ldots, b^{(\mathcal{K}-1)}] \\
& \forall i \in C_p, p \in \{1, \ldots, \mathcal{K}\}
\end{aligned} \tag{33}$$

Therefore, the $\mathcal{K}-1$ dimensional score vectors $\mathbf{z}_i^*$, that correspond to input data from the same cluster, point to the same direction in the $\mathcal{K}-1$ dimensional score space in the ideal case. Furthermore, the $\mathcal{K}$ directions, that correspond to the $\mathcal{K}$ different clusters, lie in different orthants of the $\mathcal{K}-1$ dimensional score space due to the fact that the cluster membership is coded in the signs of the $\xi_p^{(k)}$ $k=1,\ldots,\mathcal{K}-1$ constants that form the $\boldsymbol{\beta}$ eigenvectors. Therefore, the $\mathcal{K}-1$ dimensional points $\mathbf{z}_i^*$, that correspond to input data points from the same clusters, are collinear in the $\mathcal{K}-1$ dimensional space spanned by the $\mathcal{K}-1$ score vectors correspond to the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix and the $\mathcal{K}$ distinct directions lie in different orthants of this score space.

According to section 3.2.3, the similarity matrix becomes only nearly block diagonal if the between cluster similarities are not necessarily zero but small compare to the within cluster similarities. As a consequence, the contribution of the summation in Eq. (32) according to $j \notin C_p$ is not zero but still much smaller than the corresponding contribution of $j \in C_p$. Furthermore, the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix remain approximately piecewise constants on the clusters in the presence of such a small perturbation (and/or in case of large eigengap). The $\mathcal{K}$ clusters are represented as $\mathcal{K}$ well separated clouds in the $\mathcal{K}-1$ dimensional space, spanned by the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix, scattered around the corresponding points of the ideal case. The closer to the ideal case we are, the smaller the spreading of the individual clouds is. Therefore, the $\mathcal{K}-1$ dimensional points $\mathbf{z}_i^*$, that correspond to input data points from the same clusters, are still approximately collinear in the $\mathcal{K}-1$ dimensional space spanned by the $\mathcal{K}-1$ score vectors correspond to the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix. The closer to the ideal case we are, the smaller the scatter of the $\mathcal{K}$ point clouds in the corresponding eigenspace is and more collinear the score points are.

Note, that the special features of the ideal case was exploited only in the case of the last property of the score variables.

3.3. Cluster membership encoding and decoding

In order to obtain a clustering model, based on the weighted kernel PCA formulation, one need to select a cluster encoding-decoding scheme that makes possible the cluster membership assignment. In the case of classical spectral clustering algorithms [1, 3] this step is usually done by applying K-means in the appropriate eigen-subspace as it was discussed at the end of section 2. The weighted kernel PCA formulation, the properties of the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix and the corresponding score variables discussed in the previous sections, provide different possible encoding schemes.

One, that was introduced in [11], is based on the fact that the cluster membership is coded in the signs of the corresponding components of the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix as it was discussed in section 3.2.2. Therefore, each cluster $C_k, k=1,\ldots,\mathcal{K}$ can be represented by a unique codeword $\mathbf{c}^{(k)} \in \{+1,-1\}^{(\mathcal{K}-1)}$. The candidate codewords can be obtained by converting

the rows of the matrix formed by the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix to $\mathcal{K}-1$ dimensional sign vectors. According to Property 1. of the score variables discussed in section section 3.2.4, the score variables can also be used instead of the eigenvectors. Based on section 3.2.2 and 3.2.3, there are exactly $\mathcal{K}$ different codewords in the ideal case and can be more than $\mathcal{K}$ in strongly perturbed cases. Thus, a codebook $\mathcal{CB} = \{\mathbf{c}^{(k)}\}_{k=1}^{\mathcal{K}}$ is formed by taking the $\mathcal{K}$ most frequent codewords. Then each point is clustered based on the corresponding $\mathcal{K}-1$ dimensional score point and this codebook. The score point is converted to a sign vector and compared to the elements of the codebook by calculating the Hamming distances. The codeword, yielding the smallest Hamming distance, is selected and the point is assigned to the corresponding cluster [11].

A different encoding-decoding scheme can also be obtained based on angular similarities. According to Eq. (18), the $\mathcal{K}-1$ dimensional score point $\mathbf{z}_i^*$, that corresponds to the i-th input data $\mathbf{x}_i$, can be written as the linear combination of $\mathbf{e}_j^* = [\beta_j^{(1)}, \dots, \beta_j^{(\mathcal{K}-1)}], j = 1, \dots, N$ vectors plus the bias vector $\mathbf{b} = [b^{(1)}, \dots, b^{(\mathcal{K}-1)}]$

$$\mathbf{z}_i^* = \sum_{j=1}^{N} K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{e}_j^* + \mathbf{b} \tag{34}$$

where $\boldsymbol{\beta}^{(k)}, k = 1, \dots, \mathcal{K}-1$ are the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix. In the ideal case, Eq. (34) transforms to Eq. (33) since $\mathbf{e}_i^* = \mathbf{e}_j^* = \mathbf{e}_k^* = [\xi_k^{(1)}, \dots, \xi_k^{(\mathcal{K}-1)}], \forall i, j \in C_k, k \in \{1, \dots, \mathcal{K}\}$ due to the piecewise constant property of the $\boldsymbol{\beta}^{(k)}, k = 1, \dots, \mathcal{K}-1$ eigenvectors. The $\mathcal{K}$ clusters are represented by the $\mathbf{e}_k^*, k = 1, \dots, \mathcal{K}$ vectors as $\mathcal{K}$ single points in different orthants of the $\mathcal{K}-1$ dimensional space spanned by the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix and

$$\begin{aligned} \mathbf{z}_i^* - \mathbf{b} = \sum_{j=1}^{N} K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{e}_j^* = \sum_{j \in C_k} K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{e}_j^* = d_i\mathbf{e}_k^* \\ \forall i \in C_k, k \in \{1, \dots, \mathcal{K}\} \end{aligned} \tag{35}$$

by using Eq. (34) and the $K(\mathbf{x}_i, \mathbf{x}_j) \neq 0 \iff i, j \in C_p, p \in \{1, \dots, \mathcal{K}\}$ property of the kernel matrix in the ideal case. Therefore, $\mathbf{z}^* - \mathbf{b}$ vectors, that correspond to data points from the same cluster, point into the same direction and this direction is identical to the direction of the $\mathbf{e}_k^{(*)}$ vector that represents the given cluster in the $\mathcal{K}-1$ dimensional eigen-subspace. [4]

In perturbed cases, when the between cluster similarities are small compared to the within cluster similarities, the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix are approximately piecewise constants on the clusters. Each cluster is represented by the $\mathbf{e}_j^*, j \in C_k, k \in \{1, \dots, \mathcal{K}\}$ vectors as $\mathcal{K}$ well localized point clouds in different orthants of the $\mathcal{K}-1$ dimensional space spanned by the

[4] One could write $\mathbf{z}^*$ instead of $\mathbf{z}^* - \mathbf{b}$ because $\mathbf{b} = \mathbf{0}$ in the ideal case as it was shown in section 3.2.1.

$\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix. The $\mathbf{z}^*-\mathbf{b}$ vectors, that correspond to data points from the same cluster, will point approximately to the same directions and these directions will be close to the direction of the average of the $\mathbf{e}_j^*$ vectors associated to the given cluster.

An encoding-decoding scheme, based on these properties, can be obtained in the following way. The $\mathbf{e}_j^*$ vectors can easily be clustered either by using the sign coding of the cluster membership as above or by using simply K-means. Sign coding of the cluster membership can be used for initializing K-means that will terminate after a single iteration in the nearly ideal cases. After the $\mathbf{e}_j^*$ vectors are clustered based on their angular similarities, a $\mathcal{CB}=\{\mathbf{c}^{(k)}/\|\mathbf{c}^{(k)}\|_2\}_{k=1}^{\mathcal{K}}$ codebook can be formed by calculating an average $\mathbf{c}_k$ vector for each cluster as $\mathbf{c}^{(k)}=\sum_{j\in C_k}\mathbf{e}_j^*/|C_k|, k\in\{1,\ldots,\mathcal{K}\}$. Then each point can be clustered based on the corresponding $\mathbf{z}^*-\mathbf{b}$ vector derived from the score and this codebook. The $\mathbf{z}^*-\mathbf{b}$ vector is normalized and compared to the elements of the codebook by calculating either the projections to the codewords or the distances from the codewords. The codeword, yielding the maximum projection or the minimum distance, is selected and the point is assigned to the corresponding cluster.

3.4. Out-of-sample extension

Basic spectral clustering algorithms, discussed in section 2, provide a clustering only for the given data set without a clear extension to out-of-sample points i.e. to new points that are not part of the training data set. The most popular way of extending the clustering model for test points is based on the Nyström method [9]. An embedding of a test point can be achieved by using the Nyström approximation of the implicit eigenfunctions [7, 8]. The kernel spectral clustering algorithm formulated in the weighted kernel PCA framework provides a natural and simple way of out-of-sample extension without relying on approximation techniques such as the Nyström method [11]. The out-of-sample extension proposed in [11] is based on the codebook discussed in the previous section and the score variables associated to the out-of-sample points $\mathcal{D}^t=\{\mathbf{x}_i^t\}_{i=1}^{N_t}, \mathbf{x}_i^t\in\mathbb{R}^d$. These latter correspond the projections of the mapped out-of-sample points $\varphi(\mathbf{x}^t)$ to the $\mathbf{w}^{(k)}, k=1,\ldots,\mathcal{K}-1$ vectors obtained implicitly during the training stage. Since the problem is solved in the dual weight space, these projections can be calculated by means of $\boldsymbol{\beta}^{(k)}, k=1,\ldots,\mathcal{K}-1$ leading eigenvectors obtained in the training stage as

$$z(\mathbf{x}_i^t)^{(k)}=z_i^{t(k)}=\sum_{j=1}^{N}K(\mathbf{x}_j,\mathbf{x}_i^t)\beta_j^{(k)}+b^{(k)}, k=1,\ldots,\mathcal{K}-1 \tag{36}$$

by using Eq (18). The $\mathcal{K}-1$ dimensional out-of-sample score points $\mathbf{z}_i^{*t}\in\mathbb{R}^{(\mathcal{K}-1)}, i=1,\ldots,N_t$ have the same collinear property that was discussed in section 3.2.4. Therefore, the cluster membership of the out-of-sample points can be determined based on the corresponding $\mathbf{z}^{*t}$ vectors by comparing them

to the codebook, derived in the training stage as described in the previous section, using the appropriate decoding scheme.

3.5. Model selection

The quality of the clustering model, obtained in the training stage, strongly depends on the choice of the similarity function and its parameters. The most popular similarity function is the radial basis function kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\gamma)$ with γ parameter. Beyond the value of the γ kernel parameter, the optimality of the model also depends on the chosen number of clusters $\mathcal{K}$. Different $(\gamma, \mathcal{K})$ values result in different models and the closer the $\mathcal{K}-1$ leading eigenvectors of the associated $D^{-1}M_D\Omega$ matrix to being piecewise constant, the closer to the optimal the model is. Therefore, introducing a measure of the piecewise constantness of these leading eigenvectors results in a quality measure of the different models with different $(\gamma, \mathcal{K})$ values that can serve as a model selection criterion. Choosing the $(\gamma, \mathcal{K})$ value pair, that maximize this quality measure, makes possible to select the most optimal clustering model.

The *balanced line fit* (BLF) quality measure was proposed in [11]. As it was discussed in section 3.2.4, the $\mathbf{z}^* \in \mathbb{R}^{(\mathcal{K}-1)}$ score points, that corresponds to input data points from the same cluster, are collinear in the score space if the associated $\mathcal{K}-1$ leading eigenvectors of $D^{-1}M_D\Omega$ matrix are piecewise constants on the underlying $\mathcal{K}$ clusters. The line fit part of this criterion measures the collinearity of the $\mathbf{z}_i^{v*}, i \in C_k, k \in \{1, \ldots, \mathcal{K}\}$ score points, obtained by using an independent validation set $\mathcal{D}^v = \{\mathbf{x}_i\}_{i=1}^{N_v}$, by means of PCA. The balance of the obtained clusters is also incorporated into the BLF. More details on the BLF model selection criterion can be found in [11].

A model selection criterion, different from the BLF, can be obtained by using the cluster membership encoding-decoding scheme based on angular similarities discussed in the previous section. Using this encoding-decoding scheme, the cluster membership assignment is done based on the angular similarities between the normalized codewords in the codebook and the normalized $\mathbf{z}_i^{v*}$ vectors. Angular similarity between a normalized $\mathbf{z}_i^{v*}$ vector and the normalized codebook vectors can be measured either by calculating the projections or the distances between these $\mathcal{K}-1$ dimensional points. The individual projections are in the $[-1, +1]$ interval while the corresponding distances lie in the $[0, 2]$ interval. The codeword yielding the highest projection or the smallest distance is selected and the validation point, associated to the $\mathbf{z}_i^{v*}$ vector, is assigned to the corresponding cluster. The more collinear the $\mathbf{z}_i^{v*}, i \in C_k, k \in \{1, \ldots, \mathcal{K}\}$ points are, the higher(/smaller) the sum of these maximum projections/(minimum distances) is. Maximizing(/minimizing) the average projections(/distances) on the validation set D^v results in the selection of $(\gamma, \mathcal{K})$ parameters that correspond to the most optimal clustering model. One can further refine this quality measure in order to give more penalty to those validation points that lie closer to the decision boundary of the model. Let assume that the assignment is done by selecting the codeword from the codebook that yields the minimum distance between the normalized $\mathbf{z}_i^{v*}$ point and the codeword. Let denote this minimum

distance by $s_i^{(1)}$ and the second smallest distance by $s_i^{(2)}$. The corresponding validation point lies on a decision boundary of the model if $s_i^{(1)}/s_i^{(2)} = 1$ while the validation point is assigned to the cluster surely if $s_i^{(1)} = 0$. Therefore, one can select the most optimal model by maximizing the

$$\mathrm{Q}_m(D^v, \mathrm{model}(\gamma, \mathcal{K})) = \sum_{k=1}^{\mathcal{K}} \frac{1}{k} \sum_{i \in C_k} \frac{1 - s_i^{(1)}/s_i^{(2)}}{|C_k|} \tag{37}$$

quantity on a D^v validation set. The closer the $\mathrm{Q}_m \in [0, 1]$ quantity to 1, the better the corresponding clustering model is. One can place further emphasis on balancing the obtained clusters in the model selection by introducing a balance term similarly as in case of BLF [11]. The balanced quality measure can be written as

$$\begin{aligned} \mathrm{BQ}_m(D^v, \mathrm{model}(\gamma, \mathcal{K})) = \eta \mathrm{Q}_m(D^v, \mathrm{model}(\gamma, \mathcal{K})) \\ + (1 - \eta)\mathrm{balance}(D^v, \mathrm{model}(\gamma, \mathcal{K})) \end{aligned} \tag{38}$$

where

$$\mathrm{balance}(D^v, \mathrm{model}(\gamma, \mathcal{K})) = \frac{\min\{|C_1|, \ldots, |C_{\mathcal{K}}|\}}{\max\{|C_1|, \ldots, |C_{\mathcal{K}}|\}} \tag{39}$$

and $\eta \in [0, 1]$ is the importance given to Q_m with respect to the balance. It should be noted, that this model selection criterion can only be used for $\mathcal{K} > 2$.

4. Sparse algorithms

Non-sparse kernel spectral clustering algorithms need to store and to eigen-decompose an $N \times N$ matrix where N is the size of the input data set. The memory usage and computational time of these algorithms grow quickly by increasing the size of the input data set. Kernel spectral clustering models with ability to out-of-sample extension can reduce these memory and computational time problems by training the model on a subset of the input data set and using the out-of-sample extension capability of the model to infer the cluster membership of the unseen data.

4.1. Reduced set method

When the kernel spectral clustering problem is formulated as weighted kernel PCA, the problem is solved in the dual weight space. The solution consists of the determination of the $\boldsymbol{\beta}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix associated to the training set $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}} \subset \mathcal{D}$. According to the first KKT condition in Eqs (11), the primal variables $\mathbf{w}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ are expressed as linear combinations of the mapped input data $\mathbf{w}^{(k)} = \Phi^T\boldsymbol{\beta}^{(k)}, k = 1, \ldots, \mathcal{K}-1$. Since the components of the $\boldsymbol{\beta}^{(k)}$ eigenvectors are usually not zero, each training data point contributes to the primal variable $\mathbf{w}^{(k)}$ resulting in a non-sparse model. The goal is to find $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^{R} \subset \mathcal{D}^{tr}, R < N_{tr}$ reduced set

points and corresponding $\zeta^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients to approximate $\mathbf{w}^{(k)}$ primal variables as [18]

$$\mathbf{w}^{(k)} \approx \tilde{\mathbf{w}}^{(k)} = \sum_{r=1}^{R} \zeta_r^{(k)} \varphi(\tilde{\mathbf{x}}_r) = \Psi^T \zeta^{(k)}, \quad k = 1, \ldots, \mathcal{K}-1 \tag{40}$$

where $\Psi = [\varphi(\tilde{\mathbf{x}}_1), \ldots, \varphi(\tilde{\mathbf{x}}_R)]^T \in \mathbb{R}^{R \times n_h}$. An arbitrary data point $\mathbf{x} \in \mathbb{R}^d$ can be clustered based on the corresponding approximated score variables $\tilde{z}(\mathbf{x})^{(k)}, k = 1, \ldots, \mathcal{K}-1$ that can be calculated as [18]

$$z(\mathbf{x})^{(k)} \approx \tilde{z}(\mathbf{x})^{(k)} = \sum_{r=1}^{R} \zeta_r^{(k)} K(\mathbf{x}, \tilde{\mathbf{x}}_r) + \tilde{b}^{(k)} \tag{41}$$

$k = 1, \ldots, \mathcal{K}-1$ where $\tilde{b}^{(k)}$ are the approximated bias terms determined on the training set. The calculation of these approximated bias terms will be described in sections 4.2.1 and 4.2.4.

Introducing sparsity into the model as described above corresponds to the so called reduced set method [15, 16] that usually consists of three steps. First the size of the eigenvalue problem is reduced from the full size of the input data set N to $N_{tr} < N$ by selecting a subset for training $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}} \subset \mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$. When the required eigenvectors are derived based on the training set, a sparse representation of the model is constructed by the determination of the $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^{R} \subset \mathcal{D}^{tr}$ reduced set and the corresponding $\zeta^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients. Since usually $R \ll N_{tr} < N$, a sparse model representation can be obtained and used for calculating the approximated score variables associated to arbitrary input data points by using Eq. (41).

The training data set is usually selected by random sampling in case of Nyström approximation based kernel spectral clustering algorithms [8, 24]. Then the eigenvectors, derived based on the training set, are used to approximate the underlying eigenfunctions and to carry out the out-of-sample extension. Random sampling of the training set can also be used in case of reduced set methods. The sparsity of the final model will be higher in this case compared to the Nyström method based clustering models due to the application of the reduced set method. In spite of the higher level of sparsity, these clustering models outperforms the Nyström approximation based models trained on the same training set [11, 18]. A Rényi entropy maximization based training set selection and sparse kernel spectral clustering model based on weighted kernel PCA formulation was proposed in [18]. The training set is constructed in this case by selecting points from the full input data set that gradually increase the quadratic Rényi entropy of the selected subset [25, 14]. The required eigenvectors are determined based on the selected training set and the L_1 penalization [16, 26] is used to obtain the reduced set [18]. Then the corresponding reduced set coefficients are derived through the solution of a minimization [16, 11]. problem. Another sparse version of the weighted kernel PCA formulated kernel spectral clustering was proposed in [17] based on the incomplete Cholesky decomposi-

tion (ICD) and the reduced set method. This algorithm will be discussed in the following.

4.2. ICD based sparse algorithm

The incomplete Cholesky decomposition (ICD) and reduced set method based sparse version of the weighted kernel PCA formulated kernel spectral clustering algorithm will be discussed in this section. The original algorithm, proposed in [17], will be reviewed in sections 4.2.1 and 4.2.2. The computational bottlenecks of this algorithm will be shown and a modified, much faster algorithm will be proposed in section 4.2.4 that eliminates the computationally expensive parts of the original algorithm.

4.2.1. Approximated eigenvectors

The first step is to determine the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix associated to the randomly sampled training set $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}} \subseteq \mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$. The incomplete Cholesky decomposition (ICD) of the $\Omega \in \mathbb{R}^{N_{tr}\times N_{tr}}$ matrix is used to reduce the size of the eigenproblem [27, 28, 17].

Any symmetric positive definite matrix $A \in \mathbb{R}^{M\times M}$ can be decomposed as $A = LL^T$ where $L \in \mathbb{R}^{M\times M}$ is a lower triangular matrix. If the spectrum of A decays rapidly it has a small numerical rank [29] and A can be well approximated by GG^T where $G \in \mathbb{R}^{M\times R}, R \ll M$ [30]. This is the incomplete Cholesky factorization of A and it yields a decomposition in which the column space of GG^T is spanned by a subset of the columns of A [31]. The incomplete Cholesky decomposition with symmetric pivoting greedily selects the columns of A and calculates the columns of G such that the approximation of A obtained by adding that column in the current step is the best [31]. The algorithm terminates when $\|PAP^T - GG^T\|_1 = Tr(PAP^T - GG^T) \leq \epsilon_{tol}$ where P is the permutation matrix associated to the symmetric permutations done during the ICD and $\|\cdot\|_1$ is the trace norm [28, 31].

Let use the $\tilde{\Omega} = GG^T, G \in \mathbb{R}^{N_{tr}\times R}, R \ll N_{tr}$ incomplete Cholesky decomposition of the $\Omega \in \mathbb{R}^{N_{tr}\times N_{tr}}$ matrix as a rank R approximation in the form of

$$\Omega \approx \tilde{\Omega} = GG^T \tag{42}$$

This can be written as

$$\Omega \approx U\Lambda^2 U^T \tag{43}$$

by using the singular value decomposition of G i.e. $G = U\Lambda V^T$, where $\Lambda \in \mathbb{R}^{R\times R}$ is the diagonal matrix of singular values, $U \in \mathbb{R}^{N_{tr}\times R}$ and $V \in \mathbb{R}^{N_{tr}\times R}$ are the matrix of left and right singular vectors respectively. Using this approximation in the $D^{-1}M_D\Omega$ matrix, the kernel spectral clustering eigenvalue problem can be written as [17]

$$\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2 U^T\tilde{\boldsymbol{\beta}}^{(k)} = \tilde{\lambda}^{(k)}\tilde{\boldsymbol{\beta}}^{(k)}, \quad k = 1,\ldots,\mathcal{K}-1 \tag{44}$$

where the $\tilde{\cdot}$ denotes approximation of the corresponding quantity based on Eq. (42). Premultiplying both sides by U^T and substituting $\boldsymbol{\gamma}^{(k)} = U^T\tilde{\boldsymbol{\beta}}^{(k)} \in \mathbb{R}^R$ results in [17]

$$U^T\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2\boldsymbol{\gamma}^{(k)} = \tilde{\lambda}^{(k)}\boldsymbol{\gamma}^{(k)}, \quad k = 1,\ldots,\mathcal{K}-1 \tag{45}$$

The elements of the diagonal matrix $\tilde{D}$ are equal to $[\tilde{D}]_{ii} = \sum_{j=1}^{N_{tr}}[\tilde{\Omega}]_{ij}$ that can be calculated quickly as $G(G^T\mathbb{1}_{N_{tr}})$. Note that the size of this eigenvalue problem is $R\times R$ that can be much smaller than the $N_{tr}\times N_{tr}$ size of the original problem when the spectrum of $\Omega \in \mathbb{R}^{N_{tr}\times N_{tr}}$ decays quickly. The approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)}, k = 1,\ldots,\mathcal{K}-1$ of the original problem can be obtained from the corresponding $\boldsymbol{\gamma}^{(k)}, k = 1,\ldots,\mathcal{K}-1$ eigenvectors as $\tilde{\boldsymbol{\beta}}^{(k)} = U\boldsymbol{\gamma}^{(k)}, k = 1,\ldots,K-1$ [17].

The approximated bias terms $\tilde{b}^{(k)}, k = 1,\ldots,\mathcal{K}-1$ that appear in Eq. (41) can be calculated quickly by using Eq. (13) substituting $V = D^{-1} \approx \tilde{D}^{-1}$, $\Phi\Phi^T = \Omega \approx GG^T$ and $\boldsymbol{\beta}^{(k)} = \tilde{\boldsymbol{\beta}}^{(k)}, k = 1,\ldots,\mathcal{K}-1$.

4.2.2. Reduced set and reduced set coefficients

In order to obtain a sparse model representation, the next step is the determination of the $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}$ reduced set and the corresponding $\zeta^{(k)} \in \mathbb{R}^R, k = 1,\ldots,\mathcal{K}-1$ reduced set coefficients.

According to Eq. (40), these are used then to approximate the $\mathbf{w}^{(k)}, k = 1,\ldots,\mathcal{K}-1$ vectors as the linear combination of the mapped reduced set points $\{\varphi(\tilde{\mathbf{x}}_r)\}_{r=1}^R$ thus these $\varphi(\tilde{\mathbf{x}}_r)$ vectors should be linearly independent. The ICD algorithm with pivoting gradually selects the columns of Ω and builds $G \in \mathbb{R}^{N_{tr}\times R}$ such that the approximation $\Omega \approx \tilde{\Omega} = GG^T$ by adding that column in the current step is the best [31]. The column space of $\tilde{\Omega}$ is spanned by the selected columns of Ω thus the corresponding $\{\varphi(\tilde{\mathbf{x}}_r)\}_{r=1}^R$ mapped points are linearly independent. Therefore, the pivots selected by the ICD automatically provides the reduced set $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}$ [17]. The linear independence of the mapped reduced set points $\{\varphi(\tilde{\mathbf{x}}_r)\}_{r=1}^R$ also implies that $\Omega_{\Psi\Psi} = \Psi\Psi^T \in \mathbb{R}^{R\times R}$ is full rank where $\Psi = [\varphi(\tilde{\mathbf{x}}_1),\ldots,\varphi(\tilde{\mathbf{x}}_R)]^T \in \mathbb{R}^{R\times n_h}$.

The corresponding reduced set coefficients $\zeta^{(k)} \in \mathbb{R}^R, k = 1,\ldots,\mathcal{K}-1$ can be determined by minimizing the squared distance between the corresponding $\mathbf{w}^{(k)} = \Phi^T\tilde{\boldsymbol{\beta}}^{(k)}, k = 1,\ldots,\mathcal{K}-1$ and $\tilde{\mathbf{w}}^{(k)} = \Psi^T\zeta^{(k)}, k = 1,\ldots,\mathcal{K}-1$ vectors where $\Phi = [\varphi(\mathbf{x}_1),\ldots,\varphi(\mathbf{x}_{N_{tr}})]^T \in \mathbb{R}^{N_{tr}\times n_h}$ [16, 17, 18]. This can be written as

$$\begin{aligned}
&\min_{\zeta^{(k)}}\|\mathbf{w}^{(k)} - \tilde{\mathbf{w}}^{(k)}\|_2^2 = \\
&\quad = \min_{\zeta^{(k)}}\{\mathbf{w}^{(k)^T}\mathbf{w}^{(k)} - 2\tilde{\mathbf{w}}^{(k)^{(T)}}\mathbf{w}^{(k)} + \tilde{\mathbf{w}}^{(k)^{(T)}}\tilde{\mathbf{w}}^{(k)}\} \\
&\quad = \min_{\zeta^{(k)}}\{\tilde{\boldsymbol{\beta}}^{(k)^T}\Omega\tilde{\boldsymbol{\beta}}^{(k)} - 2\zeta^{(k)^T}\Omega_{\Psi\Phi}\tilde{\boldsymbol{\beta}}^{(k)} + \zeta^{(k)^T}\Omega_{\Psi\Psi}\zeta^{(k)}\}
\end{aligned} \tag{46}$$

$k = 1, \ldots, \mathcal{K}-1$, where $\Omega_{\Phi\Phi} = \Phi\Phi^T \in \mathbb{R}^{N_{tr} \times N_{tr}}$, $\Omega_{\Psi\Phi} = \Psi\Phi^T \in \mathbb{R}^{R \times N_{tr}}$ and $\Omega_{\Psi\Psi} = \Psi\Psi^T \in \mathbb{R}^{R \times R}$ are the corresponding kernel matrices. Taking the partial derivative of this objective with respect to $\zeta^{(k)}$ and setting to zero, the first order optimality $\partial \|\mathbf{w}^{(k)} - \tilde{\mathbf{w}}^{(k)}\|_2^2 / \partial \zeta^{(k)}$ results in the following linear system

$$\Omega_{\Psi\Psi} \zeta^{(k)} = \Omega_{\Psi\Phi} \tilde{\boldsymbol{\beta}}^{(k)} \tag{47}$$

for the $\zeta^{(k)}, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients [16, 17, 18].

4.2.3. Clustering in the sparse model

Having both the reduced set and the corresponding reduced set coefficients determined, one can calculate the score variables for arbitrary data points by means of Eq. (41). The clustering of these points can be done based on the obtained score variables and the codebook as discussed in section 3.3 when the codebook is formed based on the sign encoding-decoding scheme. However, the angular similarity based encoding-decoding scheme is slightly different in the case of the sparse model.

The angular similarity based encoding-decoding scheme was introduced in section 3.3 based on the facts, that the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix are piecewise constants on the underlying clusters in the ideal case and the $\mathbf{z}^* - \mathbf{b} \in \mathbb{R}^{1 \times (\mathcal{K}-1)}$ points are the linear combination of the $\mathcal{K}$, $\mathcal{K}-1$ dimensional vectors formed from these constants (see Eq. (34)). According to Eq. (41), the approximated $\tilde{\mathbf{z}}_i^* - \tilde{\mathbf{b}}$ point, associated to an arbitrary input data point $\mathbf{x}_i \in \mathbb{R}^d$, is expressed as the linear combination of the $\tau_r^* = [\zeta_r^{(1)}, \ldots, \zeta_r^{(\mathcal{K}-1)}], r = 1, \ldots, R$ reduced set coefficient points

$$\tilde{\mathbf{z}}_i^* - \tilde{\mathbf{b}} = \sum_{r=1}^{R} K(\mathbf{x}_i, \tilde{\mathbf{x}}_r) \tau_r^* \tag{48}$$

when the reduced set method based sparse model is used. Although the τ_r^* vectors associated to reduced set points $\tilde{\mathbf{x}}_r$ from the same cluster $r \in C_k, k \in \{1, \ldots, \mathcal{K}\}$ are not piecewise constant, one can show that in the ideal case they point into the same direction i.e. they are collinear. In order to do so, let assume that the rows and columns of $\Omega_{\Psi\Psi}$,$\Omega_{\Psi\Phi}$ and the rows of $\tilde{B} = [\tilde{\boldsymbol{\beta}}^{(1)}, \ldots, \tilde{\boldsymbol{\beta}}^{(\mathcal{K}-1)}] \in \mathbb{R}^{N_{tr} \times \mathcal{K}-1}$ are ordered according to the perfect partition of the training data set. $\Omega_{\Psi\Psi}$ is a block diagonal and $\Omega_{\Psi\Phi}$ is a block matrix in the ideal case. Furthermore, $\Omega_{\Psi\Psi}$ has full rank thus invertible and its inverse $\Omega_{\Psi\Psi}^{-1}$ matrix is also block diagonal partitioned identically to $\Omega_{\Psi\Psi}$. Therefore, the $\Theta = \Omega_{\Psi\Psi}^{-1}\Omega_{\Psi\Phi} \in \mathbb{R}^{R \times N_{tr}}$ matrix is also a block matrix on the clusters which means that $[\Theta]_{rj} \neq 0 \iff r, j \in C_k, k \in \{1, \ldots, \mathcal{K}\}$. Then the $X = [\zeta^{(1)}, \ldots, \zeta^{(\mathcal{K}-1)}] \in \mathbb{R}^{R \times (\mathcal{K}-1)}$ reduced set coefficient matrix can be written as $X = \Theta\tilde{B}$ by using Eq. (47). Let assume that the r-th reduced set point $\tilde{\mathbf{x}}_r \in \mathcal{R}$ belongs to the k-th cluster. The rp-th element of X, that corresponds to the p-th components of the τ_r^* vector,

can be written as

$$
\begin{aligned}
[X]_{rp} &= \sum_{j=1}^{N_{tr}} [\Theta]_{rj} [\tilde{B}]_{jp} = \sum_{j=1}^{N_{tr}} [\Theta]_{rj} \tilde{\beta}_j^{(p)} \\
&= \sum_{j \in C_k} [\Theta]_{rj} \tilde{\beta}_j^{(p)} + \sum_{j \notin C_k} [\Theta]_{rj} \tilde{\beta}_j^{(p)} \\
&= \xi_k^{(p)} \sum_{j \in C_k} [\Theta]_{rj} + \sum_{k' \neq k} \xi_{k'}^{(p)} \sum_{j \in C_{k'}} [\Theta]_{rj}
\end{aligned}
\tag{49}
$$

$\forall r \in C_k, k \in \{1, \ldots, \mathcal{K}\}$ where the fact that in the ideal case the $\tilde{\boldsymbol{\beta}}^{(p)}, p \in \{1, \ldots, \mathcal{K}-1\}$ approximated eigenvectors are piecewise constants on the partition $\tilde{\beta}_i^{(p)} = \tilde{\beta}_j^{(p)} = \xi_k^{(p)}, \forall i, j \in C_k, k \in \{1, \ldots, \mathcal{K}\}$ was used to obtain the last equation. Since Θ is a block matrix on the partition in the ideal case, the last term is zero. The r-th row of X, that corresponds to τ_r^* can then be written as

$$
\tau_r^* = [\xi_k^{(1)}, \ldots, \xi_k^{(K-1)}] \sum_{j \in C_k} [\Theta]_{rj} \tag{50}
$$

$\forall r \in C_k, k \in \{1, \ldots, \mathcal{K}\}$. Therefore, the τ_r^* vectors, that correspond to reduced set points $\tilde{\mathbf{x}}_r \in \mathcal{R}$ from the same cluster, point to the same direction in the ideal case. Since the cluster membership is coded in the signs of the components of the $\tilde{\beta}_i^{(k)}, k = 1, \ldots, \mathcal{K}-1$ eigenvectors, these $\mathcal{K}$ directions lie in different orthants of the $\mathcal{K}-1$ dimensional reduced set coefficient space. Thus the $\mathcal{K}$ clusters are represented by these $\mathcal{K}$ directions in a well separated way. In perturbed cases, the $\Omega_{\Psi\Psi}, \Omega_{\Psi\Phi}$ matrices have nearly block structures and the $\tilde{\beta}_i^{(k)}, k = 1, \ldots, \mathcal{K}-1$ eigenvectors are approximately piecewise constants on the clusters similarly to the non-sparse model. The τ_r^* vectors, that correspond to reduced set points $\tilde{\mathbf{x}}_r \in \mathcal{R}$ from the same cluster, point approximately to the same direction in these cases.

Therefore, the angular similarity based encoding-decoding scheme is very similar to that introduced in case of the non-sparse model in section 3.3 with the difference, that now the $\tau_r^*, r = 1, \ldots, R$ vectors will play the role of the $\mathbf{e}_j^*, j = 1, \ldots, N$ vectors. The codebook can be formed by clustering the $\tau_r^*, r = 1, \ldots, R$ vectors into $\mathcal{K}$ group based on sign coding or their angular similarities and the $\mathcal{K}$ prototype vectors will form the codebook after normalization. An arbitrary point can then be clustered based on the corresponding approximated score vector calculated by using Eq. (41) and the codebook as described in section 3.3.

4.2.4. Proposed algorithm

The computational bottleneck of the ICD based sparse kernel spectral clustering algorithm, proposed in [17], is the computation of the approximated $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix associated to the $\mathcal{D}^{tr} = \{\mathbf{x}_i\}_{i=1}^{N_{tr}}$ training set discussed in section 4.2.1. In case of large scale problems, one aims

to select as large N_{tr} training set as possible in order to involve as much information on the underlying data structure, contained in the full data set, as possible into the training. Then the information contained by selected eigenvectors associated to the training set is compressed into the reduced set coefficients and the corresponding $R \ll N_{tr}$ size reduced set when the reduced set method is used. According to section 4.2.1, calculation of these selected approximated eigenvectors involves the formation of the $U^T\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2$ matrix in order to reduce the size of the eigenproblem from N_{tr} to R. The naive implementation of this step can lead to a computational complexity of $\mathcal{O}(R^2N_{tr}^2)$ that makes the original algorithm unsuitable for large-scale problems . However, this computational complexity can be reduced to $\mathcal{O}(R^2N_{tr})$ by exploiting the diagonal structure of $\tilde{D}^{-1}, \Lambda$ matrices and using the fact, that the effect of left multiplying $U \in \mathbb{R}^{N_{tr}\times R}$ by $M_{\tilde{D}}$ is that it removes the $1/\tilde{d}_i, i = 1, , N_{tr}$ weighted mean from each column of U. The i-th element of the j-th column of the $\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2 \in \mathbb{R}^{N_{tr}\times R}$ matrix can then be calculated as

$$[\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2]_{ij} = \frac{\lambda_j^2}{\tilde{d}_i}([U]_{ij} - \chi_j) \tag{51}$$

where

$$\chi_j = \frac{1}{\epsilon}\sum_{k=1}^{N_{tr}} \frac{[U]_k j}{\tilde{d}_k} \tag{52}$$

is the $1/\tilde{d}$ weighted mean of the j-th column of U and $\epsilon = \sum_{l=1}^{N_{tr}} 1/\tilde{d}_l$. Therefore, precalculating and storing χ_j for the j-th column, the $\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2$ matrix can be generated column-by-column and left multiplying by U^T yields the $U^T\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2 \in \mathbb{R}^{N_{tr}\times R}$ matrix in $\mathcal{O}(R^2N_{tr})$. The training algorithm has a $\mathcal{O}(R^2N_{tr})$ computational complexity by using the implementation described above. However, this implementation of the training algorithm is still based on the $U^T\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2$ matrix thus it involves the SVD of the $G \in \mathbb{R}^{N_{tr}\times R}$ matrix. However, this SVD step, as well as the overall computation of the $U^T\tilde{D}^{-1}M_{\tilde{D}}U\Lambda^2$ matrix, can be avoided and the training algorithm can be accelerated further by calculating the approximated $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix, associated to the training data set, as follows.

The weighted kernel PCA formulated kernel spectral clustering eigenvalue problem, given by Eq. (20), can be transformed to

$$D^{-\frac{1}{2}}M_D\Omega M_D^T D^{-\frac{1}{2}}\boldsymbol{\alpha}^{(k)} = \lambda^{(k)}\boldsymbol{\alpha}^{(k)}, \quad k = 1, \ldots, \mathcal{K}-1 \tag{53}$$

as given by Eq. (22) where $\boldsymbol{\alpha}^{(k)} = D^{1/2}\boldsymbol{\beta}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ and the corresponding $\lambda^{(k)}$ eigenvalues are the same as in the original eigenproblem. One can write

$$D^{-\frac{1}{2}}M_D\Omega M_D^T D^{-\frac{1}{2}} \approx \tilde{D}^{-\frac{1}{2}}M_{\tilde{D}}GG^T M_{\tilde{D}}^T\tilde{D}^{-\frac{1}{2}} \tag{54}$$

for the $\mathcal{D}^{tr} = \{\mathbf{x}_i\}_{i=1}^{N_{tr}}$ training data set by using the approximation given in Eq. (42). The QR factorization of $X = \tilde{D}^{-1/2}M_{\tilde{D}}G \in \mathbb{R}^{N_{tr}\times R}$ is $X = Q_X R_X$

where $Q_X \in \mathbb{R}^{N_{tr} \times R}$ and $R_X \in \mathbb{R}^{R \times R}$. Eq. (54) can then be written as

$$\begin{aligned} \tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} G G^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}} = \\ = [Q_X U_{R_X} \Sigma_{R_X} V_{R_X}^T][Q_X U_{R_X} \Sigma_{R_X} V_{R_X}^T]^T \\ = [Q_X U_{R_X}] \Sigma_{R_X}^2 [Q_X U_{R_X}]^T \end{aligned} \tag{55}$$

by substituting R_X with its singular value decomposition $R_X = U_{R_X} \Sigma_{R_X} V_{R_X}^T$ where $U_{R_X}, V_{R_X} \in \mathbb{R}^{R \times R}$ are the left and right singular vectors of R_X and $\Sigma_{R_X} \in \mathbb{R}^{R \times R}$ diagonal matrix contains the corresponding singular values. Note, that Eq. (55) is the eigenvalue decomposition of the $\tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} G G^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}} \in \mathbb{R}^{N_{tr} \times N_{tr}}$ matrix with $Q_X U_{R_X} \in \mathbb{R}^{N_{tr} \times R}$ eigenvectors and $\Sigma_{R_X}^2 = \Lambda_{R_X} \in \mathbb{R}^{R \times R}$ eigenvalues. Therefore, the $\mathcal{K}-1$ leading eigenvectors $\tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ of the $\tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} G G^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}}$ matrix can be obtained by taking the $\mathcal{K}-1$ columns of the $Q_X U_{R_X}$ matrix that corresponds to the $\mathcal{K}-1$ largest eigenvalues in $\Sigma_{R_X}^2 = \Lambda_{R_X}$. Then the ICD based approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ of the $D^{-1} M_D \Omega$ matrix, associated to the training data set, can be calculated from the corresponding $\tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ eigenvectors as $\tilde{\boldsymbol{\beta}}^{(k)} = \tilde{D}^{-1/2} \tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$. These modifications, for calculating the approximated $\mathcal{K}-1$ leading eigenvectors of the $D^{-1} M_D \Omega$ matrix associated to the training set, were motivated by the approach in [32] where the calculation of the leading $\mathcal{K}$ eigenvectors of the symmetric normalized graph Laplacian L_{sym} were accelerated in a similar way.

This modified way of calculating the approximated $\mathcal{K}-1$ leading eigenvectors of the $D^{-1} M_D \Omega$ matrix, associated to the training set, has several advantages over the original algorithm described in section 4.2.1. While the original algorithm includes an SVD of $G \in \mathbb{R}^{N_{tr} \times R}$, the proposed algorithm performs only a QR factorization on $X = \tilde{D}^{-1/2} M_{\tilde{D}} G \in \mathbb{R}^{N_{tr} \times R}$ and an SVD on $R_X \in \mathbb{R}^{R \times R}$ which can be done faster. Furthermore, since $\tilde{D}^{-1/2} \in \mathbb{R}^{N_{tr} \times N_{tr}}$ is diagonal and the effect of left multiplying G by $M_{\tilde{D}}$ is that it removes the $1/\tilde{d}_i$ weighted mean from each column of G, X can be generated quickly column-by-column in place of G as

$$[X]_{ij} = \frac{1}{\sqrt{\tilde{d}_i}} ([G]_{ij} - \chi_j) \tag{56}$$

where χ_j is the precomputed $1/\tilde{d}_i$ weighted mean of the j-th column of G. Since QR factorization of X can also be done in place, the proposed algorithm does not need extra memory.

The proposed algorithm for training the sparse kernel spectral clustering model is described in Algorithm 1 by using the angular similarity based encoding-decoding scheme and an arbitrary point can be clustered by using the trained model as described in Algorithm 2. The approximated bias terms $\tilde{b}^{(k)}, k = 1, \ldots, \mathcal{K}-1$, that appear in Eq. (41), can now be calculated quickly by using Eq. (26) substituting $N = N_{tr}$, $D \approx \tilde{D}$, $\boldsymbol{\beta}^{(k)} = \tilde{\boldsymbol{\beta}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ and the corresponding approximated leading eigenvalues $\tilde{\lambda}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ from $\Sigma_{R_X}^2 = \Lambda_{R_X} \in \mathbb{R}^{R \times R}$.

Algorithm 1 and Algorithm 2 come here...

5. Computational experiments

Some results of computational experiments will be presented in this section to demonstrate the improved computational power and performance of the presented sparse clustering algorithm over the original version. The proposed sparse spectral clustering algorithm was implemented in C programming language based on LAPACK [33] and BLAS [34, 35] libraries. The full library used in this work, together with separate applications for performing the different steps of the clustering problem such as incomplete Cholesky factorization, kernel parameter tuning, training and out-of-sample extension, is available at [36]. The presented results were obtained by using Fedora 12 operating system running on an Intel Core 2 Duo, 2.26 GHz, 3.8 GB RAM hardware with an automatically optimized BLAS library using the freely available ATLAS [37, 38]. Only a single core was used for the computations. The adjusted rand index (ARI) as an external clustering quality measure was used for the assessment of the clustering result in some cases [39].

5.1. Intertwined spiral

The first problem was taken from [17] and it consists of $N = 10^5$ two dimensional data points sampled from two intertwined spiral as shown in Fig. 1. According to section 3.2.3, this binary clustering problem is considered to be hard since the spectrum of the associated normalized graph Laplacian decays slowly and the eigengap is very small. The original sparse algorithm, discussed in section 4.2.1 and 4.2.2, was used to solve this problem as a function of the training set size in [17]. This toy problem was chosen to prove the improved performance of the presented sparse clustering algorithm compared to the original version in terms of computational time and sparsity.

The average computation times of the original and the proposed algorithm, based on 10 independent runs, are compared in Fig. 2 at different N_{tr} training set sizes sampled randomly from the full $N = 10^5$ input data. The size of the reduced set R was fixed in the incomplete Cholesky decomposition to the same value as given in [17] in order to obtain comparable computational times. An RBF kernel was used in the form of $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2)$. The optimal kernel parameter $\sigma^2 = 0.006$ was determined by maximizing the BLF model selection criterion and the same value was used during the training and the test as well as in the ICD. The algorithm was trained on the N_{tr} size training set with a reduced set size of R and the whole $N = 10^5$ input data set was clustered during the test (out-of-sample extension) by using the trained model. The results are shown in Fig. 2 and the corresponding numerical data are in Table 1 for the proposed algorithm. Note, that the original algorithm needs about eight hours to solve the problem when the whole $N_{tr} = N = 10^5$ input data set is used for training while the proposed algorithm can solve the same problem within less than 10s with exactly the same settings. The original algorithm

spends most of the time on the calculation of the approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ that has a $\mathcal{O}(R^2 N_{tr}^2)$ computational complexity in [17]. This computational complexity is reduced to $\mathcal{O}(R^2 N_{tr})$ by the proposed algorithm that can significantly accelerate the computation of these approximated eigenvectors especially in case of large training sets.

Furthermore, the proposed algorithm can solve the problem perfectly based on smaller reduced set sizes as it is shown in Table 2 which indicates a higher level of sparsity of the clustering model. The better sparsity of the presented model is originated from the different computation of the bias terms compared to the original version. In the original algorithm the bias terms are re-estimated based on the reduced set, reduced set coefficient and the degree matrix associated to the reduced set. In the proposed algorithm the bias terms are computed by using the $k = 1, \ldots, \mathcal{K}-1$ approximated leading eigenvalue-eigenvector pairs and the degree matrix associated to the approximated training set kernel matrix as described at the end of section 4.2.4. The latter results in a better approximation since this computation is based on the information content of the approximated training set kernel matrix instead of the reduced set kernel matrix. As a consequence, a smaller reduced set size is enough to obtain a better approximation of the bias terms. The minimal reduced set size, that is still enough to get an average $ARI = 1$ calculated on the whole data set based on 10 independent runs, is $R = 110$ when the full $N = 10^5$ input data is used for the training. The obtained reduced set points as well as the clustering result are shown in Fig. 1. Note, that according to Table 2, a perfect clustering model can be trained within 0.14s by applying the proposed training algorithm on a training and on a reduced set with corresponding sizes of $N_{tr} = 5000$ and $R = 122$. Then the whole 10^5 data can be clustered perfectly within 1.11s by using the trained clustering model. Thus the problem can be solved within 1.25s by using the proposed algorithm.

Fig. 1, Fig. 2, Table 1, Table 2 come here...

5.2. Image segmentation

Ten color images were selected from the Berkeley image data set [5] to demonstrate the applicability and performance of the presented clustering algorithm on real life problems such as image segmentation. Each of the RGB color images consists of $N = 321 \times 481 = 154401$ pixels. A local color histogram was computed at each pixel by taking a 5×5 window around the pixels using minimum variance color quantization of eight levels. After normalization, the $N = 154401$ histograms serve the 8 dimensional input data points of the clustering problem. The χ^2 kernel $K(h^{(i)}, h^{(j)}) = \exp(-\chi^2_{ij}/\sigma_{\chi^2})$ was used to compute the similarity between two local color histograms $h^{(i)}$ and $h^{(j)}$ with σ_{χ^2} kernel parameter and $\chi^2_{ij} = 0.5 \sum_{l=1}^{8} (h_l^{(i)} - h_l^{(j)})^2 / (h_l^{(i)} + h_l^{(j)})$ [40, 8, 11].

[5] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/`

$N_{tr} = 10000$ and $N_v = 20000$ data points were selected randomly for training and validation respectively. The optimal value of the kernel parameter σ_{χ^2} and the number of clusters $\mathcal{K}$ were determined by maximizing the model selection criterion given by Eq. (38) with $\eta = 1$ on the validation set over a $(\mathcal{K} \in \{3, \ldots, 10\}, \sigma_{\chi^2} \in [0.001, 1.0])$ grid. In those cases, when the maximum value of the model selection criterion was obtained at $\mathcal{K} = 3$ the BLF model selection criterion was also computed for $\mathcal{K} = 2, 3$ and the one that gave the highest value was accepted. A typical example is shown in Fig. 3 for image ID=145086. First the ICD algorithm was applied on the training set with a size of $N_{tr} = 10000$, kernel parameter value $\sigma_{\chi^2} = 0.05$ and $\epsilon_{tol} = 0.6$. The original image and the $R = 119$ reduced set points are shown on the top of Fig. 3. Then the results of the ICD i.e. the incomplete Cholesky factor $G \in \mathbb{R}^{N_{tr} \times R}$ and the $R = 119$ reduced set points were used together with the training set to train the sparse clustering model at each $(\mathcal{K}, \sigma_{\chi^2})$ grid point by using Algorithm 1. Following the training, the obtained clustering model was validated at each $(\mathcal{K}, \sigma_{\chi})$ grid point by using Algorithm 2 and the validation set. The (interpolated) result of this tuning procedure is shown on the bottom of Fig. 3. The optimal $(\mathcal{K} = 4, \sigma_{\chi^2} = 0.021)$ parameter value pair was chosen that gave the maximum value 0.942 of the model selection criterion. The whole procedure described above was repeated several times by using different kernel parameter σ_{χ^2} values in the ICD phase and the one $\sigma_{\chi^2} = 0.05$ that resulted in the highest model selection criterion in the tuning phase was accepted.

Following the parameter tuning procedure, the clustering model was retrained on the training set by using the obtained optimal model parameters and the whole color histogram set was clustered. A Canny edge detector was applied on the obtained segment-label data in order to compute the edges. A segment-label image and the corresponding edges are shown in Fig. 4 as an example. The F-measure, with respect to human segmentation, was computed as a performance criterion. The obtained F-measures are compared with earlier results [11] obtained by using the original non-sparse kernel spectral clustering algorithm discussed in section 3 and by means of Nyström method in Table 3. The current training and test times as well as the reduced set sizes are also shown in the last three columns.

The proposed algorithm outperforms the earlier methods in terms of agreement with human segmentation. The modifications presented in Section 4.2.4 makes the sparse KSC algorithm applicable to bigger training sets that can significantly increase the accuracy of the final clustering result. On the other hand, the linear dependence of the computational complexity on the training set size ensures that this performance increase can be achieved without paying significant extra computational costs. The computation time of the original non-sparse KSC model and the Nyström method based algorithm was investigated in [11] as a function of the training set size. Based on these results, the non-sparse KSC model is faster than the Nyström approximation based algorithm by about two order of magnitudes when the training set size is 10^4. The non-sparse KSC model needs about 900 s to compute the necessary eigenvectors, train the model on the 10^4 data pints and cluster the whole color histogram set by using

the trained model and its out-of-sample capability. The presented version of the sparse algorithm needs only 0.33 s to obtain the approximate eigenvectors, train the sparse clustering model on the 10^4 training data and the whole color histogram set can be clustered within additional 2.31 s.

Fig. 3, Fig. 4, Table 3 come here...

6. Conclusion

A fast spectral clustering algorithm, suitable for large scale data clustering problems, is presented. The algorithm is an improved version of the sparse multiway spectral clustering algorithm, formulated as weighted kernel principal component (PCA) analysis within primal-dual least-squares support vector machine (LS-SVM) framework. Sparsity is achieved by the combination of reduced set method with the incomplete Cholesky factorization based low rank approximation of the kernel matrix. The quadratic computational complexity in the training set size prevented the application of the original algorithm to large scale data analysis problems. This quadratic computational complexity is reduced to linear in the proposed algorithm while all the attractive properties such as the simple way of out-of-sample extension, possibility of model selection through training and validation are conserved. A new, soft cluster membership encoding-decoding scheme and a corresponding new model selection criterion is also proposed both for the original non-sparse KSC algorithm and for the presented sparse version. These modifications make the algorithm capable to solve much larger size of clustering problems than the original one. The significantly reduced computational complexity, the improved accuracy and sparsity of the presented algorithm compared to the original version are illustrated by experimental results obtained on known difficult toy as well as real life problems such as image segmentation.

Acknowledgement

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants IWT: projects: SBO POM (100031); PhD/Postdoc grants iMinds Medical Information Technologies SBO 2014 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017)

Appendices

A. Explicit weighted centering of the feature map

Eq. (5) can be written, by using explicit centering of the feature map, as

$$\max_{\mathbf{w}^{(k)}} \sum_{k=1}^{\mathcal{K}} \sum_{l=1}^{N} v_l [0 - \mathbf{w}^{(k)^T} (\varphi(\mathbf{x}_l) - \hat{\mu}^v_{\varphi(\mathbf{x})})]^2 \tag{57}$$

where

$$\hat{\mu}^v_{\varphi(\mathbf{x})} = \frac{\sum_{k=1}^{N} v_k \varphi(\mathbf{x}_k)}{\sum_{k=1}^{N} v_k} = \frac{\Phi^T V \mathbb{1}_N}{\mathbb{1}_N^T V \mathbb{1}_N} \tag{58}$$

is the weighted mean of the feature map. This can be formulated as

$$\max_{\mathbf{w}^{(k)}, \mathbf{e}^{(k)}} \sum_{k=1}^{\mathcal{K}} \mathbf{e}^{(k)^T} V \mathbf{e}^{(k)} \tag{59}$$

where $\mathbf{e}^{(k)} \in \mathbb{R}^{N \times 1}, k = 1, \ldots, \mathcal{K}$ error vectors now have the following form

$$\begin{aligned} \mathbf{e}^{(k)} &= (\Phi - \mathbb{1}_N \hat{\mu}^{v^T}_{\varphi(\mathbf{x})}) \mathbf{w}^{(k)} \\ \mathbf{e}^{(k)} &= \begin{bmatrix} e_1^{(k)} \\ \vdots \\ e_N^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{w}^{(k)^T} (\varphi(\mathbf{x}_1) - \hat{\mu}^v_{\varphi(\mathbf{x})}) \\ \vdots \\ \mathbf{w}^{(k)^T} (\varphi(\mathbf{x}_N) - \hat{\mu}^v_{\varphi(\mathbf{x})}) \end{bmatrix} \end{aligned} \tag{60}$$

The primal optimization problem becomes

$$\begin{aligned} \max_{\mathbf{w}^{(k)}, \mathbf{e}^{(k)}} J(\mathbf{w}, \mathbf{e}) &= \frac{1}{2} \sum_{k=1}^{\mathcal{K}} \gamma^{(k)} \mathbf{e}^{(k)^T} V \mathbf{e}^{(k)} \\ &\quad - \frac{1}{2} \sum_{k=1}^{K} \mathbf{w}^{(k)^T} \mathbf{w}^{(k)} \\ \textit{such that} \quad \mathbf{e}^{(k)} &= (\Phi - \mathbb{1}_N \hat{\mu}^{v^T}_{\varphi(\mathbf{x})}) \mathbf{w}^{(k)}, k = 1, \ldots, \mathcal{K} \end{aligned} \tag{61}$$

The corresponding Lagrangian is

$$\begin{aligned} &\mathcal{L}(\mathbf{w}^{(k)}, \mathbf{e}^{(k)}, b^{(k)}; \boldsymbol{\beta}^{(k)}) = \frac{1}{2} \sum_{k=1}^{\mathcal{K}} \gamma^{(k)} \mathbf{e}^{(k)^T} V \mathbf{e}^{(k)} - \\ &\frac{1}{2} \sum_{k=1}^{\mathcal{K}} \mathbf{w}^{(k)^T} \mathbf{w}^{(k)} - \sum_{k=1}^{\mathcal{K}} \boldsymbol{\beta}^{(k)^T} (\mathbf{e}^{(k)} - \Phi \mathbf{w}^{(k)} + \mathbb{1}_N \hat{\mu}^{v^T}_{\varphi(\mathbf{x})} \mathbf{w}^{(k)}) \end{aligned} \tag{62}$$

with $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{N\times 1}, k = 1, \ldots, \mathcal{K}$ Lagrange multiplier vectors. The Karush-Kuhn-Tucker (KKT) optimality conditions are

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mathbf{w}^{(k)}} = 0 \quad \rightarrow \quad \mathbf{w}^{(k)} = \Phi^T \boldsymbol{\beta}^{(k)} - \hat{\mu}^v_{\varphi(\mathbf{x})} \mathbb{1}_N^T \boldsymbol{\beta}^{(k)} \\ \dfrac{\partial \mathcal{L}}{\partial \mathbf{e}^{(k)}} = 0 \quad \rightarrow \quad \boldsymbol{\beta}^{(k)} = \gamma^{(k)} V \mathbf{e}^{(k)} \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}^{(k)}} = 0 \quad \rightarrow \quad \mathbf{e}^{(k)} - \Phi \mathbf{w}^{(k)} + \mathbb{1}_N \hat{\mu}^{v^T}_{\varphi(\mathbf{x})} \mathbf{w}^{(k)} = 0 \end{cases} \tag{63}$$

for $k = 1, \ldots, \mathcal{K}$. Note, that the centered property of the Lagrange multiplyer vectors $\mathbb{1}_N^T \boldsymbol{\beta}^{(k)} = 0$ is explicitly not among these KKT conditions now. However, this property can be seen by substituting the third equation into the second and using the expression for $\hat{\mu}^v_{\varphi(\mathbf{x})}$ given by Eq. (58)

$$\begin{aligned} \mathbb{1}_N^T \boldsymbol{\beta}^{(k)} &= \gamma^{(k)} \mathbb{1}_N^T V \mathbf{e}^{(k)} = \gamma^{(k)} \mathbb{1}_N^T V [\Phi - \mathbb{1}_N \hat{\mu}^{v^T}_{\varphi(\mathbf{x})}] \mathbf{w}^{(k)} \\ &= \gamma^{(k)} [\mathbb{1}_N^T V \Phi - \frac{\mathbb{1}_N^T V \mathbb{1}_N}{\mathbb{1}_N^T V \mathbb{1}_N} \mathbb{1}_N^T V \Phi] \mathbf{w}^{(k)} = 0 \end{aligned} \tag{64}$$

for $k = 1, \ldots, \mathcal{K}$. Thus, the first equation in Eqs. (63) becomes $\mathbf{w}^{(k)} = \Phi^T \boldsymbol{\beta}^{(k)}$ for all $k = 1, \ldots, \mathcal{K}$. Eliminating the primal variables $\mathbf{w}^{(k)}$ and $\mathbf{e}^{(k)}$ by substituting the second ($\mathbf{e}^{(k)} = \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}}$) and these simplified first equations into the third results in

$$\begin{aligned} \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} - \Phi\Phi^T \boldsymbol{\beta}^{(k)} + \frac{\mathbb{1}_N \mathbb{1}_N^T V \Phi \Phi^T \boldsymbol{\beta}^{(k)}}{\mathbb{1}_N^T V \mathbb{1}_N} = 0 \\ \frac{V^{-1}\boldsymbol{\beta}^{(k)}}{\gamma^{(k)}} - \left[I_N - \frac{\mathbb{1}_N \mathbb{1}_N^T V}{\mathbb{1}_N^T V \mathbb{1}_N} \right] \Phi\Phi^T \boldsymbol{\beta}^{(k)} = 0 \end{aligned} \tag{65}$$

$k = 1, \ldots, \mathcal{K}$ where the expression for $\hat{\mu}^v_{\varphi(\mathbf{x})}$ given by Eq. (58) was also used. Since these equations are identical to Eqs. (14), the same eigenvalue problem (given by Eq. (15)) can be derived with explicit weighted centering of the feature map as by using the bias term approach and this latter results in an automatic, implicit, weighted centering of the kernel matrix. Also note, that eliminating the primal variables from the KKT equations (Eqs. (63)) without using the $\mathbb{1}_N^T \boldsymbol{\beta}^{(k)} = 0$ property would yield the $V M_v \Omega M_v^T \boldsymbol{\beta}^{(k)} = \lambda^{(k)} \boldsymbol{\beta}^{(k)}, k = 1, \ldots, N$ eigenvalue problem with the same $\lambda^{(k)}, \boldsymbol{\beta}^{(k)}$ eigenvalue-eigenvector pairs. Thus, the identity given by Eq. (66) can also be shown by using the explicit centering approach.

B. Mutual V^{-1}-orthogonality of the $\boldsymbol{\beta}^{(k)}$ eigenvectors of the $VM_v\Omega$ matrix

According to the third KKT condition in Eqs. (11), the $\boldsymbol{\beta}$ eigenvectors of the $VM_v\Omega$ matrix are centered i.e. $\mathbb{1}_N^T\boldsymbol{\beta} = 0$. Therefore, these vectors are also eigenvectors of the $VM_v\Omega M_v^T$ matrix with the same eigenvalues

$$VM_v\Omega M_v^T\boldsymbol{\beta} = VM_v\Omega\{\boldsymbol{\beta} - \frac{V\mathbb{1}_N\overbrace{\mathbb{1}_N^T\boldsymbol{\beta}}^{0}}{\mathbb{1}_N^TV\mathbb{1}_N}\} = VM_v\Omega\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \tag{66}$$

The eigenproblem can be written as $M_v\Omega M_v^T\boldsymbol{\beta} = \lambda V^{-1}\boldsymbol{\beta}$ where V^{-1} is a real, positive definite matrix and $M_v\Omega M_v^T$ is a real, symmetric and positive semi definite. Hence, this problem corresponds to the usual real symmetric eigenproblem [41], which means that there are N real non-negative eigenvalues of this problem and the corresponding $\boldsymbol{\beta}$ eigenvectors form an orthogonal set with respect to the V^{-1}-*inner product* defined as

$$(\mathbf{x}, \mathbf{y})_{V^{-1}} = (V^{-1}\mathbf{x}, \mathbf{y}) \tag{67}$$

and the V^{-1}-*norm* is given by

$$\|\mathbf{x}\|_{V^{-1}} = (V^{-1}\mathbf{x}, \mathbf{x})^{-1/2} \tag{68}$$

Note, that although the matrix $VM_v\Omega M_v^T$ is non-symmetric with respect to the Euclidean inner product, it is symmetric with respect to the V^{-1}-*inner product*

$$(VM_v\Omega M_v^T\mathbf{x}, \mathbf{y})_{V^{-1}} = (\mathbf{x}, VM_v\Omega M_v^T\mathbf{y})_{V^{-1}},\ \forall\, \mathbf{x}, \mathbf{y} \tag{69}$$

Thus the $\boldsymbol{\beta}^{(k)}, k = 1\ldots,\mathcal{K}$ eigenvectors of the $VM_v\Omega$ matrix are mutually V^{-1}-orthogonal. It is further assumed in Eq. (17) that these eigenvectors are normalized according to the V^{-1}-norm.

References

[1] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[2] F. R. K. Chung, Spectral graph theory, Vol. 92, American Mathematical Soc., 1997.

[3] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering analysis and an algorithm, Proceedings of Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press 14 (2001) 849–856.

[4] L. Hagen, A. B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 11 (9) (1992) 1074–1085.

[5] M. Meila, J. Shi, A random walks view of spectral segmentation, AI and Statistics (AISTATS) (2001).

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

[7] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, M. Ouimet, Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering, in: S. Thrun, L. K. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference, Vol. 16, MIT Press, 2004.

[8] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 214–225.

[9] E. J. Nyström, Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben, Acta Mathematica 54 (1) (1930) 185–204.

[10] C. Alzate, J. A. K. Suykens, A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods, in: Neural Networks, 2006. IJCNN'06. International Joint Conference on, IEEE, 2006, pp. 138–144.

[11] C. Alzate, J. A. K. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2) (2010) 335–347.

[12] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10 (5) (1998) 1299–1319.

[13] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, A support vector machine formulation to pca analysis and its kernel version, IEEE Transactions on Neural Networks 14 (2) (2003) 447–450.

[14] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.

[15] C. J. Burges, et al., Simplified support vector decision rules, in: L. Saitta (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning, Vol. 96, Morgan Kaufmann, 1996, pp. 71–77.

[16] B. Schölkopf, S. Mika, C. J. Burges, P. Knirsch, K. Müller, G. Rätsch, A. J. Smola, Input space versus feature space in kernel-based methods, IEEE Transactions on Neural Networks 10 (5) (1999) 1000–1017.

[17] C. Alzate, J. A. K. Suykens, Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 3556–3563.

[18] C. Alzate, J. A. K. Suykens, Sparse kernel spectral clustering models for large-scale data analysis, Neurocomputing 74 (9) (2011) 1382–1390.

[19] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, Czechoslovak Mathematical Journal 25 (4) (1975) 619–633.

[20] M. Gu, H. Zha, C. Ding, X. He, H. Simon, J. Xia, Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering, Technical Report CSE-01-007 (2001).

[21] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations I, Proceedings of the National Academy of Sciences of the United States of America 35 (11) (1949) 652.

[22] L. Lovász, Random walks on graphs: A survey, Combinatorics, Paul Erdös is eighty 2 (1) (1993) 1–46.

[23] C. Alzate, J. A. K. Suykens, Kernel component analysis using an epsilon-insensitive robust loss function, IEEE Transactions on Neural Networks 19 (9) (2008) 1583–1598.

[24] M. Li, X.-C. Lian, J. T. Kwok, B.-L. Lu, Time and space efficient spectral clustering via column sampling, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 2297–2304.

[25] M. Girolami, Orthogonal series density estimation and the kernel eigenvalue problem, Neural Computation 14 (3) (2002) 669–688.

[26] S. P. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[27] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, The Journal of Machine Learning Research 2 (2002) 243–264.

[28] F. R. Bach, M. I. Jordan, Kernel independent component analysis, The Journal of Machine Learning Research 3 (2003) 1–48.

[29] C. Williams, M. Seeger, The effect of the input density distribution on kernel-based classifiers, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 1159–1166.

[30] S. J. Wright, Modified Cholesky factorizations in interior-point algorithms for linear programming, SIAM Journal on Optimization 9 (4) (1999) 1159–1191.

[31] F. R. Bach, M. I. Jordan, Predictive low-rank decomposition for kernel methods, in: Proceedings of the 22nd international conference on Machine learning, ACM, 2005, pp. 33–40.

[32] K. Frederix, M. Van Barel, Sparse spectral clustering method based on the incomplete Cholesky decomposition, Journal of Computational and Applied Mathematics 237 (1) (2013) 145–161.

[33] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al., LAPACK Users' guide, Siam, 1999.

[34] C. L. Lawson, R. J. Hanson, D. R. Kincaid, F. T. Krogh, Basic linear algebra subprograms for Fortran usage, ACM Transactions on Mathematical Software (TOMS) 5 (3) (1979) 308–323.

[35] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, et al., An updated set of basic linear algebra subprograms (BLAS), ACM Transactions on Mathematical Software 28 (2) (2002) 135–151.

[36] M. Novák, C. Alzate, R. Langone, J. A. K. Suykens, KSC-ICD package user's manual, Internal Report 14-120, Katholieke Universiteit Leuven, ESAT-SISTA, Leuven, Belgium, source codes are available at `http://esca.atomki.hu/~novmis/kscicd/` (2014).

[37] R. Clint Whaley, A. Petitet, J. J. Dongarra, Automated empirical optimizations of software and the atlas project, Parallel Computing 27 (1) (2001) 3–35.

[38] `http://math-atlas.sourceforge.net/`.

[39] L. Hubert, P. Arabie, Comparing partitions, Journal of classification 2 (1) (1985) 193–218.

[40] J. Puzicha, T. Hofmann, J. M. Buhmann, Non-parametric similarity measures for unsupervised texture segmentation and image retrieval, in: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, IEEE, 1997, pp. 267–272.

[41] Y. Saad, Numerical methods for large eigenvalue problems, Vol. 158, SIAM, 1992.

Algorithm 1 Proposed sparse kernel spectral clustering algorithm for training

Input: $\mathcal{K}$ number of desired clusters, a p.d. kernel and its parameters, $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}}$ training set, $G \in \mathbb{R}^{N_{tr} \times R}$ incomplete Cholesky factor as Eq. (42), $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^{R} \subset \mathcal{D}^{tr}$ the corresponding reduced set i.e. pivots selected during the ICD

Output: A sparse clustering model trained on the $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}}$ training set with the $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^{R} \subset \mathcal{D}^{tr}$ reduced set i.e. $\mathcal{CB} = \{\mathbf{c}^{(k)}/\|\mathbf{c}^{(k)}\|_2\}_{k=1}^{\mathcal{K}}$ codebook and $\zeta^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients

I. Calculation of the approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$

1: Calculate $[\tilde{D}]_{ii} = \tilde{d}_i, i = 1, \ldots, N_{tr}$ as $G(G^T \mathbb{1}_{N_{tr}})$
2: Generate $X = \tilde{D}^{-1/2} M_{\tilde{D}} G \in \mathbb{R}^{N_{tr} \times R}$ by using Eq. (56)
3: Perform QR factorization on $X \rightarrow X = Q_X R_X$, with $Q_X \in \mathbb{R}^{N_{tr} \times R}$ and $R_X \in \mathbb{R}^{R \times R}$
4: Perform SVD on $R_X \rightarrow R_X = U_{R_X} \Sigma_{R_X} V_{R_X}^T$ with $U_{R_X}, V_{R_X} \in \mathbb{R}^{R \times R}$ and $\Sigma_{R_X} \in \mathbb{R}^{R \times R}$
5: Form $U_{R_{X_{\cdot,\mathcal{K}-1}}}$ i.e. $\mathcal{K}-1$ left singular vectors from U_{R_X} that corresponds to the $K-1$ largest eigenvalues in $\Lambda_X = \Sigma_{R_X}^2$ and calculate $\tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ eigenvectors as $Q_X U_{R_{X_{\cdot,\mathcal{K}-1}}}$
6: Calculate the approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)} = \tilde{D}^{-\frac{1}{2}} \tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$
7: (If approximated score variables are needed, calculate the approximated bias terms $\tilde{b}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ as described at the end of section 4.2.4)

II. Calculation of the reduced set coefficients $\zeta^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$

8: Generate $\Omega_{\Psi\Phi} \in \mathbb{R}^{R \times N_{tr}}$ and $\Omega_{\Psi\Psi} \in \mathbb{R}^{R \times R}$ kernel matrices by using $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}\}_{i=1}^{N_{tr}}$ and $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^{R} \subset \mathcal{D}^{tr}$ respectively
9: Solve the system $\Omega_{\Psi\Psi} \zeta^{(k)} = \Omega_{\Psi\Phi} \tilde{\boldsymbol{\beta}}^{(k)}$ for $\zeta^{(k)}, k = 1, \ldots, \mathcal{K}-1$

III. Codebook generation based on angular similarities

10: Cluster the $\tau_r^* = [\zeta_r^{(1)}, \ldots, \zeta_r^{(\mathcal{K}-1)}], r = 1, \ldots, R$ points into $\mathcal{K}$ groups $\{C_1, \ldots, C_{\mathcal{K}}\}$ based on their angular similarities
11: Form the $\mathcal{CB} = \{\mathbf{c}^{(k)}/\|\mathbf{c}^{(k)}\|_2\}_{k=1}^{\mathcal{K}}$ codebook based on the $\mathcal{K}$ cluster means $\mathbf{c}^{(k)} = \sum_{r \in C_k} \tau_r^* / |C_k|, k = 1, \ldots, \mathcal{K}$

Algorithm 2 Clustering and out-of-sample extension

Input: A model trained as described in Algorithm 1 ($\mathcal{CB}$ codebook, $\zeta^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients), $\mathcal{R} = \{\tilde{\mathbf{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}$ reduced set, an arbitrary input data point $\mathbf{x}_i \in \mathbb{R}^d$

Output: Assign the $\mathbf{x}_i$ to C_k, where C_k is one of the $\mathcal{K}$ clusters of the model, together with a clustering quality measure

1: Calculate the $\tilde{\mathbf{z}}_i^* - \tilde{\mathbf{b}} = \sum_{r=1}^R K(\tilde{\mathbf{x}}_r, \mathbf{x}_i)(\zeta_r^{(1)}, \ldots, \zeta_r^{(\mathcal{K}-1)})^T \in \mathbb{R}^{\mathcal{K}-1}$ vector
2: Normalize the $\tilde{\mathbf{z}}_i^* - \tilde{\mathbf{b}}$ vector and compare to the elements of $\mathcal{CB}$ by calculating the K distances from the codewords
3: Choose the codeword from $\mathcal{CB}$ that yields the $s^{(1)}$ minimum distance and assign the $\mathbf{x}_i$ point to the corresponding cluster
4: Take the second smallest distance $s^{(2)}$ and assign a clustering quality measure to the $\mathbf{x}_i$ point as $1 - s^{(1)}/s^{(2)}$

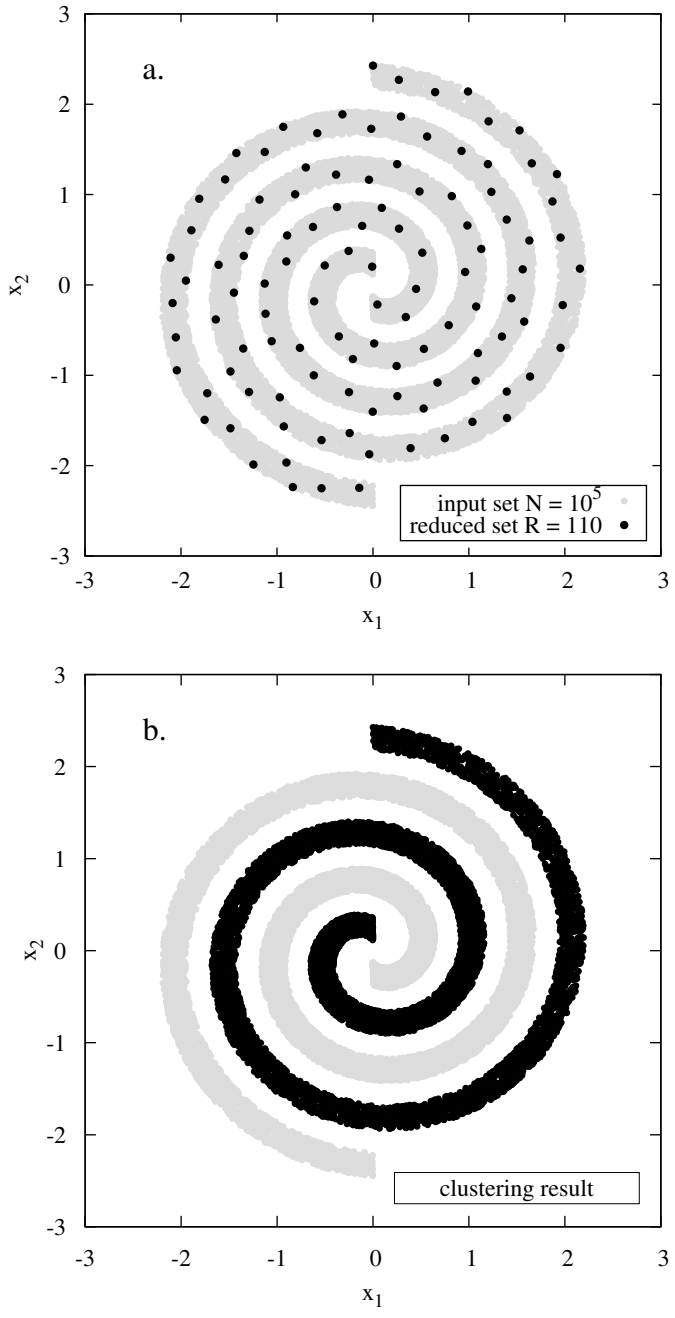


Figure 1: Intertwined spiral with $N = 10^5$ input data points. **a.**: the input data set is shown in gray and the $R = 110$ reduced set points, obtained by applying ICD on the $N_{tr} = N = 10^5$ points with $\sigma^2 = 0.006$ kernel parameter, are shown in black. **b.**: perfect clustering results obtained by using the proposed algorithm given in Algorithms 1 and 2.

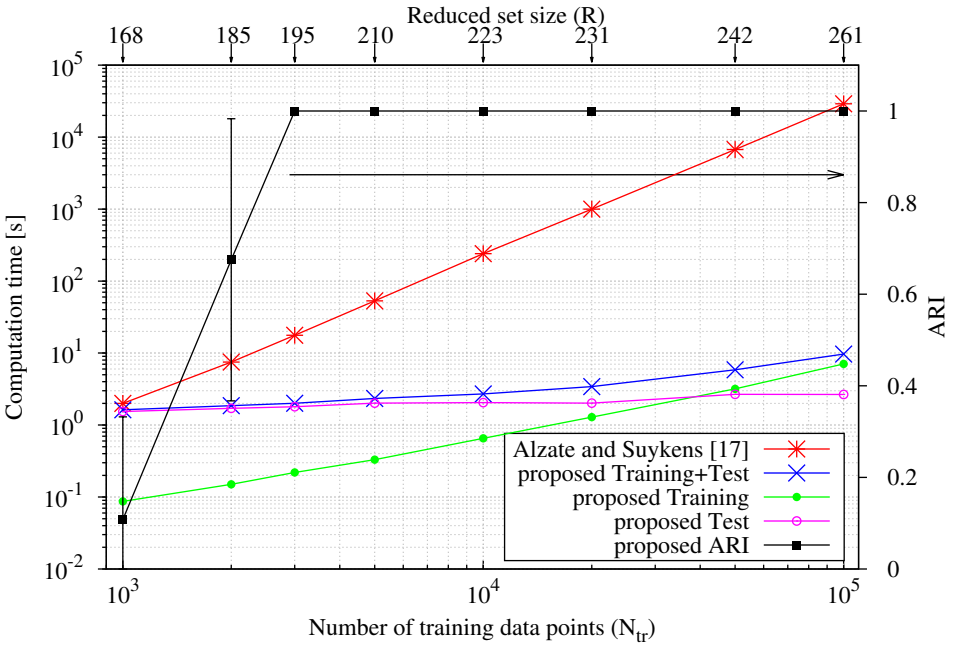


Figure 2: Average computation times of the original [17] and the proposed algorithms based on 10 independent runs on the intertwined spiral problem as a function of the N_{tr} training set size. The same R reduced set size, indicated on the top, was used in both cases of the algorithms. The same kernel parameter $\sigma^2 = 0.006$ was used during the training and the test as well as in the ICD. The test was performed on the full $N_{tr} = 10^5$ data points in all cases and the ARI corresponds to these results. Numerical data are shown in Table 1 for the proposed algorithm.

Table 1: Intertwined spiral: average computational times of the presented algorithm associated to the training (Algorithms 1) and out-of-sample extension (Algorithms 2) parts plotted in Fig. 2. Reduced set size R was fixed to the same value as in [17].

N_{tr}	R	Training [s]	Test [s]	ARI
$1 \cdot 10^3$	168	0.087	1.54	0.11±0.22
$2 \cdot 10^3$	185	0.15	1.71	0.67±0.31
$3 \cdot 10^3$	195	0.21	1.79	1.0
$5 \cdot 10^3$	210	0.33	2.01	1.0
$1 \cdot 10^4$	223	0.65	2.05	1.0
$2 \cdot 10^4$	231	1.29	2.13	1.0
$5 \cdot 10^4$	242	3.18	2.66	1.0
$1 \cdot 10^5$	261	7.27	2.66	1.0

Table 2: Intertwined spiral: minimal reduced set sizes R to get an average $ARI = 1$ based on 10 independent runs and the corresponding computational times of the proposed algorithm as a function of the N_{tr} training set size. The percentage indicates the level of sparsity.

N_{tr}	R	Training [s]	Test [s]	ARI
$3 \cdot 10^3$	180(82%)	0.17	1.66	1.0
$5 \cdot 10^3$	122(97.56%)	0.14	1.11	1.0
$1 \cdot 10^4$	121(98.79%)	0.27	1.12	1.0
$2 \cdot 10^4$	110(99.45%)	0.46	1.0	1.0
$5 \cdot 10^4$	110(99.78%)	1.23	1.0	1.0
$1 \cdot 10^5$	110(99.89%)	2.45	1.0	1.0

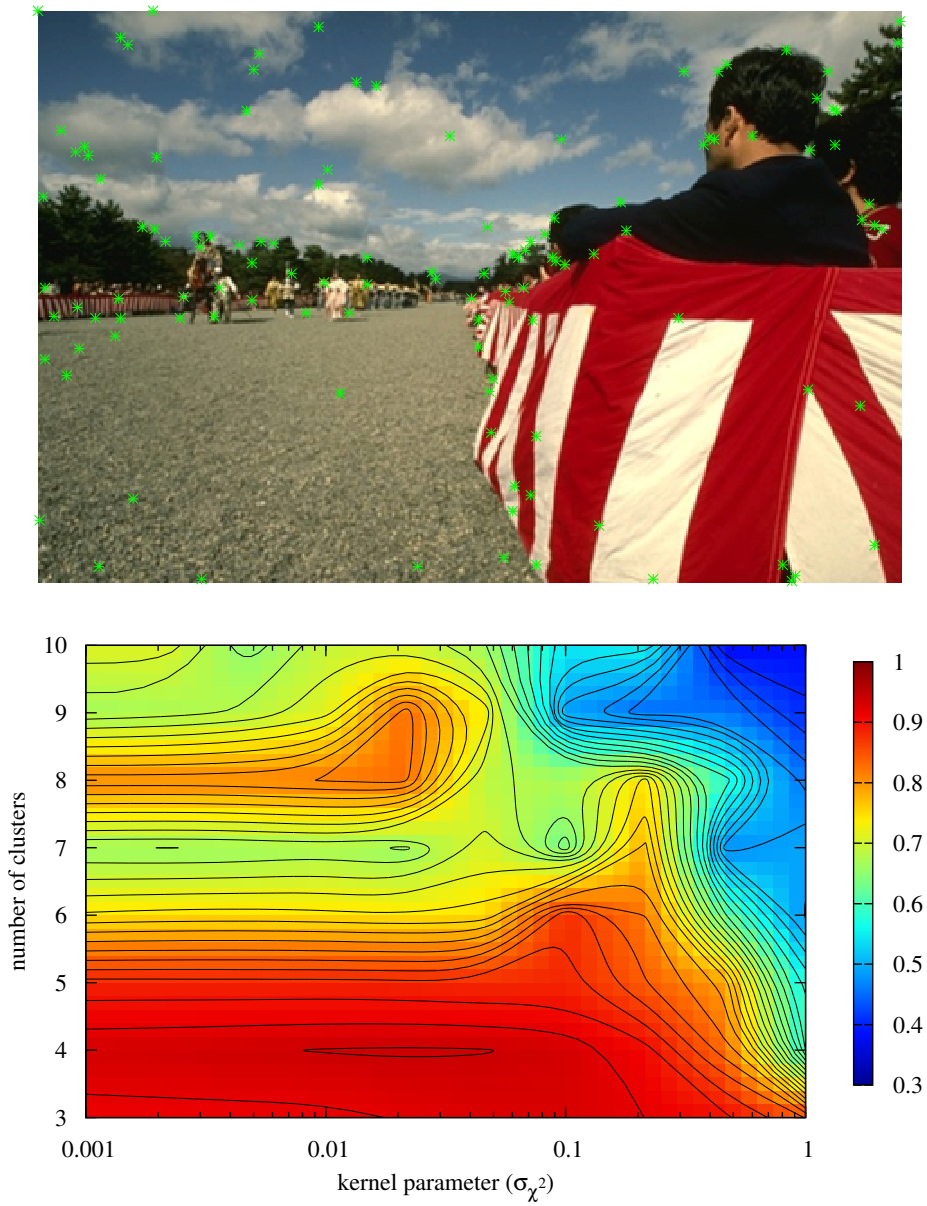


Figure 3: Image segmentation example 1. ID=145086. **top:** original image and the 119 reduced set points selected as pivots by the ICD applying to the $N_{tr} = 10^4$ training data set. **bottom:** values of model selection criterion obtained on the $N_v = 2 \cdot 10^4$ validation set over the $(\mathcal{K} \in \{3, \ldots, 10\}, \sigma_{\chi^2} \in [0.001, 1.0])$ grid (interpolated). The highest value of the model selection criterion 0.942 is given by $\mathcal{K} = 4$ and $\sigma_{\chi^2} = 0.021$ model parameter values.

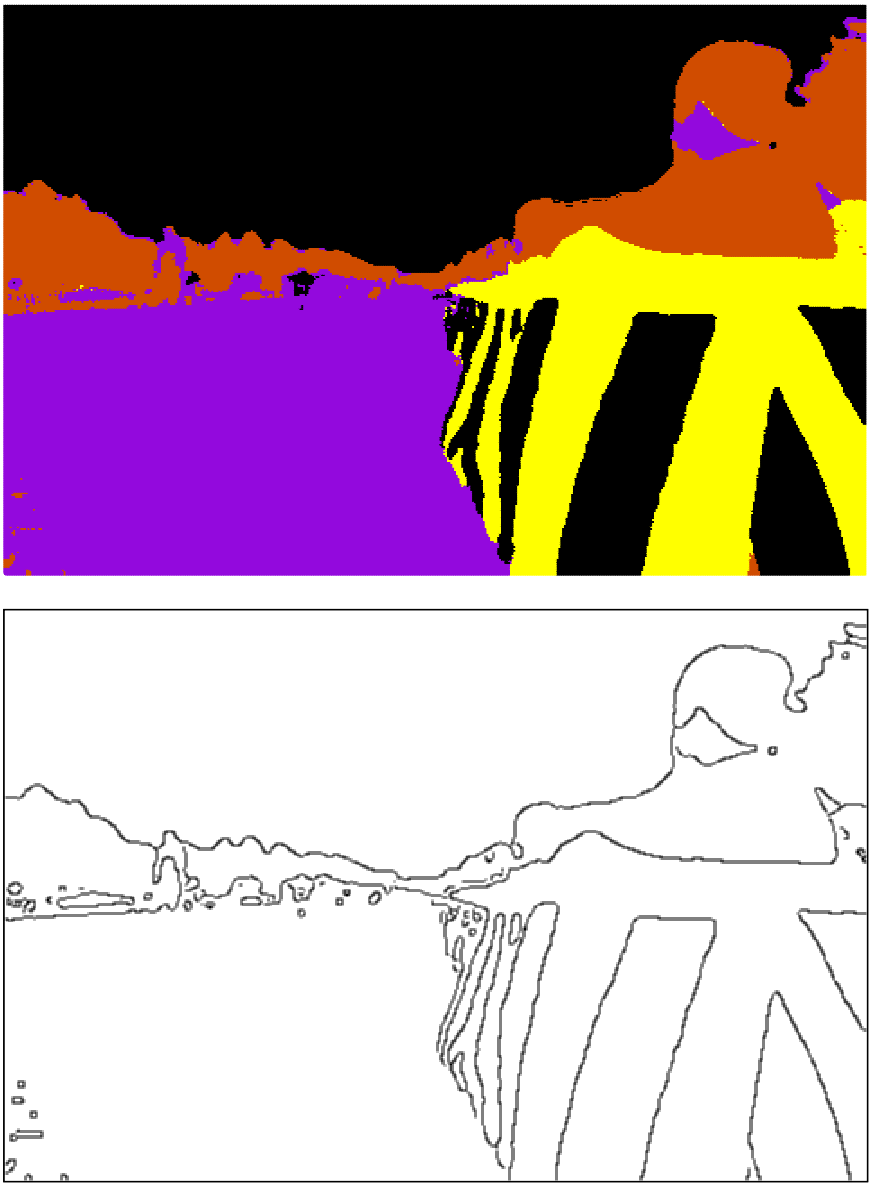


Figure 4: Image segmentation example 2. ID=145086. **top:** clustering result obtained by the proposed sparse model on the whole $N = 154401$ color histogram set. The model was trained on the $N_{tr} = 1 \cdot 10^4$ training data using the $R = 119$ reduced set and $\mathcal{K} = 4$, $\sigma_{\chi^2} = 0.021$ model parameters. **bottom:** the obtained edges derived from the segment-label image shown on the top by using a Canny edge detector. The corresponding F-measure, with respect of human segmentation is 0.9.

Table 3: Berkeley Image Data: $N = 154401$, $N_{tr} = 10000$

	F-measure			Time [s]		
Image ID	Nyström [11]	KSC [11]	Present	Training	Test	R
119082	0.62	0.73	**0.77**	0.90	3.86	196
145086	0.78	0.88	**0.90**	0.33	2.31	119
147091	0.68	0.8	**0.81**	0.31	2.30	116
167062	0.46	0.85	**0.90**	0.25	2.02	98
182053	0.71	0.65	**0.72**	0.60	3.09	183
196073	0.74	0.79	0.79	0.54	3.93	169
295087	0.60	0.72	**0.73**	0.51	3.62	176
3096	0.27	0.72	**0.76**	0.53	3.57	174
42049	0.87	0.88	**0.92**	0.38	2.86	138
62096	0.76	0.78	**0.83**	0.60	4.12	192