Accelerated sparse Kernel Spectral Clustering for large scale data clustering problems

Mihaly Novak, Rocco Langone, Carlos Alzate, Johan Suykens, *Fellow, IEEE,*

***Abstract*—An improved version of the sparse multiway kernel spectral clustering (KSC) is presented in this brief. The original algorithm is derived from weighted kernel principal component (KPCA) analysis formulated within the primal-dual least-squares support vector machine (LS-SVM) framework. Sparsity is achieved then by the combination of the incomplete Cholesky decomposition (ICD) based low rank approximation of the kernel matrix with the so called reduced set method. The original ICD based sparse KSC algorithm was reported to be computationally far too demanding, especially when applied on large scale data clustering problems that actually it was designed for, which has prevented to gain more than simply theoretical relevance so far. This is altered by the modifications reported in this brief that drastically improve the computational characteristics. Solving the alternative, symmetrized version of the computationally most demanding core eigenvalue problem eliminates the necessity of forming and SVD of large matrices during the model construction. This results in solving clustering problems now within seconds that were reported to require hours without altering the results. Furthermore, sparsity is also improved significantly, leading to more compact model representation, increasing further not only the computational efficiency but also the descriptive power. These transform the original, only theoretically relevant ICD based sparse KSC algorithm applicable for large scale practical clustering problems. Theoretical results and improvements are demonstrated by computational experiments on carefully selected synthetic data as well as on real life problems such as image segmentation.**

***Index Terms*—spectral clustering, sparse model, large-scale data, kernel methods, LS-SVM**

I. Introduction

SPECTRAL Clustering (SC) algorithms are known to perform well even in case of complex structures in the data when classical methods fail. This has led to several successful applications in computer vision [1]–[3], load balancing [4], [5], bioinformatics [6], [7], network [8], [9] and many other data analysis and machine learning related problems.

Classical formulations of SC [1], [10]–[14] starts from a graph partitioning problem that is NP-hard due to the discrete constraints on the indicators. By letting the indicator vectors to be real valued, the optimal solution of this relaxed version is provided by some of the eigenvectors of the underlying normalised affinity matrix. The discrete cluster indicators of the original problem can then be inferred from these eigenvectors.

Traditional SC algorithms are typically based on the complete data set due to the lack of a clear extension to unseen data. Since the memory requirement of the affinity matrix as well as the computing time of the corresponding eigenvalue decomposition grow quickly with increasing data size, these SC algorithms are suitable only for relatively small size problems. A possible extension is offered in [15], [16] by using the Nyström method [17]. These algorithms rely only on a subset of the data to obtain an approximation of the implicit eigenfunctions that is used then to cluster the out-of-sample data points. However, the underlying clustering model is unknown and the hyper-parameter selection is done in a heuristic way.

Unlike the traditional graph partition based SC, the so called Kernel Spectral Clustering (KSC) [18], [19] has been formulated as weighted Kernel Principal Component Analysis (KPCA) [20] using the primal-dual Least Square Support Vector Machine (LS-SVM) framework [21], [22]. It has been shown in [18], that the optimal solution of this weighted KPCA problem in the dual space resembles to the *random walk* SC eigenvalue problem [1] when choosing the weights appropriately. However, the weight centered kernel matrix plays the role of the affinity matrix in the case of weighted KPCA hence the name KSC. Unlike the classical SC algorithms, KSC provides model selection criterion for hyper-parameter tuning that can be used to find the optimal number of clusters and kernel parameters. The most important advantage of KSC though is the natural and straightforward way that it offers for out-of-sample extension without relying any Nyström like approximations. This makes possible to construct the clustering model based on a subset of the data while the obtained model can be used then to assign any remaining or new data points to the clusters.

For extending the KSC applicability to large scale data, sparse models have been developed [23], [24] based on the reduced set method [25], [26] and exploiting its out-of-sample extension capability. One [24] relies on an iterative, quadratic Rényi entropy maximisation based subset selection [22], [27] before obtaining the reduced set points. A mathematically more concise solution [23], [24] is built on the Incomplete Cholesky Decomposition (ICD) based low rank approximation of the kernel matrix [28]–[30]. While both outperform the Nyström approximation based sparse version of the traditional SC [24], the ICD based algorithm has the great advantage that it automatically provides the reduced set, used for the sparse

Mihaly Novak is with the Software for Experiments group of the Experimental Physics Department of CERN - 1211 Geneva 23 - Switzerland (e-mail: mihaly.novak@gmail.com)

Rocco Langone was with the Department of Electrical Engineering (ESAT-STADIUS), Katholieke Universiteit Leuven, Leuven B-3001, Belgium (e-mail: roccolangone@hotmail.com) when the research was performed

Carlos Alzate is with AI Fund, Palo Alto, CA 94306, USA (e-mail: carlos.alzate@gmail.com)

Johan Suykens is with the Department of Electrical Engineering (ESAT-STADIUS), Katholieke Universiteit Leuven, Leuven B-3001, Belgium (e-mail: johan.suykens@esat.kuleuven.be)

model construction, while the former requires an additional L_1+L_2 penalisation [26], [31] based step for this. On the other hand, the ICD based sparse KSC was observed [24] to require larger reduced set size (i.e. worse sparsity) compared to the Rényi entropy maximisation based version. More importantly, the ICD based sparse KSC algorithm was found to be computationally far too demanding [23], [24], especially in case of larger data sets which the sparse algorithm was actually designed for which has prevented to gain any practical importance so far. Our main contributions, summarized below, result in a new version of the algorithm that alter this situation:

1) The computationally expensive core part of the original algorithm is replaced with a significantly faster but equivalent alternative. This drastically accelerates the ICD based sparse KSC, especially in case of large data sets, resulting in solving the same clustering problem within seconds that were reported to require hours for the original version without altering the results.
2) A more accurate computation of the approximated bias terms, depending less directly on the reduced set size, is introduced. This leads to a substantially increased sparsity of the obtained clustering model which not only improves further the computational efficiency but also results in a more compact model representation.
3) The theoretical results and improvements are demonstrated by computational experiments on carefully selected synthetic as well as on real life problems such as image segmentation.

II. Problem Formulation

A. SC formulation as weighted KPCA

While all the details can be found elsewhere [18], [19], the most important characteristics of the (none-sparse) multiway KSC is summarized first briefly in order to provide the necessary basis for the corresponding sparse problem formulation.

Given an input data set $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{x}_i \in \mathbb{R}^d$ together with the corresponding weights $\mathcal{V} = \{v_l\}_{l=1}^N, v_l \in \mathbb{R}^+$, the goal of weighted KPCA is to find the directions such that the weighted variance of the projections of the weight centered $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{n_h}$ feature map of $\mathcal{D}$ onto these $\boldsymbol{w}^{(k)} \in \mathbb{R}^{n_h}, k = 1, \ldots, \mathcal{K} < N$ direction vectors is maximal. This leads to the following primal optimisation problem [18], [19]

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

where $\gamma^{(k)} \in \mathbb{R}^+$, $V \in \mathbb{R}^{N\times N}, [V]_{ii} = v_i, i = 1, ..., N$ is the diagonal weight matrix, $\Phi = [\varphi(\boldsymbol{x}_1), \ldots, \varphi(\boldsymbol{x}_N)]^T \in \mathbb{R}^{N\times n_h}$ is the feature map matrix and $\boldsymbol{e}^{(k)} = \Phi \boldsymbol{w}^{(k)} + b^{(k)}\mathbb{1}_N \in \mathbb{R}^N, k = 1, \ldots, \mathcal{K}$ are the error vectors with the $b^{(k)} \in \mathbb{R}, k = 1, \ldots, \mathcal{K}$ bias terms. It can be shown, that using the bias terms leads to the same weighted centering of the feature map as the corresponding explicit centering [21]. The Lagrangian of this constrained optimization problem is

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

with $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^N, k = 1, ..., \mathcal{K}$ Lagrange multiplier vectors. Starting from the Karush-Kuhn-Tucker (KKT) optimality conditions and eliminating the primal variables yield the following eigenvalue problem

$$V M_v \Omega \boldsymbol{\beta}^{(k)} = \lambda^{(k)} \boldsymbol{\beta}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{3}$$

for the optimal solution of the wighted KPCA problem involving the dual variables $\boldsymbol{\beta}^{(k)}$ as eigenvectors. $M_v = I_N - \mathbb{1}_N\mathbb{1}_N^T V/[\mathbb{1}_N^T V \mathbb{1}_N]$ above is the weighted centering matrix, $\Omega = \Phi\Phi^T$ is the kernel matrix with $[\Omega]_{ij} = \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j), i, j = 1, ..., N$ and $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive definite kernel while $\lambda^{(k)} = 1/\gamma^{(k)}$. It can be shown easily, that the objective given by (1) can be maximised by taking the $\mathcal{K}$ leading eigenvectors of the $VM_v\Omega\boldsymbol{\beta}^{(k)}$ matrix.

The KKT optimality conditions (Eqs.(11) in [19]) provide the $\boldsymbol{w}^{(k)} = \Phi^T\boldsymbol{\beta}^{(k)}$ connection between the primal - dual solutions as well as the $b^{(k)} = -\mathbb{1}_N^T V\Phi\Phi^T\boldsymbol{\beta}^{(k)}/[\mathbb{1}_N^T V\mathbb{1}_N]$ expression for the bias terms. These yield $\boldsymbol{z}^{(k)} = \Phi\boldsymbol{w}^{(k)} + b^{(k)}\mathbb{1}_N = M_v\Omega\boldsymbol{\beta}^{(k)} \in \mathbb{R}^N$ for the k-th score variable, i.e. projection of the weight centered feature map of the input data set $\mathcal{D}$ onto the k-th optimal direction vector.

As discussed in depth in [18], [19], (3) becomes

$$D^{-1}M_D\Omega\boldsymbol{\beta}^{(k)} = \lambda^{(k)}\boldsymbol{\beta}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{4}$$

when using the $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ kernel to measure the pairwise similarities between the $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}$ input data points and choosing the corresponding inverse degree matrix $V = D^{-1}, \mathrm{diag}(D) = \Omega\mathbb{1}_N$ as the weight matrix. This resembles to the eigenproblem of a classical SC algorithm, introduced in [1] based on the normalized *random walk* graph Laplacian, with the only difference that now the weight centered kernel matrix plays the role of the affinity matrix. However, the properties of the eigenvectors $\boldsymbol{\beta}^{(k)}$ are different now from those obtained during the related classical SC algorithm due to the weighted centering of the kernel matrix. Nevertheless, the special properties of some of these $\boldsymbol{\beta}^{(k)}$ eigenvectors make possible the clustering interpretation of the weighted KPCA, hence the name Kernel Spectral Clustering (KSC). Construction of the clustering model together with some of its most attractive properties are summarized below. Interested readers can find all details elsewhere [18], [19].

1) Cluster membership encoding-decoding: The special properties of some selected eigenvectors of the $D^{-1}M_D\Omega$ matrix and the corresponding score variables are discussed in detail in [18], [19] under the assumption, that the input data set contains $\mathcal{K}$ clusters. It has been shown, that the more

similar a subset of the input data are the more collinear their $\mathcal{K}-1$ dimensional representations in the subspace spanned by the columns of the $Z = [\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(\mathcal{K}-1)}] \in \mathbb{R}^{N\times\mathcal{K}-1}$ score matrix that corresponds to the $\mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix. Moreover, well separated clusters are mapped into different orthant of this $\mathcal{K}-1$ space. These makes possible the *cluster membership encoding* and model construction either by selecting the $\mathcal{K}$ most frequent $\mathcal{K}-1$ dimensional sign based code words [18], [19], constructed from the rows of Z, or finding $\mathcal{K}$ direction based encoding exploiting the above mentioned collinearity [32]. Either way, the model construction results in $\mathcal{K}$ cluster prototypes that can be used then to assign the individual data points to one of the clusters. This is done by selecting the cluster that yields the minimal distance measured between the $\mathcal{K}$ cluster prototypes and the $\mathcal{K}-1$ dimensional score space representation of a given data point, i.e. the corresponding row of Z. The distance is either Hamming or direction based depending on the selected encoding.

2) Out-of-sample extension and model selection: The fact that KSC includes the above mentioned model construction step before the cluster membership assignment has important consequences.

The first is the natural way that KSC offers for *out-of-sample extension*. This is simple because the KSC model can be constructed based on a $\mathcal{D}^{tr} = \{\boldsymbol{x}_i^{tr}\}_{i=1}^{N_{tr}} \subset \mathcal{D}$ subset of the input data set and the constructed model can be used then to assign any $\boldsymbol{x} \in \mathbb{R}^d$ data point to one of the clusters. The model construction requires the $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{N_{tr}}, \quad k = 1, ..., \mathcal{K}-1$ leading eigenvectors of the corresponding $D^{-1}M_D\Omega \in \mathbb{R}^{N_{tr}\times N_{tr}}$ matrix. Then before the assignment, one needs to compute the

$$z^{(k)}(\boldsymbol{x}) = \varphi(\boldsymbol{x})^T\boldsymbol{w}^{(k)} + b^{(k)} = \sum_{i=1}^{N_{tr}} K(\boldsymbol{x}, \boldsymbol{x}_i)\beta_i^{(k)} + b^{(k)} \quad (5)$$

$k = 1, ..., \mathcal{K}-1$ projections of the given $\boldsymbol{x}$ data point.

The second is the KSC *model selection* capability. As mentioned above, the collinearity of the $\mathcal{K}-1$ dimensional score space representation of the input data, assigned to the same cluster, indicates how well the data set is partitioned into $\mathcal{K}$ clusters using the given kernel parameter and $\mathcal{K}$ cluster number hyper-parameters. This can be exploited for defining model selection criterion that accounts and measures the related collinearity. One can construct KSC models with different hyper-parameter values and then select the one that yields the maximal model selection criterion. Combining this with the out-of-sample extension capability, the model construction can be done based on a *training* subset of the data while a different, *validation* subset can be used for the evaluation of the model selection criteria.

B. Original ICD based sparse KSC

As discussed in the previous section, KSC requires to compute the $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^N, \quad k = 1, ..., \mathcal{K}-1$ leading eigenvectors of the $D^{-1}M_D\Omega$ matrix, assuming $\mathcal{K}$ clusters in the data. The score variables can be computed then based on these eigenvectors, as given by (5), and used for the KSC model construction ($\boldsymbol{x} \in \mathcal{D}$) as well as for clustering any $\boldsymbol{x} \in \mathbb{R}^d$ data point.

While the *out-of-sample extension* capability of KSC offers the $\mathcal{D}^{tr} \subset \mathcal{D}, N_{tr} = |\mathcal{D}^{tr}|$ based model construction, one still would like to use as large $\mathcal{D}^{tr}$ subset as possible in order to incorporate as much information in the model construction as available in the entire $\mathcal{D}$ data set. On the other hand, the size of the related eigenvalue problem (4) grows rapidly with increasing N_{tr} which leads to an intractable problem again in case of large $\mathcal{D}^{tr}$. This requires an approximate solution of the associated eigenvalue problem that is suitable even in case of large N_{tr}.

Furthermore, the primal solutions of the underlying weighted KPCA problem are expressed as linear combinations of the mapped input data $\boldsymbol{w}^{(k)} = \Phi^T\boldsymbol{\beta}^{(k)}, k = 1, \ldots, \mathcal{K}-1$. Since the components of the $\boldsymbol{\beta}^{(k)}$ eigenvectors are usually not zero, each data point contributes to the primal variable $\boldsymbol{w}^{(k)}$ resulting in a non-sparse model. The so-called reduced set method [25], [26] was utilised in [23], [24] to construct the sparse model by finding $\mathcal{R} = \{\tilde{\boldsymbol{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}, R < N_{tr}$ reduced set points and the corresponding $\boldsymbol{\xi}^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients such that $\boldsymbol{w}^{(k)} = \Phi^T\boldsymbol{\beta}^{(k)} \approx \tilde{\boldsymbol{w}}^{(k)} = \Psi^T\boldsymbol{\xi}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ with $\Psi = [\varphi(\tilde{\boldsymbol{x}}_1), \ldots, \varphi(\tilde{\boldsymbol{x}}_R)]^T \in \mathbb{R}^{R\times n_h}$. After finding an appropriate set of reduced set points, the reduced set coefficients are determined by minimizing the squared distance of the approximation that yields the

$$\Omega_{\Psi\Psi}\boldsymbol{\xi}^{(k)} = \Omega_{\Psi\Phi}\tilde{\boldsymbol{\beta}}^{(k)} \quad (6)$$

linear system at the first order optimality 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.

Note, that $\boldsymbol{w}^{(k)} \approx \tilde{\boldsymbol{w}}^{(k)}$ above is nothing more than an approximation based on the linear combination of a small $\{\varphi(\tilde{\boldsymbol{x}}_r)\}_{r=1}^R$ subset of the $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^{N_{tr}}$ feature map vectors. Therefore, finding an appropriate reduced set corresponds to identify the $\mathcal{R} = \{\tilde{\boldsymbol{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}, R < N_{tr}$ points such that the corresponding $\{\varphi(\tilde{\boldsymbol{x}}_r)\}_{r=1}^R$ feature map vectors are linearly independent.

The incomplete Cholesky decomposition of the $\Omega_{\Phi\Phi} = \Phi\Phi^T \in \mathbb{R}^{N_{tr}\times N_{tr}}$ kernel matrix provides solutions to both problems mentioned above. It offers a reduced size, approximate solution of the related eigenvalue problem (4) that is suitable even in case of large N_{tr}. Moreover, it automatically provides an appropriate reduced set $\mathcal{R} = \{\tilde{\boldsymbol{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}, R < N_{tr}$ that can be used for the sparse KSC model construction as discussed above.

1) On the ICD of the kernel matrix: 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 [33] and A can be well approximated by GG^T where $G \in \mathbb{R}^{M\times R}, R \ll M$ [34]. This is the incomplete Cholesky decomposition of A. ICD with symmetric pivoting greedily selects the columns of A and calculates the columns of G such that a lower bound on the actual gain in the corresponding approximation error $\|PAP^T - GG^T\|_1 = Tr(PAP^T - GG^T) = \epsilon$ is maximised [30]. P is the permutation matrix associated to the symmetric pivoting and $\|\cdot\|_1$ is the trace

norm. The algorithm terminates when the approximation error drops below a certain limit $\epsilon \leq \epsilon_{tol}$.

Since the spectrum of the $\Omega_{\Phi\Phi} = \Phi\Phi^T \in \mathbb{R}^{N_{tr}\times N_{tr}}$ kernel matrix decays rapidly in case of many kernels, it can be very often well approximated by $GG^T = \tilde{\Omega}_{\Phi\Phi} \approx \Omega_{\Phi\Phi},\ G \in \mathbb{R}^{N_{tr}\times R}$ with a low $R \ll N_{tr}$ rank [29], [33], [35]. This is utilised in [23], [24] to obtain an approximate, reduced size solution of the eigenvalue problem given by (4), that is suitable even in case of large N_{tr}.

Moreover, as the Cholesky decomposition of the $\Omega_{\Phi\Phi}$ kernel matrix is equivalent to the QR factorization of the corresponding feature map $\Phi = [\varphi(\boldsymbol{x}_1), \ldots, \varphi(\boldsymbol{x}_{N_{tr}})]^T \in \mathbb{R}^{N_{tr}\times n_h}$ [36], its incomplete version $GG^T = \tilde{\Omega}_{\Phi\Phi} \approx \Omega_{\Phi\Phi}$ can be seen as the result of a truncated, pivoted Gram-Schmidt(GS) orthogonalization of these feature map vectors. The selected pivots, i.e. $\{\varphi(\tilde{\boldsymbol{x}}_r)\}_{r=1}^R$ that correspond to the columns of Ω selected during the ICD, are linearly independent. Therefore, ICD automatically provides a suitable reduced set $\mathcal{R} = \{\tilde{\boldsymbol{x}}_r\}_{r=1}^R$ such that the corresponding $\{\varphi(\tilde{\boldsymbol{x}}_r)\}_{r=1}^R$ linearly independent feature vectors are suitable for the sparse KSC model construction.

2) The original algorithm: As mentioned above, the $GG^T = \tilde{\Omega}_{\Phi\Phi} \approx \Omega_{\Phi\Phi},\ G \in \mathbb{R}^{N_{tr}\times R}$ ICD of the $\mathcal{D}^{tr}$ training data $\Omega_{\Phi\Phi}$ kernel matrix is exploited in two ways in the original sparse KSC algorithm [23], [24].

First, the size of the related eigenvalue problem (4) is reduced from N_{tr} to $R \ll N_{tr}$ by using the the $G = U\Lambda V^T$ Singular Value Decomposition (SVD) in the corresponding low rank approximation of the training data kernel matrix $\Omega_{\Phi\Phi} \approx \tilde{\Omega}_{\Phi\Phi} = GG^T = U\Lambda^2 U^T$ in (4). This leads to the

$$U^T \tilde{D}^{-1} M_{\tilde{D}} U \Lambda^2 \boldsymbol{\gamma}^{(k)} = \tilde{\lambda}^{(k)} \boldsymbol{\gamma}^{(k)}, \quad k = 1, ..., \mathcal{K} \tag{7}$$

eigenvalue problem with $\boldsymbol{\gamma}^{(k)} = U^T \boldsymbol{\beta}^{(k)}$ such that the approximated eigenvectors of the original problem can be obtained as $\tilde{\boldsymbol{\beta}}^{(k)} = U\boldsymbol{\gamma}^{(k)}$. $\tilde{}$ denotes approximation of the corresponding quantity based on the $\Omega_{\Phi\Phi} \approx \tilde{\Omega}_{\Phi\Phi} = GG^T$ low rank approximation.

After the $\tilde{\boldsymbol{\beta}}^{(k)}$ approximated eigenvectors are determined, the sparse solution of the KPCA can be composed that corresponds to the $\boldsymbol{w}^{(k)} = \Phi^T \boldsymbol{\beta}^{(k)} \approx \tilde{\boldsymbol{w}}^{(k)} = \Psi^T \boldsymbol{\xi}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ approximation. The ICD of the training data kernel matrix is utilised again at this point by taking the pivots selected during the decomposition as the $\mathcal{R} = \{\tilde{\boldsymbol{x}}_r\}_{r=1}^R \subset \mathcal{D}^{tr}$ reduced set. The $\boldsymbol{\xi}^{(k)} \in \mathbb{R}^R, k = 1, \ldots, \mathcal{K}-1$ reduced set coefficients can then be determined by solving (6) with the $\Omega_{\Psi\Psi}$ reduced-reduced, $\Omega_{\Psi\Phi}$ reduced-training set kernel matrices and the corresponding $\tilde{\boldsymbol{\beta}}^{(k)}$ approximated eigenvectors.

The approximated score variables of any $\boldsymbol{x} \in \mathbb{R}^d$ data can be expressed as

$$z^{(k)}(\boldsymbol{x}) \approx \tilde{z}^{(k)}(\boldsymbol{x}) = \sum_{r=1}^{R} K(\boldsymbol{x}, \boldsymbol{x}_r)\xi_r^{(k)} + \tilde{b}^{(k)} \tag{8}$$

$k = 1, \ldots, \mathcal{K}-1$ by using the above reduced set based sparse approximation in (5). The corresponding sparse clustering model can then be constructed based on the approximated score variables related to the training data set and any data point can be assigned to one of the underlying clusters as described in sections II-A1 and II-A2.

III. Main Results

A. Modified Algorithm

1) Making it faster: The computational bottleneck of the original ICD based sparse KSC algorithm [23], [24] is the computation of the $\tilde{\boldsymbol{\beta}}^{(k)}$ approximated eigenvectors of the $D^{-1}M_D\Omega$ matrix associated to the $\mathcal{D}^{tr} \subset \mathcal{D}, N_{tr} = |\mathcal{D}^{tr}|$ training data set. As mentioned above, the algorithm first exploits the $GG^T = \tilde{\Omega}_{\Phi\Phi} \approx \Omega_{\Phi\Phi}$ ICD of the training set kernel matrix then the $G = U\Lambda V^T$ SVD of the resulted $G \in \mathbb{R}^{N_{tr}\times R}$ matrix. The corresponding original $N_{tr} \times N_{tr}$ sized eigenvalue problem (4) is then relaxed to solve (7). Indeed, the size of the matrix involved in (7) is $R\times R$ with $R \ll N_{tr}$ that makes possible to use significantly larger N_{tr} training data sets. However, the computation time, required to obtain the corresponding eigenvectors, was reported to increase rapidly with N_{tr} by the authors [23], reaching already to $\sim$hours with $N_{tr} \sim 10^5$. This prevented the algorithm to gain any practical importance as these eigenvectors need to be computed several times during the hyper-parameter tuning.

Having a closer look to the algorithm, one can recognise that the naive computation of the above mentioned $R\times R$ sized matrix, involved in (7), can lead to a $\mathcal{O}(R^2 N_{tr}^2)$ computational complexity. This might be avoided by exploiting both the special structure of the individual matrices and the effects of the corresponding operations ensuring to maintain a $\mathcal{O}(R^2 N_{tr})$ complexity. However, the algorithm still includes the $G = U\Lambda V^T$ SVD of the large $G \in \mathbb{R}^{N_{tr}\times R}$ matrix that keeps the algorithm computationally demanding with increasing N_{tr}. All these complications can be fully avoided by computing the required $\tilde{\boldsymbol{\beta}}^{(k)}$ approximated eigenvectors of the $D^{-1}M_D\Omega$ matrix in the following alternative way.

Instead of performing the SVD of G then constructing and solving (7), one can transform (4) to the corresponding

$$D^{-\frac{1}{2}} M_D \Omega M_D^T D^{-\frac{1}{2}} \boldsymbol{\alpha}^{(k)} = \lambda^{(k)} \boldsymbol{\alpha}^{(k)}, \quad k = 1, \ldots, \mathcal{K}-1 \tag{9}$$

symmetric problem where $\boldsymbol{\alpha}^{(k)} = D^{1/2}\boldsymbol{\beta}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ with the same $\lambda^{(k)}$ eigenvalues as in (4). The $\Omega_{\Phi\Phi} \approx \tilde{\Omega}_{\Phi\Phi} = GG^T$ ICD based approximation of the kernel matrix can be exploited at this point leading to $D^{-\frac{1}{2}} M_D \Omega M_D^T D^{-\frac{1}{2}} \approx \tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} GG^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}}$. Taking $X = \tilde{D}^{-1/2} M_{\tilde{D}} G \in \mathbb{R}^{N_{tr}\times R}$, performing its QR factorisation that leads to $X = Q_X R_X$ with $Q_X \in \mathbb{R}^{N_{tr}\times R}$ and $R_X \in \mathbb{R}^{R\times R}$, then utilising the $R_X = U_{R_X}\Sigma_{R_X}V_{R_X}^T$ SVD of this small R_X matrix result in the following eigenvalue decomposition

$$\tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} GG^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}} = [Q_X U_{R_X}] \Sigma_{R_X}^2 [Q_X U_{R_X}]^T \tag{10}$$

with $Q_X U_{R_X} \in \mathbb{R}^{N_{tr}\times R}$ eigenvectors and $\Sigma_{R_X}^2 = \Lambda_{R_X} \in \mathbb{R}^{R\times R}$ $\mathrm{diag}(\Lambda_{R_X})$ eigenvalues of the $\tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} GG^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}} \in \mathbb{R}^{N_{tr}\times N_{tr}}$ matrix.

Therefore, the $\tilde{\boldsymbol{\alpha}}^{(k)}, k = 1, \ldots, \mathcal{K}-1$ leading eigenvectors of the $\tilde{D}^{-\frac{1}{2}} M_{\tilde{D}} GG^T M_{\tilde{D}}^T \tilde{D}^{-\frac{1}{2}}$ matrix can be obtained by simply taking the $\mathcal{K}-1$ columns of the $Q_X U_{R_X}$ matrix that correspond to the $\mathcal{K}-1$ largest eigenvalues in the diagonal

$\Sigma^2_{R_X} = \Lambda_{R_X}$. Then the $\Omega_{\Phi\Phi} \approx \tilde{\Omega}_{\Phi\Phi} = GG^T$ ICD based approximated eigenvectors $\tilde{\boldsymbol{\beta}}^{(k)}$ of the $D^{-1}M_D\Omega$ matrix can be calculated easily from the corresponding $\tilde{\boldsymbol{\alpha}}^{(k)}$ eigenvectors as $\tilde{\boldsymbol{\beta}}^{(k)} = \tilde{D}^{-1/2}\tilde{\boldsymbol{\alpha}}^{(k)}$. These modifications were motivated by [37] where a similar trick was used to accelerate the computation of some selected eigenvectors of the symmetric normalised graph Laplacian.

It must be noted, that this modified and the original computation of the required approximate eigenvectors of $D^{-1}M_D\Omega$ matrix give identical results. However, the proposed modification has several advantages. First and most importantly, while the original algorithm includes an SVD of the large $G \in \mathbb{R}^{N_{tr}\times R}$, the proposed algorithm performs only a much simpler QR factorisation on the large $X = \tilde{D}^{-1/2}M_{\tilde{D}}G \in \mathbb{R}^{N_{tr}\times R}$ and an SVD only on the small $R_X \in \mathbb{R}^{R\times R}$ which can be done significantly faster. Furthermore, since $\tilde{D}^{-1/2}$ 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. Since the QR factorisation of X can then also be done in place, the proposed algorithm does not need any extra memory. The greatly reduced computation time of the approximate eigenvectors is clearly demonstrated in the next section.

2) Improved sparsity: It has been shown in [24], that the original ICD based sparse KSC algorithm is not only computationally more demanding but also requires remarkably larger reduced set size R compared to its alternative.

As briefly discussed in Section II-A1, clusters are well separated in the score variable space, lying even in different orthant, that eventually makes possible the clustering model construction. This separation is due to the weighted centering of the feature map, achieved through the $b^{(k)}$ bias term, that leads to a similar centering of the score variables in (5) and (8) [19]. Therefore, the accuracy of the $\tilde{b}^{(k)}$ approximated bias term in (8) can greatly influence the quality of the constructed clustering model due to the importance of this centering.

As mentioned in Section (II-A), the KKT optimality conditions provide the expression for the bias term that has the form of $b^{(k)} = -\mathbb{1}_N^T D^{-1}\Omega_{\Phi\Phi}\boldsymbol{\beta}^{(k)}/[\mathbb{1}_N^T D^{-1}\mathbb{1}_N]$ in case of KSC. In the original ICD based sparse KSC algorithm, the corresponding $b^{(k)} \approx \tilde{b}^{(k)}$ approximate values, required in (8), are simple estimated based on the small reduced set, i.e. computing the inverse degrees based on the small $\Omega_{\Psi\Psi} \in \mathbb{R}^{R\times R}$ reduced set kernel matrix and relying on the $\boldsymbol{w}^{(k)} = \Phi^T\boldsymbol{\beta}^{(k)} \approx \Psi^T\boldsymbol{\xi}^{(k)}$ approximation. Therefore, the accuracy of this estimate depends on the R reduced set.

A more accurate value of $\tilde{b}^{(k)}$ can be obtained relying consistently on the $\Omega_{\Phi\Phi} \approx \tilde{\Omega} = GG^T$ ICD based low rank approximation of the entire training set kernel matrix instead of only its reduced subset. This leads to the $D \approx \tilde{D}$, $\mathrm{diag}\,(\tilde{D}) = \tilde{D}\mathbb{1}_{N_{tr}} = \tilde{\Omega}_{\Phi\Phi}\mathbb{1}_{N_{tr}} = G[G^T\mathbb{1}_{N_{tr}}]$, $\boldsymbol{\beta}^{(k)} \approx \tilde{\boldsymbol{\beta}}^{(k)}$ approximations and the simpler $\tilde{b}^{(k)} = [\tilde{\lambda}^{(k)} - 1][\mathbb{1}_{N_{tr}}^T\tilde{D}]\boldsymbol{\beta}^{(k)}/N_{tr}$ expression, also given by KKT optimality conditions. Since all required quantities are already calculated during the computation of the $\tilde{\boldsymbol{\beta}}^{(k)}$ approximate eigenvectors, including the corresponding $\lambda^{(k)} \approx \tilde{\lambda}^{(k)}$ approximate eigenvalues as the diagonals of $\Sigma^2_{R_X} = \Lambda_{R_X}$ in (10), these approximated bias terms can be computed very quickly.

The original algorithm requires a larger R reduced set size just to obtain an estimate of the $\tilde{b}^{(k)}$ bias term in (8) that is good enough to provide an appropriate centering which is essential for an accurate clustering model construction. In contrast, the proposed modification results in more accurate values, depending less directly on R, leading to correct centering in the score variable space. Therefore, the modified algorithm can lead to accurate clustering model even at lower reduced set sizes increasing significantly the KSC model sparsity which is clearly demonstrated in the next section.

B. Computational experiments

1) Implementation and configuration: The modified ICD based sparse KSC algorithm has been implemented in the `leuven` test environment [38], [39] using the C++ object-oriented language. This lightweight framework provides the possibility of utilising the most popular optimised BLAS [40] and LAPACK [41] numerical linear algebra libraries as computing backends. While this multi-threaded implementation can also exploit multiple CPU cores and even NVIDIA GPUs through the appropriate CUDA libraries [42] when available, all reported experiments were carried out using only a single Intel core i7 CPU core with the optimised Intel MKL BLAS/LAPACK backend (v-2019.4.223) [43] on MacOS (High Sierra 10.13.6) system with a Clang (v-1000.11.45.5) compiler.

The sparse KSC model is constructed during the *training phase* based on $N_{tr} = |\mathcal{D}^{tr}| \leq N$ training data points sampled uniformly random from the entire $\mathcal{D}$ data set. The $\mathcal{R} \subset \mathcal{D}^{tr}$ reduced set is formed by the pivots selected during the ICD of the training data kernel matrix. The result of this ICD is the input of the training phase, which includes the computation of the $\tilde{\boldsymbol{\beta}}^{(k)}$ approximated eigenvectors of the $D_{tr}^{-1}M_{D_{tr}}\Omega$ matrix associated to the training data set, the construction of the sparse representation by solving (6) and it terminates by producing the cluster prototypes based on the selected cluster membership encoding as discussed in Section II-A1. This last step relies on the sparse approximation based expression of the score variables as given by (8).

The resulted sparse KSC model can then be used to assign any data points to one of the clusters during the so called *test* or *out-of-sample extension phase* as described in II-A2 relying again on (8).

2) The intertwined spiral synthetic data: The significantly improved computation time is illustrated first by reproducing the same synthetic experiment that was used by the authors of the original algorithm to investigate its characteristics and performance in [23]. The data set contains $N = 10^5$, $d = 2$ dimensional data points, sampled from two intertwined spirals as shown in figure 1.

The average computation time and Adjusted Rand Index(ARI) [44] (based on 10 independent runs), are reported in figure 2 as a function of the N_{tr} training set size. The corresponding run times, obtained by using the original algorithm in [23], are also shown for reference. In order to

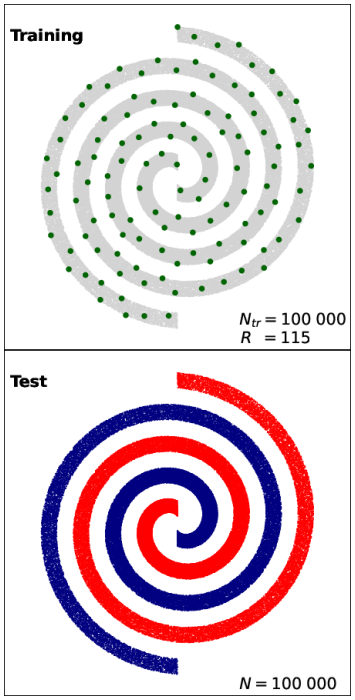

Fig. 1. The intertwined spiral synthetic data. **top**: the entire $N = 10^5$ input data set is shown in grey and the $R = 115$ reduced set points, obtained by applying ICD using the entire $N_{tr} = N = 10^5$ data points with $\gamma = 0.006$ kernel parameter, are shown in green. **bottom**: the perfect clustering result, obtained by using the proposed algorithm. (Note, that this figure corresponds to the last row of Table II.)

TABLE I
NUMERICAL VALUES OF THE PROPOSED ALGORITHM SHOWN IN FIGURE 2.

N_{tr}	R	Training [s]	Test [s]	ARI
1×10^3	168	0.026	0.332	0.036±0.039
2×10^3	185	0.042	0.369	0.903±0.291
3×10^3	195	0.059	0.383	1.0
5×10^3	210	0.093	0.419	1.0
1×10^4	223	0.174	0.442	1.0
2×10^4	231	0.347	0.456	1.0
5×10^4	242	0.976	0.479	1.0
1×10^5	261	2.428	0.514	1.0

ensure comparable characteristics and computation times, the reduced set size R was fixed to the same value at each N_{tr} as in [23]. An RBF kernel was utilised in the form of $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\gamma)$ with kernel parameter of $\gamma_{\text{opt}} = 0.006$ that was determined during a hyper-parameter tuning by maximising the Balanced Line Fit (BLF) model selection criterion [19]. At each different N_{tr} training set sizes, the sparse KSC model was constructed during the training phase based on the N_{tr} training data points and the corresponding R reduced set size while the entire $N = 10^5$ data set was partitioned during the test phase. Numerical values are also shown in Table I. While the original algorithm was reported to require hours to solve this clustering problem when the N_{tr} training set size approaches 10^5 [23], it takes only approximately two seconds for the proposed version with the given implementation using exactly the same configurations.

As mentioned above, the R reduced set sizes were fixed to be the same at each N_{tr} as in [23] in order to ensure

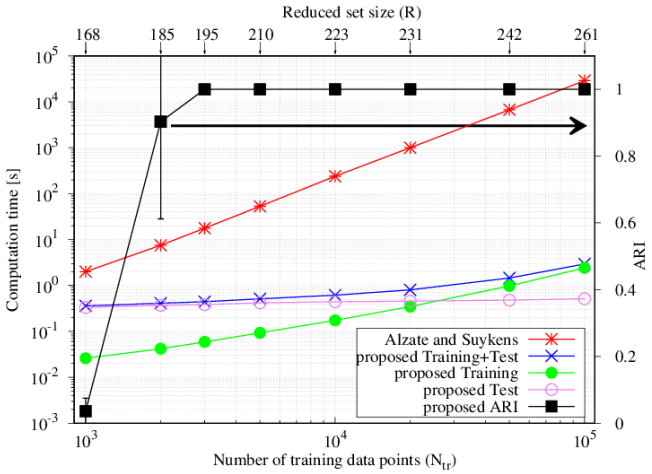

Fig. 2. Spiral : average computation time and ARI of the proposed algorithm (based on 10 independent runs) as a function of the N_{tr} training set size. Numerical values are shown in Table I. The corresponding run times, as reported by Alzate & Suykens in [23], are also shown for reference.

comparable characteristics affecting the computing time. The minimum reduced set sizes R_{min}, required to obtain a perfect clustering (average ARI=1), are reported in Table II using the *original*($\gamma_{\text{opt}} = 0.0102$) and *proposed*($\gamma_{\text{opt}} = 0.006$) versions of computing the approximated bias terms in (8). As discussed in Section III-A2, the accuracy of the bias term approximation influences the quality of the corresponding clustering model. Since the related modification leads to a more accurate approximation, depending less directly on R, the new algorithm requires significantly smaller reduced set sizes for a perfect clustering as demonstrated in Table II. Moreover, the related R_{min} value decreases rapidly between $N_{tr} = 10^3 - 10^4$, then stays constant. This indicates that the corresponding clustering model can well exploit and benefit from the new information incorporated into the increasing training data set while no significant additional information arrives at $N_{tr} > 1 - 2 \times 10^4$. In contrast, the original version not only requires significantly larger reduced set sizes for a perfect clustering, but the corresponding R_{min} value decreases only very mildly with increasing N_{tr}. This shows that the related clustering model accuracy is limited more by the corresponding bias term approximation as it is determined more directly by the reduced set size.

It must be noted, that according to the last row of Table II, the proposed modifications led to a new ICD based sparse KSC algorithm that requires only a second to solve the same clustering problem that was reported to take about 8 hours for the original version in [23].

3) Image segmentation: Ten color images were selected from the Berkeley image data set [45] to demonstrate the performance of the proposed sparse KSC algorithm on a real life problem such as image segmentation.

Each of the RGB color images consists of 321×481 pixels. A local color histogram was computed at each pixel by taking a 5×5 window around the pixels using minimum variance color quantisation of eight levels. After normalisation, the $N = 154\,401$ color histograms serve the 8 dimensional data

TABLE II
SPIRAL: MINIMUM REDUCED SET SIZE R_{MIN} AND CORRESPONDING SPARSITY IN % REQUIRED TO GET A PERFECT CLUSTERING (ARI=1).

N_{tr}	Original	Proposed		
	R_{min}	R_{min}	Training [s]	Test [s]
3×10^3	270 (91.00%)	180 (94.00%)	0.052	0.356
5×10^3	252 (94.96%)	138 (97.24%)	0.056	0.275
1×10^4	251 (97.49%)	121 (98.79%)	0.074	0.243
2×10^4	246 (98.77%)	115 (99.42%)	0.155	0.232
5×10^4	244 (99.51%)	115 (99.77%)	0.411	0.233
1×10^5	244 (99.76%)	115 (99.88%)	0.869	0.193

TABLE III
IMAGE DATA: DETAILS ON THE ICD AND THE HYPER-PARAMETERS.

Image ID	ICD ($R_{\text{max}} = 500$) (input parameters, rank, time)				Optimal hyper-parameters	
	σ_{χ^2}	ϵ_{tol}	R	T[s]	$\mathcal{K}$	σ_{χ^2}
119082	0.07	0.600	196	0.260	3	0.0251
145086	0.05	0.600	119	0.122	4	0.0841
147091	0.50	0.030	116	0.122	2	0.0800
167062	0.01	0.900	98	0.085	3	0.0720
182053	0.20	0.010	179	0.222	5	0.0022
196073	0.05	0.800	169	0.204	2	0.0475
295087	0.07	0.032	196	0.253	3	0.1170
3096	0.01	0.800	142	0.152	3	0.0660
42049	0.01	0.850	138	0.147	3	0.0128
62096	0.20	0.120	192	0.246	2	0.0120

set 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 $\chi^2_{ij} = 0.5 \sum_{l=1}^{8} (h_l^{(i)} - h_l^{(j)})^2 / (h_l^{(i)} + h_l^{(j)})$ [16], [19], [46].

The so called Balanced Angular Similarity(BAS) [39], a direction based cluster membership encoding-decoding similar to the one in [32], was utilised in this experiment. The optimal values of the σ_{χ^2} kernel parameter and $\mathcal{K}$ cluster number hyper-parameters were determined by maximising the corresponding model selection criterion over a $(\mathcal{K} \in \{3, \ldots, 10\}, \sigma_{\chi^2} \in [0.001, 1.0])$ 2D grid. The previously utilised BLF was also evaluated at $\mathcal{K} = 2, 3$ whenever the optimal cluster number was found to be $\mathcal{K} = 3$ (as the BAS based criterion defined only for $\mathcal{K} > 2$). During the hyper-parameter tuning, a sparse KSC model was constructed at each 2D grid point based on the $N_{tr} = 10\ 000$ training data points while the obtained clustering model was used to partition $N_v = 20\ 000$ independent validation points on which the model selection criterion was computed.

The final sparse KSC model was constructed on the training data set using the optimal hyper-parameter values (reported in Table III with some details on the ICD phase) and utilised to partition the entire $N = 154\ 401$ color histograms to produce the image segmentation. The corresponding F-measure, with respect to human segmentation, as a performance criterion is reported in Table IV together with earlier results [19] obtained by using the *dense* version of the KSC and the Nyström method. The present ICD based sparse KSC algorithm provides better F-measure values than the corresponding dense KSC while it has already been reported in [19], that the dense KSC outperforms the Nyström method both in terms of

TABLE IV
IMAGE DATA: F-MEASURE WITH RESPECT TO HUMAN SEGMENTATION ($\mathcal{K}, \sigma_{\chi^2}$ HYPER-PARAMETERS AND R REDUCED SET SIZE OF THE PROPOSED ICD BASED SPARSE KSC ARE REPORTED IN TABLE III).

Image ID	F-measure			Time [s] (proposed)	
	Nyström	KSC	Proposed	Training	Test
119082	0.62	0.73	**0.77**	0.190	1.08
145086	0.78	**0.88**	**0.88**	0.114	0.59
147091	0.68	**0.80**	**0.80**	0.100	0.58
167062	0.46	0.85	**0.89**	0.048	0.48
182053	**0.71**	0.65	**0.71**	0.078	0.93
196073	0.74	**0.79**	0.75	0.143	0.97
295087	0.60	0.72	**0.74**	0.174	1.03
3096	0.27	0.72	**0.78**	0.068	0.67
42049	0.87	0.88	**0.90**	0.105	0.68
62096	0.76	0.78	**0.82**	0.084	1.00

accuracy (F-measure) and speed. As discussed in Section II-B, the dense KSC algorithm is feasible only for relatively small training data sets as it relies on the eigenvalue decomposition of the entire training data kernel matrix. Therefore, the F-measures were achieved in [19] by constructing a dense KSC model on a small $N_{tr} = 2000$ only randomly selected training set then partitioning the entire $N = 154\ 401$ color histograms utilising its out-of-sample extension capability. In contrast, the present sparse KSC model can easily be constructed based on significantly larger $N_{tr} = 10\ 000$ data points exploiting the incorporated information that eventually results in a more accurate clustering model. Moreover, the proposed modifications make now possible to obtain a more compact, sparse clustering model and to perform all the related computations within a fraction of a second demonstrating clearly the benefits of the presented results. It must be noted, that while the proposed algorithm can handle significantly larger training data sets, it has been found that $N_{tr} > 10\ 000$ do not provide additional information that would yield a significant increase in the segmentation quality.

IV. CONCLUSION

An improved version of the sparse multiway Kernel Spectral Clustering (KSC) algorithm, exploiting the incomplete Cholesky decomposition (ICD) based low rank approximation of the kernel matrix, is presented. The computational characteristics of the original algorithm is drastically improved, especially when applied on large scale data, by replacing the computationally most demanding core part with a significantly faster but equivalent alternative. Furthermore, the level of sparsity is also increased significantly by changing the related part of the original algorithm to a new, more accurate approximation leading to more compact clustering model representation. These result in solving clustering problems now within a second that were reported to require hours for the original version with an even more compact clustering model representation with increased descriptive power. This transforms the original, only theoretically relevant ICD based sparse KSC algorithm to applicable for large scale data clustering problems. The theoretical results are demonstrated by computational experiments on carefully chosen synthetic data as well as on real life problems such as image segmentation.

ACKNOWLEDGMENTS

Johan Suykens acknowledges ERC Advanced Grant E-DUALITY (787960), iBOF/23/064, Flemish Government (AI Research Program), Leuven.AI Institute.

REFERENCES

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

[2] X. Zhang, L. Jiao, F. Liu, L. Bo, and M. Gong, "Spectral clustering ensemble applied to sar image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2126–2136, 2008.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[4] L. Ding, F. M. Gonzalez-Longatt, P. Wall, and V. Terzija, "Two-step spectral clustering controlled islanding algorithm," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 75–84, 2012.

[5] C. Alzate and M. Sinn, "Improved electricity load forecasting via kernel spectral clustering of smart meters," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 943–948.

[6] B. Snel, P. Bork, and M. A. Huynen, "The identification of functional modules from the genomic association of genes," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5890–5895, 2002.

[7] D. J. Higham, G. Kalna, and M. Kibble, "Spectral clustering and its use in bioinformatics," *Journal of computational and applied mathematics*, vol. 204, no. 1, pp. 25–37, 2007.

[8] J. Lei and A. Rinaldo, "Consistency of spectral clustering in stochastic block models," *The Annals of Statistics*, vol. 43, no. 1, pp. 215–237, 2015.

[9] F. Liu, D. Choi, L. Xie, and K. Roeder, "Global spectral clustering in dynamic networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 5, pp. 927–932, 2018.

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

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

[12] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[13] M. Meilă and J. Shi, "A random walks view of spectral segmentation," in *International Workshop on Artificial Intelligence and Statistics*. PMLR, 2001, pp. 203–208.

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

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

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

[17] E. J. Nyström, "Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben," *Acta Mathematica*, vol. 54, no. 1, pp. 185–204, 1930.

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

[19] ——, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 335–347, 2010.

[20] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[21] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor, "A support vector machine formulation to pca analysis and its kernel version," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 447–450, 2003.

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

[23] C. Alzate and J. A. K. Suykens, "Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition," in *Proc. IJCNN'08 International Joint Conference on Neural Networks*, 2008, pp. 3556–3563.

[24] ——, "Sparse kernel spectral clustering models for large-scale data analysis," *Neurocomputing*, vol. 74, no. 9, pp. 1382–1390, 2011.

[25] C. J. Burges *et al.*, "Simplified support vector decision rules," in *Proceedings of the Thirteenth International Conference on Machine Learning*, L. Saitta, Ed., vol. 96. Morgan Kaufmann, 1996, pp. 71–77.

[26] B. Schölkopf, S. Mika, C. J. Burges, P. Knirsch, K. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[27] M. Girolami, "Orthogonal series density estimation and the kernel eigenvalue problem," *Neural Computation*, vol. 14, no. 3, pp. 669–688, 2002.

[28] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *The Journal of Machine Learning Research*, vol. 2, pp. 243–264, 2002.

[29] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *The Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2003.

[30] ——, "Predictive low-rank decomposition for kernel methods," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 33–40.

[31] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[32] R. Langone, R. Mall, and J. A. K. Suykens, "Soft kernel spectral clustering," in *Proc. IJCNN'13 International Joint Conference on Neural Networks*, 2013, pp. 1028–1035.

[33] C. Williams and 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.

[34] S. J. Wright, "Modified Cholesky factorizations in interior-point algorithms for linear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1159–1191, 1999.

[35] A. J. Smola, "Sparse greedy matrix approximation for machine learning," in *Proceedings of the 17th international conference on machine learning, June 29-July 2 2000*. Morgan Kaufmann, 2000.

[36] J. Shawe-Taylor, N. Cristianini *et al.*, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[37] K. Frederix and M. Van Barel, "Sparse spectral clustering method based on the incomplete cholesky decomposition," *Journal of Computational and Applied Mathematics*, vol. 237, no. 1, pp. 145–161, 2013.

[38] M. Novák, "mnovak42/leuven: The Leuven library and framework," 2021, accessed on 1 December 2022, https://leuven.readthedocs.io/en/latest/ https://github.com/mnovak42/leuven.

[39] ——, "Sparse Kernel Spectral Clustering Applications - documentation and user guide," 2021, accessed on 1 December 2022, https://leuven-ksc.readthedocs.io/en/latest/.

[40] 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*, vol. 28, no. 2, pp. 135–151, 2002.

[41] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.

[42] NVIDIA, "CUDA, cuBLAS, cuSOLVER," 2020, https://developer.nvidia.com/cuda-toolkit,https://developer.nvidia.com/cublas,https://developer.nvidia.com/cusolver.

[43] INTEL MKL, "Intel Math Kernel Library (MKL)," 2020, https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html.

[44] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[45] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[46] J. Puzicha, T. Hofmann, and J. M. Buhmann, "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 267–272.