Automatic Motion Segmentation via a Cumulative Kernel Representation and Spectral Clustering

Conference Paper · October 2017

DOI: 10.1007/978-3-319-68935-7_44


Omar Oña
Universidad Técnica del Norte


Ana Cristina Umaquinga
Universidad Técnica del Norte


Paul Rosero
Universidad Técnica del Norte


Luis Suárez Zambrano
Universidad Técnica del Norte

Automatic motion segmentation via a cumulative kernel representation and spectral clustering

O. R. Oña-Rocha [1,2], T. Sáchez-Manosalvas[2], A. C. Umaquinga-Criollo[1], P. Rosero-Montalvo[1,4], L. E. Suárez-Zambrano[1], J. L. Rodríguez-Sotelo[5], and D. H. Peluffo-Ordóñez[1,3,6]

[1] Universidad Técnica del Norte - Ecuador,
[2] Universidad de las Fuerzas Armadas - ESPE - Ecuador,
[3] Corporación Universitaria Autónoma de Nariño - Colombia,
[4] Instituto Tecnológico Superior 17 de Julio - Ecuador,
[5] Universidad Autónoma de Manizales - Colombia
[6] Universidad de Nariño - Colombia

Abstract. Dynamic or time-varying data analysis is of great interest in emerging and challenging research on automation and machine learning topics. In particular, motion segmentation is a key stage in the design of dynamic data analysis systems. Despite several studies have addressed this issue, there still does not exist a final solution highly compatible with subsequent clustering/classification tasks. In this work, we propose a motion segmentation compatible with kernel spectral clustering (KSC), here termed KSC-MS, which is based on multiple kernel learning and variable ranking approaches. Proposed KSC-MS is able to automatically segment movements within a dynamic framework while providing robustness to noisy environments.

Keywords: Kernel spectral clustering, motion segmentation, time-varying data, variable ranking.

1 Introduction

The analysis of dynamic or time-varying data has increasingly taken an important place in scientific research, mainly in automation and pattern recognition, being video analysis [1] as well as motion identification for surveillance [2] and body movement classification [3] some of its remarkable applications. In this connection, kernel-based and spectral matrix analysis approaches have arisen as suitable alternatives. For instance, [4] proposes a continuos scheme of weighted kernel principal component analysis (WKPCA) able to capture the dynamic behavior of data. Another study [5] takes advantages of spectral clustering properties to explore the time-varying nature by adding the memory effect into a kernel spectral clustering (KSC) framework [6]. Also, there exists another alternative, known as multiple kernel learning (MKL), which has emerged to deal with different issues in machine learning, mainly, regarding support-vector-machines (SVM) [7] and KSC [8] formulations. The premise that underlies the use of MKL is that learning can be enhanced when using different kernels rather than a single one. Followed from this premise, in this work, we present a KSC-based motion segmentation (KSC-MS) method, that is based on formulation for both MKL and variable ranking. Broadly speaking, KSC-MS works as follows: Firstly, MKL is used in such a manner that kernel matrices are computed from an input data sequence, in which each data matrix represents a frame at a different time instance. Secondly, weighting factors are obtained by ranking each sample contained in the frame in order to form a cumulative kernel matrix as a linear combination of the previously obtained kernels. Herein, to perform such a ranking procedure, we propose an improved alternative to previously introduced approaches used in weighted approaches for dimensionality reduction [8] and motion tracking [9]. Our approach consists of a variable ranking elegantly obtained from a spectral formulation, wherein the eigenvector are those calculated into the KSC process. The positivity of ranking values is clearly guaranteed given the square nature of the proposed formulation. The big advantage of our method is that there is no need for an extra calculation of eigenvectors as it can be formulated within a SVM approach. To show the ability of our approach to automatically segment movements within a dynamic framework, experiments on a moving-curve toy example are performed. As a result, it is demonstrated that our approach can determine the starting and ending of movements as well as its robustness to noisy environments.

The rest of this paper is as follows: Section 2 briefly outlines the so-called kernel spectral clustering. Next, in section 3, we describe the proposed approach KSC-MS. Section 4 presents some experimental results. Finally, in section 5, final concluding remarks are drawn.

2 Kernel spectral clustering

Let us consider $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_N^\top]^\top$, with $\boldsymbol{X} \in \mathbb{R}^{N\times d}$, as the input data matrix, which is to be divided into K disjoint subsets, being $\boldsymbol{x}_i \in \mathbb{R}^d$ the i-th d dimensional data point, N the number of data points, and K the number of desired groups. In this work, we employ the *Kernel Spectral Clustering* (KSC) [6] that is based on least squares- Support Vector Machines (LS-SVM) and can be seen as a weighted kernel principal component analysis, with primal-dual formulation wherein the following clustering model is assumed. Let us define $\boldsymbol{e}^{(l)} \in \mathbb{R}^N$ as the l-th projection vector as the latent variable model in the form:

$$\boldsymbol{e}^{(l)} = \boldsymbol{\Phi}\boldsymbol{w}^{(l)} + b_l \mathbf{1}_N, l \in \{1, \ldots, n_e\}, \tag{1}$$

where $\boldsymbol{w}^{(l)} \in \mathbb{R}^{d_h}$ is the l-th weighting vector, b_l is a bias term, n_e is the number of considered latent variables, notation $\mathbf{1}_N$ stands for a N dimensional all-ones vector, and the matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}(\boldsymbol{x}_1)^\top, \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)^\top\right]^\top$, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times d_h}$, is a high dimensional representation of data. Then, vector $\boldsymbol{e}^{(l)}$ represents the latent variables from a set of n_e binary cluster indicators obtained by $\mathrm{sign}(\boldsymbol{e}^{(l)})$, which are further encoded to obtain the K resultant groups. Therefore, by aiming to cluster input data and considering a least-squares SVM formulation [10] for Eq. (1), we can pose, in matrix terms, the following optimization problem:

$$\max_{\boldsymbol{E},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{2N}\ \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{V}\boldsymbol{E}\boldsymbol{\Gamma}) - \frac{1}{2}\ \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{W}) \tag{2a}$$

$$\text{s. t.} \quad \boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{W} + \mathbf{1}_N \otimes \boldsymbol{b}^\top \tag{2b}$$

where $\boldsymbol{b} = [b_1, \ldots, b_{n_e}]$, $\boldsymbol{b} \in \mathbb{R}^{n_e}$, $\boldsymbol{\Gamma} =$ Diag$([\gamma_1, \ldots, \gamma_{n_e}])$, $\boldsymbol{W} = [\boldsymbol{w}^{(1)}, \cdots, \boldsymbol{w}^{(n_e)}]$, $\boldsymbol{W} \in \mathbb{R}^{d_h \times n_e}$, $\gamma_l \in \mathbb{R}^+$ is the l-th regularization parameter, $\boldsymbol{V} \in \mathbb{R}^{N\times N}$ is a weighting matrix for projections, and $\boldsymbol{E} = [\boldsymbol{e}^{(1)}, \cdots, \boldsymbol{e}^{(n_e)}]$, $\boldsymbol{E} \in \mathbb{R}^{N\times n_e}$. Notations $\mathrm{tr}(\cdot)$ and $\otimes$ stand for the trace and the Kronecker product, respectively. The Lagrangian of problem Eq. (2) can be expressed as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \boldsymbol{\Gamma}, \boldsymbol{A}) =& \frac{1}{2N}\ \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{V}\boldsymbol{E}) - \frac{1}{2}\ \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{W}) \\ &-\ \mathrm{tr}(\boldsymbol{A}^\top (\boldsymbol{E} - \boldsymbol{\Phi}\boldsymbol{W} - \mathbf{1}_N \otimes \boldsymbol{b}^\top))\end{aligned} \tag{3}$$

where matrix $\boldsymbol{A} \in \mathbb{R}^{N\times n_e}$ is formed by the Lagrange multiplier vectors, i.e., $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)}, \cdots, \boldsymbol{\alpha}^{(n_e)}]$, and $\boldsymbol{\alpha}^{(l)} \in \mathbb{R}^N$ represents the l-th vector of Lagrange multipliers. By solving the partial derivatives of Eq. (3), we yield the following Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned}&\frac{\partial \mathcal{L}}{\partial \boldsymbol{E}} = 0 \quad \Rightarrow \boldsymbol{E} = N\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}, &&\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = 0 \quad \Rightarrow \boldsymbol{W} = \boldsymbol{\Phi}^\top \boldsymbol{A},\\ &\frac{\partial \mathcal{L}}{\partial \boldsymbol{A}} = 0 \quad \Rightarrow \boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{W}, \text{and} &&\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}} = 0 \quad \Rightarrow \boldsymbol{b}^\top \mathbf{1}_N = 0.\end{aligned}$$

Now, by eliminating the primal variables from initial problem of Eq. (2), the following eigenvectors-based dual solution is obtained:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{A}\boldsymbol{V}(\boldsymbol{I}_N + (\mathbf{1}_N \otimes \boldsymbol{b}^\top)(\boldsymbol{\Omega}\boldsymbol{\Lambda})^{-1})\boldsymbol{\Omega}, \tag{4}$$

where $\boldsymbol{\Lambda} =$ Diag$(\boldsymbol{\lambda})$, $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the vector of eigenvalues with $\lambda_l = N/\gamma_l$, $\lambda_l \in \mathbb{R}^+$ and $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$ is a given kernel matrix, such that $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{\Omega}$. In particular, we employ $\boldsymbol{V} = \boldsymbol{D}^{-1}$ and since $K-1$ eigenvectors $\boldsymbol{A}$ are indicators of cluster assignment, value n_e is set to be $k-1$ [11]. Since the condition $\boldsymbol{b}^\top \mathbf{1}_N = 0$ can be satisfied by centering vector $\boldsymbol{b}$, the bias term is chosen in the form $b_l = -1/(\mathbf{1}_N^\top \boldsymbol{V}\mathbf{1}_N)\mathbf{1}_N^\top \boldsymbol{V}\boldsymbol{\Omega}\boldsymbol{\alpha}^{(l)}$. Thus, the solution of problem of Eq. (2) is reformulated to the following eigenvector problem:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{V}\boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{A}, \tag{5}$$

where matrix $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is the centering matrix that is defined as $\boldsymbol{H} = \boldsymbol{I}_N - 1/(\mathbf{1}_N^\top \boldsymbol{V}\mathbf{1}_N)\mathbf{1}_N\mathbf{1}_N^\top \boldsymbol{V}$,($\boldsymbol{I}_N$ denotes a N-dimensional identity matrix) and $\boldsymbol{\Omega} = [\Omega_{ij}]$, $\boldsymbol{\Omega} \in \mathbb{R}^{N\times N}$, being $\Omega_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j), i, j \in 1, \ldots, N$. Notation $\mathcal{K}(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ stands for the introduced kernel function. Consequently, we can calculate the set of projections as follows:

$$\boldsymbol{E} = \boldsymbol{\Omega}\boldsymbol{A} + \mathbf{1}_N \otimes \boldsymbol{b}^\top. \tag{6}$$

Finally, projections are encoded in order to determine the the cluster assignments. Since each cluster is represented by a single point in the $K-1$-dimensional eigenspace, such that those single points are always in different orthants, eigenvectors encoding can be done considering that two points are in the same cluster when they are in the same orthant in the corresponding eigenspace [11]. Thus, a codebook can be obtained from the rows of the matrix containing the $K-1$ binarized leading eigenvectors in the columns, by using $\operatorname{sgn}(\boldsymbol{e}^{(l)})$. Then, matrix $\widetilde{\boldsymbol{E}} = \operatorname{sgn}(\boldsymbol{E})$ becomes the codebook being each row a codeword. Under the same framework, out-of-samples extensions to determine the assignment cluster membership for new testing data can be calculated as follows. Define $\boldsymbol{z} \in \mathbb{R}^{n_e}$ as the projection vector of a testing data point $\boldsymbol{x}_{\text{test}}$. Testing projections can be computed as $\boldsymbol{z} = \boldsymbol{A}^\top \widehat{\boldsymbol{\Omega}}_{\text{test}} + \boldsymbol{b}$, where $\widehat{\boldsymbol{\Omega}}_{\text{test}} \in \mathbb{R}^{n_e}$ is the kernel vector such that $\widehat{\boldsymbol{\Omega}}_{\text{test}} = [\widehat{\Omega}_{\text{test}_1}, \ldots, \widehat{\Omega}_{\text{test}_N}]^\top$, being $\widehat{\Omega}_{\text{test}_i} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_{\text{test}})$. Once the test projection vector $\boldsymbol{z}$ is computed, a decoding stage is carried out that consists of comparing the binarized projections with respect to the codewords in the codebook $\widetilde{\boldsymbol{E}}$ and assigning cluster membership based on minimal Hamming distance [6].

3 KSC-based motion segmentation

The KSC-based motion segmentation (KSC-MS) approach proposed here arises from the combination of KSC with a relevance ranking for feature selection. Such ranking comes from a definition of feature relevance aimed to selecting a subset of features founded on spectral properties of the Laplacian of the data matrix, which is based on a continuous ranking of the features by means of a least-squares maximization problem. The maximization is done over a functional such that sparse solutions for the ranking values are obtained by a spectral decomposition of a non-negative matrix. What makes this approach interesting within the context of dynamic data analysis is the possibility to get a ranking value for each frame in the process of time-varying data analysis. In addition, such feature relevance approach measures the relevance of a subset of features against its influence on the cluster assignment. KSC-MS is aimed to track moving samples or frames matching each frame to an unique meaningful value. This approach may be of helpfulness because of its unsupervised nature, since, in practice, labeling is often not available for motion tracking applications.

Proposed approach works as follows: It starts by clustering the input data by means of KSC with a manually established number of groups and a determined kernel function. Then, we linearly project the high dimensional representation of input data in order to apply a sample relevance ranking process as proposed in [12]. The projection matrix is obtained as a sparse solution of a quadratic optimization problem, where an energy (also called variance) term is maximized. It is proved that projection matrix is the same as the eigenvector matrix given by KSC. Finally, a tracking vector is obtained by a linear combination of such eigenvectors in a similar way as that described in [13].

Similarly as the relevance analysis explained in [14] in which a functional regarding a non-negative matrix is introduced, we pose an optimization problem with the difference that our focus is obtaining the ranking values for samples instead of features, as follows: Consider a data matrix sequence $\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(N_f)}\}$, where N_f is the number of frames and $\boldsymbol{X}^{(t)} = [\boldsymbol{x}_1^{(t)\top}, \ldots, \boldsymbol{x}_N^{(t)\top}]^\top$ is the data matrix associated to time instance

t in size $N \times d$. Also, consider the frame matrix $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N_f \times Nd}$ which is formed in such a way that each row is a frame by letting $\tilde{\boldsymbol{x}}_t \in \mathbb{R}^{Nd}$ be the vectorization of coordinates representing the t-th frame. In other words, $\boldsymbol{\mathcal{X}} = [\tilde{\boldsymbol{x}}_1^\top, \ldots, \tilde{\boldsymbol{x}}_{N_f}^\top]^\top$ and $\tilde{\boldsymbol{x}}_t = \mathrm{vec}(\boldsymbol{X}^{(t)})$. The corresponding kernel matrix can be expressed as $\widetilde{\boldsymbol{\Omega}} \in \mathbb{R}^{N_f \times N_f}$ such that $\widetilde{\Omega}_{ij} = \mathcal{K}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$. Then, the high dimensional representation matrix $\widetilde{\boldsymbol{\Phi}} \in \mathbb{R}^{N_f \times d_h}$ is

$$\widetilde{\boldsymbol{\Phi}} = \left[\boldsymbol{\phi}(\tilde{\boldsymbol{x}}_1)^\top, \ldots, \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_{N_f})^\top\right]^\top,$$

where $\boldsymbol{\phi}(\cdot) : \mathbb{R}^{Nd} \to \mathbb{R}^{d_h}$. Assume a linear projection in the form $\boldsymbol{Z} = \widetilde{\boldsymbol{\Phi}}^\top \boldsymbol{U}$, where $\boldsymbol{U}$ is an orthonormal matrix in size $N_f \times N_f$. Also, a lower rank representation of $\boldsymbol{Z}$ is assumed in the form $\widehat{\boldsymbol{Z}} = \widetilde{\boldsymbol{\Phi}}^\top \widehat{\boldsymbol{U}}$, where $\widehat{\boldsymbol{U}}$ is in size $N_f \times c$ ($c < N_f$). Then, an energy maximization problem can be written as:

$$\max_{\widehat{\boldsymbol{U}}} \ \mathrm{tr}(\widehat{\boldsymbol{U}}^\top \widetilde{\boldsymbol{\Omega}} \widehat{\boldsymbol{U}}) \tag{7a}$$

$$\text{s. t.} \quad \widehat{\boldsymbol{U}}^\top \widehat{\boldsymbol{U}} = \boldsymbol{I}_c. \tag{7b}$$

Indeed, by using the kernel trick, we have

$$\mathrm{tr}(\widehat{\boldsymbol{Z}}^\top \widehat{\boldsymbol{Z}}) = \mathrm{tr}(\widehat{\boldsymbol{U}}^\top \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^\top \widehat{\boldsymbol{U}}) = \mathrm{tr}(\widehat{\boldsymbol{U}}^\top \widetilde{\boldsymbol{\Omega}} \widehat{\boldsymbol{U}}).$$

Then, recalling the KSC dual problem explained in Eq. (5), we can write the centering matrix for frame matrix $\boldsymbol{\mathcal{X}}$, so:

$$\widetilde{\boldsymbol{H}} = \boldsymbol{I}_{N_f} - \frac{1}{\boldsymbol{1}_{N_f}^\top \widetilde{\boldsymbol{V}} \boldsymbol{1}_{N_f}} \boldsymbol{1}_{N_f} \boldsymbol{1}_{N_f}^\top \widetilde{\boldsymbol{V}},$$

where $\widetilde{\boldsymbol{V}}$ is chosen as the degree matrix given by $\widetilde{\boldsymbol{D}} = \mathrm{Diag}(\widetilde{\boldsymbol{\Omega}} \boldsymbol{1}_{N_f})$. Next, by normalizing regarding degree and centering both $\boldsymbol{Z}$ and $\widehat{\boldsymbol{Z}}$, which means to pre-multiply $\widetilde{\boldsymbol{\Phi}}$ by $\widetilde{\boldsymbol{L}} \widetilde{\boldsymbol{V}}^{-1/2}$, we can infer that

$$\mathrm{tr}(\widehat{\boldsymbol{U}}^\top \widetilde{\boldsymbol{\Omega}} \widehat{\boldsymbol{U}}) = \sum_{t=1}^{c} \widetilde{\lambda}_t,$$

where $\widetilde{\boldsymbol{L}}$ comes from the Cholesky decomposition of $\widetilde{\boldsymbol{H}}$ such that $\widetilde{\boldsymbol{L}}^\top \widetilde{\boldsymbol{L}} = \widetilde{\boldsymbol{H}}$ and $\widetilde{\lambda}_l$ is the l-th eigenvalue obtained by KSC when applied over $\boldsymbol{\mathcal{X}}$ with a determined number of clusters $\widetilde{K}$. Therefore, a feasible solution of the problem is $\boldsymbol{U} = \widetilde{\boldsymbol{A}}$, being $\widetilde{\boldsymbol{A}} = [\tilde{\boldsymbol{\alpha}}^{(1)}, \ldots, \tilde{\boldsymbol{\alpha}}^{(c)}]$ the corresponding eigenvector matrix. Thus, c is the same number of considered support vectors $\tilde{n}_e$. Similarly as the MKL approach explained in [13], we introduce a tracking vector $\boldsymbol{\eta} \in \mathbb{R}^{N_f}$ as the solution of minimizing the dissimilarity term given by $\|\widetilde{\boldsymbol{\Phi}} - \widehat{\boldsymbol{\Phi}}\|_F^2$, subject to some orthogonality conditions, being $\widehat{\boldsymbol{\Phi}}$ a lower-rank representation of $\widetilde{\boldsymbol{\Phi}}$. Then, the ranking vector can be calculated by:

$$\boldsymbol{\eta} = \sum_{\ell=1}^{\tilde{n}_e} \tilde{\lambda}_\ell \tilde{\boldsymbol{\alpha}}^{(\ell)} \circ \tilde{\boldsymbol{\alpha}}^{(\ell)}, \tag{8}$$

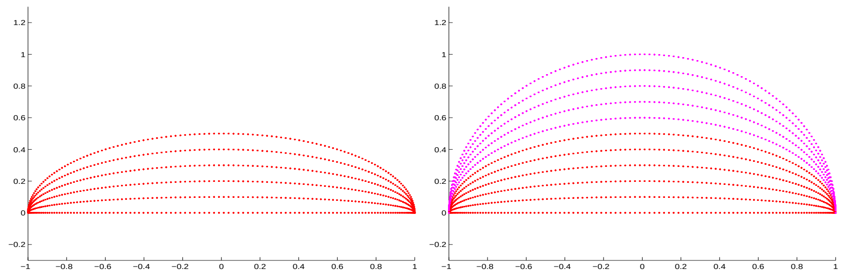
where $\circ$ denotes Hadamard (element-wise) product. Accordingly, the ranking factor η_i is a single value representing an unique frame in a sequence. Notation $\tilde{a}$ means that variable a is related to $\widetilde{\boldsymbol{\Omega}}$. Since $\boldsymbol{\eta}$ comes from a linear combination of the squared eigenvectors being the coefficients the corresponding eigenvalues which are positive, the positivity of tracking vector is guaranteed. In addition, we can normalize the vector by multiplying it by $1/\max|\boldsymbol{\eta}|$ in order to keep the entries of $\boldsymbol{\eta}$ ranged into the interval $[0, 1]$.

4 Experimental results and discussion

To illustrate the performance of the tracking vector, let us consider the following toy example of a moving-curve. At time instance t, the effect of a 2-D curve moving in an arc from down up is emulated by the XY coordinates:

$$\boldsymbol{x}^{(t)} = \begin{pmatrix} |\cos(2\pi\boldsymbol{\tau})|^\top \\ -|\cos(2\pi\boldsymbol{\tau})|^\top \end{pmatrix} \quad \text{and} \quad \boldsymbol{y}^{(t)} = \begin{pmatrix} |t\sin(2\pi\boldsymbol{\tau})|^\top \\ |t\sin(2\pi\boldsymbol{\tau})|^\top \end{pmatrix},$$

where each entry of vector $\boldsymbol{\tau}$ is $\tau_n = n/N$ with $n \in \{1, \dots, N/2\}$, being N the number of samples per frame. Then, we can form the corresponding data matrix $\boldsymbol{X}^{(t)} \in \mathbb{R}^{N\times 2}$ as $\boldsymbol{X}^{(t)} = [\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}]$ as well as the frame sequence $\{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(N_f)}\}$. Then the video effect until a certain frame T is done by keeping the previous frames to show the trace of path followed by the curve. Figure 1 depicts the arc moving effect, when considering $N = 100$, and $N_f = 10$.



(a) Frames from 1 to $N_f/2$ (b) Frames from $N_f/2+1$ to N_f

Fig. 1. 2-D moving-curve

The clustering is done by using KSC aimed to identify two natural movements or clusters ($\widetilde{K} = 2$). To carried out the clustering, KSC is used with a scaled Gaussian kernel $\mathcal{K}(\cdot,\cdot)$ by selecting the 7th $(m = 7)$ nearest neighbor as scaling parameter. The whole frame matrix is used for training, then $\tilde{\boldsymbol{q}}_{\text{train}}$ are the cluster assignment to color the frames according to the found groups, being

$$\tilde{\boldsymbol{q}}_{\text{train}} = \mathrm{KSC}(\boldsymbol{\mathcal{X}}, \mathcal{K}(\cdot,\cdot), \widetilde{K}). \tag{9}$$

To add noise to the moving-curve model, we consider an additive noise to be applied over the Y coordinate in the form $\nu\mathfrak{n}$, where ν is the noise level and $\mathfrak{n} \in \mathbb{R}^N$ is the introduced noise following a Gaussian distribution $\mathfrak{n} \sim \mathcal{N}(0,1)$, so: $\boldsymbol{y}_{\mathfrak{n}}^{(t)} = \boldsymbol{y} + \nu\mathfrak{n}$.

Some examples of the tracking performance are shown in figures 2 and 3 when setting the number of frames to be $N_f = 100$ and the number of samples per frame to be $N = 100$, representing two experiments changing the number of groups and noise conditions. Figure 2 shows the behavior of tracking vector for clustering the frame matrix into 3 clusters ($\widetilde{K} = 3$) while Figure 3 is for $\widetilde{K} = 4$. Tracking vector is calculated in both noisy and non-noisy environments by varying the noise level ν, by assuming that the frame matrix $\boldsymbol{\mathcal{X}}$ is to be split into $\widetilde{K}$ clusters. In both figures, for each considered level noise, clustered sequence at time instance t is shown on the top row as well as the behavior of the corresponding vector plotting on the bottom row. To assess the robustness of this approach, we assess the tracking vector over the noise levels $\nu \in \{0.05, 0.1\}$. No noise case is also considered ($\nu = 0$). For visualization, purposes some meaningful frames from the two experiments are selected. For the first experiments, selected frames are $t \in \{31, 71, 100\}$. Likewise, frames $t \in \{31, 50, 75, 100\}$ are selected for the second experiment. Also, in order to observe the behavior of tracking vector between clusters, frame matrices are clustered with KSC as shown in equation 9 by considering the same conditions mentioned above. As it can be appreciated, the shape of the tracking vector has inflections on the transition from a cluster to the next one. Then, the vector plotting is directly related to the the changes along the frame sequence. In addition, since there are no changes in the tracking vector when adding noise, it is possible to say that is less sensitive to noisy input data.

5 Final remarks

When analyzing a sequence of frames represented by a single data matrix, aiming the identification of underlying dynamic events, kernel-based approaches represent a suitable alternative. Certainly, kernel functions come from an estimation of inner product of high-dimensional representation spaces where clusters are assumed to be separable, and are often defined as a similarity measure between data points from the original space. Such similarity is designed for a local data analysis. In other words, kernels allow a piecewise data exploring by means of an estimation of a generalized variance. Therefore, we can infer that the evolutionary behavior of the sequence can be tracked by some ranking values derived from a kernel-based formulation. Indeed, in this work, we demonstrate that a feasible tracking approach can be accomplished by maximizing an energy term regarding an approximated version of the high-dimensional representation space. We use a linear orthonormal model, being the aim of maximization problem the calculation of an optimal low rank projection or rotation matrix. Finally, taking advantage of the singular value decomposition of kernel matrix, we deduce a tracking vector as a linear combination of the squared eigenvectors. Tracking is done aimed to find an unique value representing adequately each single frame. The here proposed approach (KSC-MS) approach determines a vector that has a direct relationship with the underlying dynamic behavior of the analyzed sequence, allowing even to estimate the number of groups as well as the ground truth.

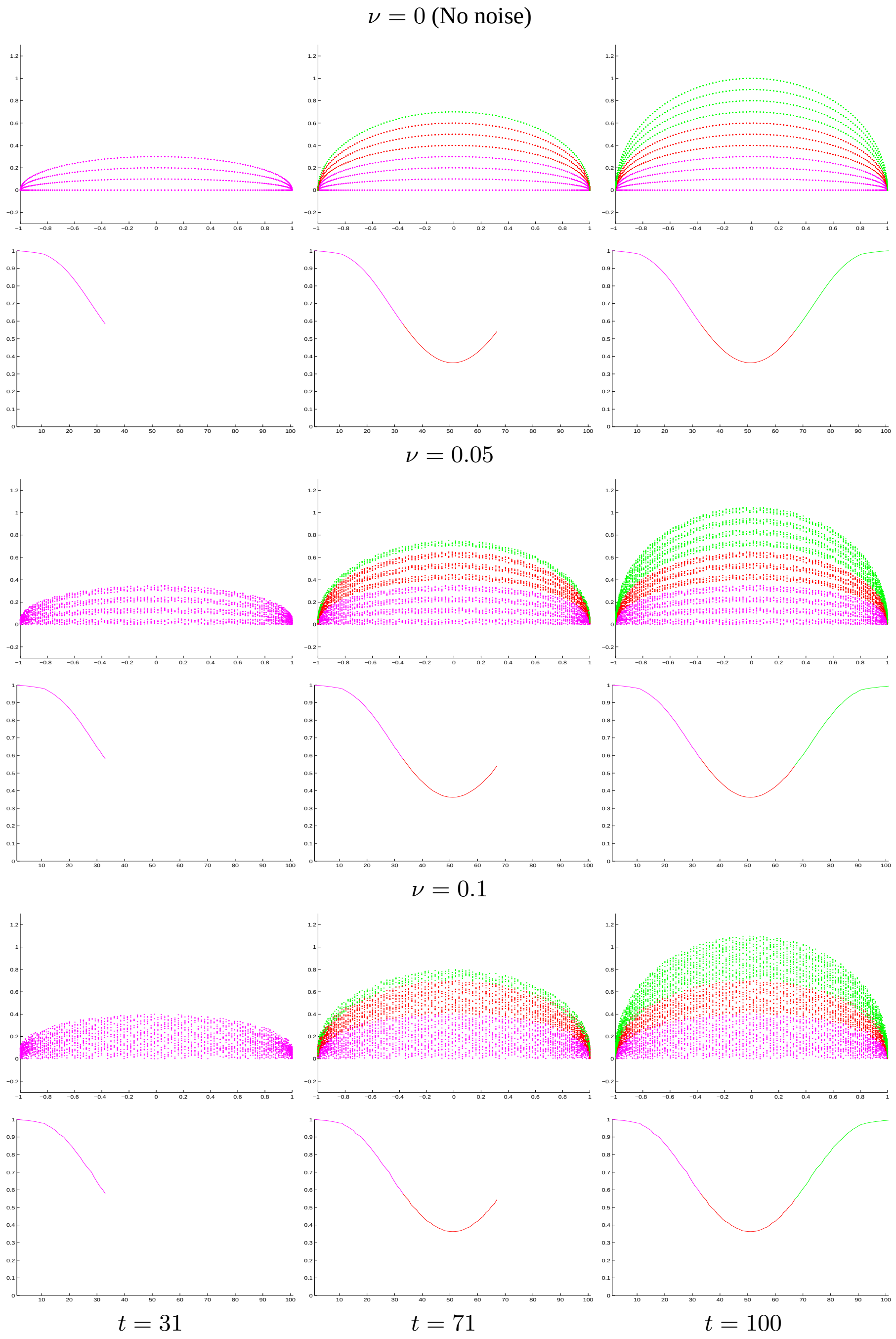


Fig. 2. Clustering of 2-D moving-curve into $\widetilde{K} = 3$ clusters with $N_f = 100$ frames and $N = 100$ samples per frame

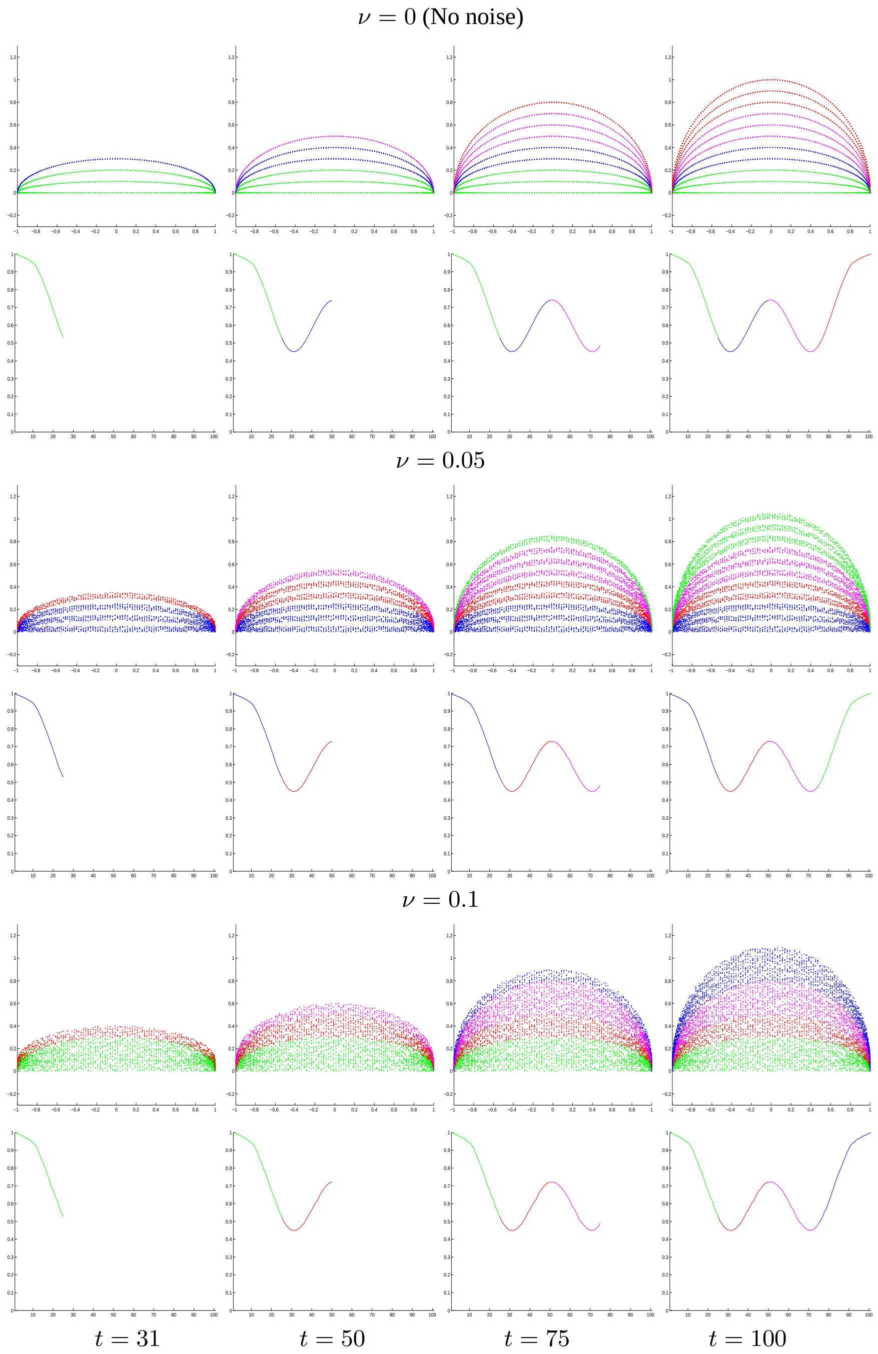


Fig. 3. Clustering of 2-D moving-curve into $\widetilde{K} = 4$ clusters with $N_f = 100$ frames and $N = 100$ samples per frame

In further studies, the true meaning of the amplitude values of the tracking vectors and the dynamic behavior of data is to be formally revealed. As well, other kernel alternatives and properties are to be explored in order to reach low-computational-cost and efficient approaches for motion segmentation and tracking.

References

1. Lu, L., Zhang, X., Xu, X., Shang, D.: Video analysis using spatiotemporal descriptor and kernel extreme learning machine for lip reading. Journal of Electronic Imaging **24** (2015) 053023–053023
2. Yadav, S., Dubey, R., Ahmed, M.: An advanced motion detection algorithm with video quality analysis for video surveillance systems. International Journal of Advanced Research in Computer Science **5** (2014)
3. Saripalle, S.K., Paiva, G.C., Cliett, T.C., Derakhshani, R.R., King, G.W., Lovelace, C.T.: Classification of body movements based on posturographic data. Human movement science **33** (2014) 238–250
4. Shanmao, G., Yunlong, L., Lijun, L., Ni, Z.: Weighted principal component analysis applied to continuous stirred tank reactor system with time-varying. In: Control Conference (CCC), 2015 34th Chinese, IEEE (2015) 6377–6381
5. Langone, R., Alzate, C., Suykens, J.A.: Kernel spectral clustering with memory effect. Physica A: Statistical Mechanics and its Applications (2013)
6. Alzate, C., Suykens, J.A.: Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. IEEE transactions on pattern analysis and machine intelligence **32** (2010) 335–347
7. Bucak, S.S., Jin, R., Jain, A.K.: Multiple kernel learning for visual object recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2014) 1354–1369
8. Peluffo, D., Garcia, S., Langone, R., Suykens, J., Castellanos, G.: Kernel spectral clustering for dynamic data using mulitple kernel learning. In: Proc. of the International Joint Conference on Neural Networks. (2013) 1085–1090
9. Peluffo-Ordónez, D., García-Vega, S., Castellanos-Domínguez, C.G.: Kernel spectral clustering for motion tracking: A first approach. In: International Work-Conference on the Interplay Between Natural and Artificial Computation, Springer (2013) 264–273
10. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
11. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 335–347
12. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. J. Mach. Learn. Res. **6** (2005) 1855–1887
13. Molina-Giraldo, S., Álvarez-Meza, A., Peluffo-Ordoñez, D., Castellanos-Domínguez, G.: Image segmentation based on multi-kernel learning and feature relevance analysis. Advances in Artificial Intelligence–IBERAMIA 2012 (2012) 501–510
14. Peluffo, D., Lee, J., Verleysen, M., Rodríguez-Sotelo, J., Castellanos-Domínguez, G.: Unsupervised relevance analysis for feature extraction and selection: A distance-based approach for feature relevance. In: International conference on pattern recognition, applications and methods-ICPRAM. (2014)