

# Kernel Spectral Clustering for Dynamic Data

Diego Hernán Peluffo-Ordóñez, Sergio García-Vega,  
Andrés Marino Álvarez-Meza, and César Germán Castellanos-Domínguez

Universidad Nacional de Colombia, Sede Manizales  
Signal Processing and Recognition Group  
km 7 vía al Magdalena, Colombia  
{dhpeluffoo, segarciave, amalvarezme, cgcastellanosd}@unal.edu.co  
<http://portal.manizales.unal.edu.co/gta/signal/>

**Abstract.** This paper introduces a novel spectral clustering approach based on kernels to analyze time-varying data. Our approach is developed within a multiple kernel learning framework, which, in this case is assumed as a linear combination model. To perform such linear combination, weighting factors are estimated by a ranking procedure yielding a vector calculated from the eigenvectors-derived-clustering-method. Particularly, the method named kernel spectral clustering is considered. Proposed method is compared to some conventional spectral clustering techniques, namely, kernel k-means and min-cuts. Standard k-means as well. The clustering performance is quantified by the normalized mutual information and Adjusted Rand Index measures. Experimental results prove that proposed approach is an useful tool for both tracking and clustering dynamic data, being able to manage applications for human motion analysis.

**Keywords:** Dynamic data, kernels, support vector machines, spectral clustering.

## 1 Introduction

Undoubtedly, an emergent issue that has become of great interest today for the scientific community on pattern recognition and machine learning is the analysis of the evolutionary or dynamic behavior of time-varying data. There is a lot of highly important applications in which a dynamic analysis is needed, such as human motion analysis and people identification [1], image segmentation [2] and video analysis [3], among others. In this connection, clustering represents a good alternative since it allows for grouping and/or ranking data, mainly, when dealing with unlabeled problems. Due to its versatility, applicability and feasibility, it has been preferred in many approaches. In the literature, many recommended approaches the use of kernels since they allow to incorporate prior knowledge into the clustering procedure. However, the design of a whole kernel-based clustering scheme able to group time-varying data achieving a high accuracy is still an open issue. Taking advantage of kernel-based formulations, some works have been proposed to manage the temporal information for clustering, mainly in segmentation of human motion [4]. Other approaches include either the design of dynamic kernels for clustering [5] or a dynamic kernel principal component analysis (KPCA) based model [6]. Also, another study [7] introduces a variation of the primal

functional of a KPCA formulation for spectral clustering to incorporate the memory effect.

In this paper, aiming to consider the temporal effect of dynamic data, we introduce a spectral clustering approach based on multiple kernel learning. Our approach consists of a variation of the so-called kernel spectral clustering (KSC), introduced in [8], which is a KPCA formulation from least-square support vector machines for clustering being useful to group hardly separable data. It allows for out-of-samples extensions [9]. Multiple kernel learning (MKL) emerges from the premise that the learning can be enhanced when using a set of kernels instead of an unique kernel [10]. The MKL here considered is a linear combination of kernels. Then, after computing the kernel matrices from an input data sequence – in which each data matrix represents a frame –, a cumulative kernel matrix is calculated as a linear combination of the previously obtained kernels. The weighting factors for such linear combinations are estimated by ranking each sample contained in the frame. Needed ranking is accomplished by combining the relevance procedure proposed in [11] as well as the MKL approach presented in [2]. This ranking approach is detailed in [12], in which it is used as a tracking approach. Experiments are carried out using a subject from Graphics Lab Motion Capture Database from Carnegie Mellon University. For comparison purposes, some conventional spectral clustering techniques are also considered, namely, kernel k-means (KKM) and min-cuts (MC) [13]. Also, standard k-means is considered. The normalized mutual information [14] and Adjusted Rand Index [15] metrics are used to quantify the clustering performance.

## 2 Kernel Spectral Clustering (KSC)

Let  $\mathbf{X} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top$  be a data matrix, where  $\mathbf{X} \in \mathbb{R}^{N \times d}$  and  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $\Phi \in \mathbb{R}^{N \times d_h}$ ,  $\Phi = [\phi(\mathbf{x}_1)^\top, \dots, \phi(\mathbf{x}_N)^\top]^\top$ ,  $\phi(\cdot)$  is a function mapping data from the original dimension to a higher one  $d_h$ , that is  $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$ . Then, projection matrix  $\mathbf{E} \in \mathbb{R}^{N \times n_e}$  is grounded in a latent variable model in the form  $\mathbf{E} = \Phi \mathbf{W} + \mathbf{1}_N \otimes \mathbf{b}^\top$ , where  $\otimes$  denotes the Kronecker product, vector  $\mathbf{b}$  holds the bias terms and  $n_e$  denotes the number of considered support vectors. Therefore, the kernel spectral clustering introduced in [9] is aimed to maximize the following expression:

$$\max_{\mathbf{E}, \mathbf{W}, \Gamma} \frac{1}{2N} \text{tr}(\mathbf{E}^\top \mathbf{V} \mathbf{E} \Gamma) - \frac{1}{2} \text{tr}(\mathbf{W}^\top \mathbf{W}) \quad (1a)$$

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

where  $\text{tr}(\cdot)$  denotes the trace,  $\Gamma = \text{Diag}([\gamma_1, \dots, \gamma_{n_e}])$ ,  $\mathbf{W} \in \mathbb{R}^{d_h \times n_e}$ , where  $\mathbf{W} = [w^{(1)}, \dots, w^{(n_e)}]$  and  $\mathbf{E} \in \mathbb{R}^{N \times n_e}$ ,  $\mathbf{E} = [e^{(1)}, \dots, e^{(n_e)}]$ . Problem stated in (1), after forming a Lagrangian and applying the Karush-Kuhn-Tucker conditions becomes to the following dual problem:

$$\mathbf{A} \mathbf{A} = \mathbf{V} \mathbf{H} \Phi \Phi^\top \mathbf{A}, \quad (2)$$

where  $\mathbf{A} = \text{Diag}(\lambda_1, \dots, \lambda_{n_e})$  is a diagonal matrix formed by the eigenvalues  $\lambda_l = N/\gamma_l$ , matrix  $\mathbf{H} \in \mathbb{R}^{N \times N}$  is the centering matrix that is defined as  $\mathbf{H} = \mathbf{I}_N - \mathbf{1}/(\mathbf{1}_N^\top \mathbf{V} \mathbf{1}_N) \mathbf{1}_N \mathbf{1}_N^\top \mathbf{V}$ , where  $\mathbf{I}_N$  a  $N$ -dimensional identity matrix.

From the Mercer's conditions, we have that  $\Phi\Phi^\top = \Omega$ , where  $\Omega \in \mathbb{R}^{N \times N}$  is a kernel matrix such that its entries are given by  $\Omega_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ ,  $i, j \in [N]$ . Therefore, projections can be calculated as follows:

$$\mathbf{E} = \Omega \mathbf{A} + \mathbf{1}_N \otimes \mathbf{b}^\top \quad (3)$$

According to [9], by binaryzing the projection matrix  $\mathbf{E}$ , we obtain a code book  $\tilde{\mathbf{E}}$  as  $\tilde{\mathbf{E}} = \text{sgn}(\mathbf{E})$ , where  $\text{sgn}(\cdot)$  is the sign function. Thus, its corresponding rows are codewords, which allow to form the clusters according to the minimal Hamming distance.

### 3 Dynamic Kernel Spectral Clustering

Dynamic Kernel Spectral Clustering (DKSC), which is an approach based on Kernel Spectral Clustering (KSC) and Multiple Kernel Learning (MKL), works, as follows: Given a sequence of data matrices  $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N_f)}\}$ , being  $\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)T}, \dots, \mathbf{x}_N^{(t)T}]^T$  the data matrix at instance time  $t$ ,  $N_{fm}$  the number of input data (frames), then the corresponding kernel matrices are calculated  $\{\Omega^{(1)}, \dots, \Omega^{(N_f)}\}$  with  $\Omega_{ij}^{(t)} = \mathcal{K}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$ .

Afterwards, MKL is applied with the purpose to obtain accumulated kernel, as a result of the relation between kernel functions and weighting factors (calculated previously). Finally, assuming a certain number of clusters  $K$ , KSC is applied. Since the accumulated kernel matrix is used, the information of the frame at instance  $T$  contains the information of the previous ones. Hence, this approach can be called as dynamic.

#### 3.1 Multiple Kernel Learning (MKL)

Here, multiple kernel learning is applied to obtain a linear combination of all the input kernel matrices until the current matrix. At instance  $T$ , the cumulative kernel matrix is computed as follows:

$$\check{\Omega}^{(T)} = \sum_{t=1}^T \eta_t \Omega^{(t)} \quad (4)$$

where  $\boldsymbol{\eta} = [\eta_1, \dots, \eta_T]$  are the weighting factors or coefficients and  $\Omega^{(t)}$  is the kernel matrix associated to  $\mathbf{X}^{(t)}$  such that  $\Omega_{ij}^{(t)} = \mathcal{K}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$ . Nonetheless, the cumulative kernel matrix can be also calculated in a recursive fashion as:  $\check{\Omega}^{(T)} = \check{\Omega}^{(T-1)} + \eta_T \Omega^{(T)}$  with  $\check{\Omega}^{(0)} = \mathbf{0}_{N \times N}$ .

#### 3.2 Weighting Factors

The basic idea is to obtain the ranked values for samples, here, termed tracking vector that works as follow: first, we apply clustering to the input data by means KSC; at this step, it is necessary to establish manually the number of clusters previously and the kernel function, after we linearly project the high dimensional representation of

input data (to obtain a sample relevance ranking process). Then, the projection matrix is obtained as a sparse solution of a quadratic optimization problem, where an energy term is maximized and, finally, a tracking vector is obtained by a linear combination of vectors solving such optimization problem.

Consider the frame matrix  $\mathcal{X} \in \mathbb{R}^{N_f \times N_d}$  where,  $\mathcal{X} = [\tilde{\mathbf{x}}_1^\top, \dots, \tilde{\mathbf{x}}_{N_f}^\top]^\top$  and  $\tilde{\mathbf{x}}_t = \text{vec}(\mathbf{X}^{(t)})$ . The corresponding kernel matrix can be expressed as  $\tilde{\Omega} \in \mathbb{R}^{N_f \times N_f}$  such that  $\tilde{\Omega}_{ij} = \mathcal{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ . Then, the high dimensional representation matrix  $\tilde{\Phi} \in \mathbb{R}^{N_f \times d_h}$  is  $\tilde{\Phi} = [\phi(\tilde{\mathbf{x}}_1)^\top, \dots, \phi(\tilde{\mathbf{x}}_{N_f})^\top]^\top$ , where  $\phi(\cdot) : \mathbb{R}^{N_d} \rightarrow \mathbb{R}^{d_h}$ . Moreover, we assume a linear projection in the form  $\mathbf{Z} = \tilde{\Phi}^\top \mathbf{U}$ , where  $\mathbf{U} \in \mathbb{R}^{N_f \times N_f}$  is an orthonormal matrix. Likewise, a lower rank representation of  $\mathbf{Z}$  is assumed in the form  $\hat{\mathbf{Z}} = \tilde{\Phi}^\top \hat{\mathbf{U}}$ , where  $\hat{\mathbf{U}} \in \mathbb{R}^{N_f \times c}$  ( $c < N_f$ ). Therefore, an energy maximization problem can be written as:

$$\max_{\hat{\mathbf{U}}} \text{tr}(\hat{\mathbf{U}}^\top \tilde{\Omega} \hat{\mathbf{U}}) \quad \text{s.t.} \quad \hat{\mathbf{U}}^\top \hat{\mathbf{U}} = \mathbf{I}_c \quad (5a)$$

Indeed, by using the kernel trick, we have  $\text{tr}(\hat{\mathbf{Z}}^\top \hat{\mathbf{Z}}) = \text{tr}(\hat{\mathbf{U}}^\top \tilde{\Omega} \hat{\mathbf{U}})$ . Furthermore, the KSC allows centering matrix for frame matrix  $\mathcal{X}$ . Therefore, normalizing regarding the degree and centering both  $\mathbf{Z}$  and  $\hat{\mathbf{Z}}$ , which means to pre-multiply  $\tilde{\Phi}$  by  $\tilde{\mathbf{L}}\mathbf{V}^{-1/2}$ , we can infer that  $\text{tr}(\hat{\mathbf{U}}^\top \tilde{\Omega} \hat{\mathbf{U}}) = \sum_{t=1}^c \tilde{\lambda}_t$ , where  $\tilde{\mathbf{L}}$  comes from the Cholesky decomposition of  $\tilde{\mathbf{H}}$  such that  $\tilde{\mathbf{L}}^\top \tilde{\mathbf{L}} = \tilde{\mathbf{H}}$  and  $\tilde{\lambda}_l$  is the  $l$ -th eigenvalue obtained by KSC when applied over  $\mathcal{X}$  with a determined number of clusters  $\tilde{K}$ . Therefore, a feasible solution of the problem is  $\mathbf{U} = \hat{\mathbf{A}}$ , being  $\hat{\mathbf{A}} = [\tilde{\alpha}^{(1)}, \dots, \tilde{\alpha}^{(c)}]$  the corresponding eigenvector matrix. Thus,  $c$  is the same number of considered support vectors  $\tilde{n}_e$ .

We introduce a tracking vector  $\boldsymbol{\eta} \in \mathbb{R}^{N_f}$  as the solution of minimizing the dissimilarity term given by  $\|\tilde{\Phi} - \hat{\Phi}\|_F^2$ , subject to some orthogonality conditions, being  $\hat{\Phi}$  a lower-rank representation of  $\tilde{\Phi}$ . Then, the ranked vector can be calculated by:

$$\boldsymbol{\eta} = \sum_{\ell=1}^{\tilde{n}_e} \tilde{\lambda}_\ell \tilde{\alpha}^{(\ell)} \circ \tilde{\alpha}^{(\ell)} \quad (6)$$

where  $\circ$  denotes Hadamard (element-wise) product. Accordingly, the ranking factor  $\eta_t$  is a single value representing an unique frame in a sequence. Notation  $\tilde{a}$  means that variable  $a$  is related to  $\tilde{\Omega}$ . Finally, it is possible to normalize the vector by multiplying it by  $1/\max|\boldsymbol{\eta}|$  to keep the entries of  $\boldsymbol{\eta}$  ranged into the interval  $[0, 1]$ .

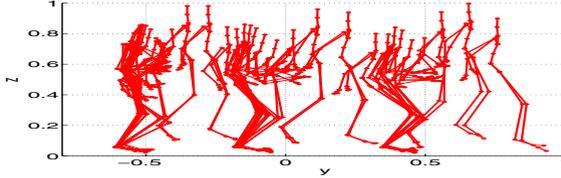
### 3.3 Clustering

The proposed DKSC works as follows: Let  $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N_f)}\}$  be a sequence of data matrices, being  $N_f$  the number of frames, the corresponding kernel matrices are computed with  $\Omega_{ij}^{(t)} = \mathcal{K}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$ . Then, the weighting factor  $\boldsymbol{\eta}$  is calculated from equation 6 over the frame matrix  $\mathcal{X}$ . Further, we obtain the accumulated kernel matrices  $\{\hat{\Omega}^{(1)}, \dots, \hat{\Omega}^{(N_f)}\}$  and, lastly, the KSC is applied over each pair  $(\mathbf{X}^{(t)}, \hat{\Omega}^{(t)})$ , where  $t \in \{1, \dots, N_f\}$ , achieving as a result the cluster assignment vector  $\hat{\mathbf{q}}_{\text{train}}^{(t)}$ .

## 4 Experimental Set-Up

### 4.1 Considered Database

*Motion caption Database:* The data used in this work was obtained from *mocap.cs.cmu.edu*. The database was created with funding from NSF EIA-0196217. In this work, we use the *subject #1* (progressive jump), for which the three first jumps are considered as shown Figure 1.



**Fig. 1.** Subject #1 (progressive jump)

Four frames are considered, where each frame  $\mathbf{X}^{(t)} \in \mathbb{R}^{160 \times 114}$  has rows containing the vectorization of coordinates  $X$ ,  $Y$ , and  $Z$  of the subject's body points. Therefore  $\hat{\mathbf{x}}_i \in \mathbb{R}^{1 \times 18240}$  and  $\mathcal{X} \in \mathbb{R}^{4 \times 18240}$ . Data matrices from the above database are  $z$ -score normalized regarding their columns.

### 4.2 Applying DKSC

All the experiments are performed for a given set of initial parameters, that is, the number of clusters  $K$  per each frame and the kernel function parameter. To recognize three underlying movements in Motion Caption database (*subject #1*), parameter  $K$  is set to be 3. Kernel matrices associated to the data sequence are calculated by the local-scaled Gaussian kernel [16]. Then, each entry of kernel matrix related to frame  $t$  is given by:  $\Omega_{ij}^{(t)} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / \sigma_i \sigma_j)$ , where  $\|\cdot\|$  denotes the Euclidean norm and the scale parameter  $\sigma_i$  is chosen as the Euclidean distance between the sample  $\mathbf{x}_i$  and its corresponding  $m$ -th nearest neighbor. Free parameter  $m$  is empirically set as that one supplying the greatest Fisher's criterion value. For both databases, the value  $m = 10$  is adjusted. To compute the lower-rank matrix  $\hat{\Omega}$ , a scaled Gaussian kernel is applied. Afterwards, the clustering for the pair  $(\mathcal{X}, \hat{\Omega})$  is carried out by setting the number  $K = N_{fm}$  whereas  $m$  is fixed as the entire closest number to  $0.1N_{fm}$  in case of Motion Caption.

In order to numerically assess the performance of DKSC against other methods, we suggest an estimation of the ground truth. Since the tracking vector plotting depicts a concave mode for each identified dynamic event, we can suggest that the ground truth can be estimated by detecting the beginning and ending of each mode. In this sense, as seen from Fig. 2, we apply a Gaussian fitting process where each mean  $\mu_k$  is located

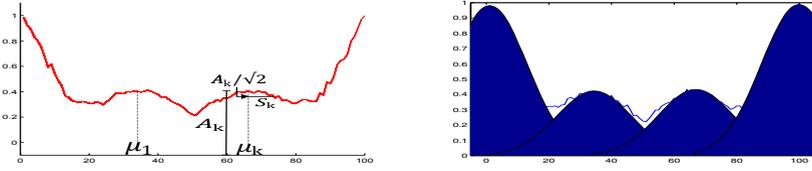


Fig. 2. Gaussian fitting process

at that entry  $t$  of  $\eta$  showing maximum value  $A_k$ . Likewise, the standard deviation  $s_k$  is the distance between  $\mu_k$  and the entry corresponding to  $A_k/\sqrt{2}$ .

For comparison purposes, kernel K-means (KKM) and min-cuts(MC) are also considered [13], which are applied over the data sequence by considering the same MKL approach as considered for KSC. The clustering performance is evaluated by two metrics: normalized mutual information (NMI) [14] and Adjusted Rand Index (ARI) [15]. Both metrics return values ranged into the interval  $[0, 1]$ , where 1 is the best clustering performance.

### 5 Results and Discussion

Motion caption database has not a ground truth to apply label-based metric to assess the clustering performance. However, because weighting factors  $\eta$  are ranking values related to samples, we can consider each instant (man position) as a sample. Then, KSC can be applied to generate the eigenvectors needed to compute  $\eta$ . If analyzing each jump separately, corresponding  $\eta$  vectors should provide information about the clusters contained in the frame (jump). Figure 3 shows the  $\eta$  vector corresponding to each jump.

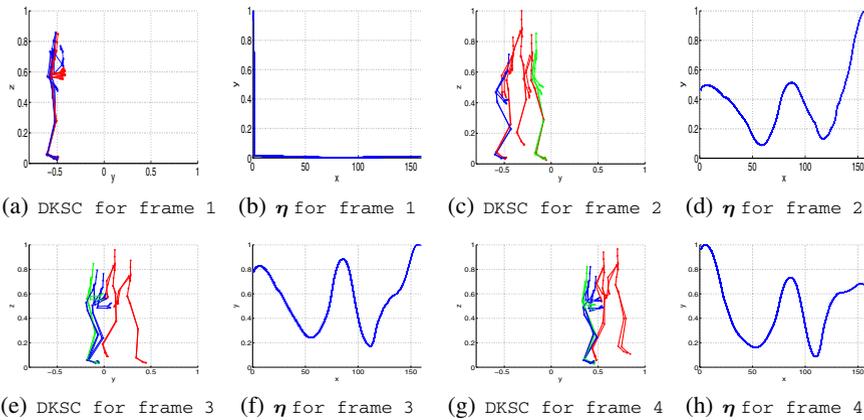


Fig. 3. MKL weighting factors

Besides, we can observe that  $\eta$  has a multi-modal shape. According to (6),  $\eta$  is computed from the eigenvectors  $\alpha^{(l)}$ . Such eigenvectors point out the direction where samples having the most variability measured in terms of a generalized inner product ( $\Phi^T \Phi$ ). Then, we can argue that each mode might represent a different cluster. Under this assumption, we obtain the reference label vectors by detecting the local minima, considering each inflection as a cluster. Results are obtained by comparing the clustering indicators of each method with the determined reference labels. It can be appreciated in Table 1 that our approach reaches greater values than the other methods, then in terms of NMI and ARI it is possible to affirm that DKSC and KSC are suitable approaches to cluster frames in this kind of applications.

**Table 1.** Clustering performance

Measure	Frame	Clustering Method			
		DKSC	KKM	KM	MC
NMI	1	0.6545	0.6583	0.6610	0.7537
	2	0.9046	0.9186	0.7527	0.9046
	3	0.8839	0.9028	0.6245	0.9045
	4	0.8533	0.8498	0.5993	0.5780
ARI	1	0.6583	0.6583	0.6800	0.6832
	2	0.9140	0.9304	0.7468	0.9140
	3	0.9054	0.9060	0.4372	0.9238
	4	0.8680	0.8542	0.4827	0.5101

## 6 Conclusions

A spectral clustering approach for time varying data is introduced. In this sense, the proposed approach is based on multiple kernel learning to keep the evolutionary information by means of a cumulative kernel. Results show that a linear combination of kernels as a MKL approach is enough to cluster dynamic data taking into account past information. Besides, an estimation for coefficients or weighting factors for a linear combination is provided, which is founded on the spectrum of kernel spectral clustering. In addition, we verified that introduced weighting factors give relevant information to both track the dynamic behavior and determine the supposed ground truth. As a future work, we are interested in studying different alternatives to estimate the weighting factors as well as another multiple kernel learning approaches to design clustering approaches able to deal with dynamic data.

**Acknowledgments.** Authors thank to “*Jóvenes Investigadores*” COLCIENCIAS program with the project entitled “*Comparativo de métodos kernel para agrupamiento espectral de datos desde un enfoque primal-dual*” (A comparative study of kernel-based methods for primal-dual formulations aimed spectral clustering).

## References

1. Cong, D.N.T., Khoudour, L., Achard, C., Meurie, C., Lezoray, O.: People re-identification by spectral classification of silhouettes. *Signal Processing* 90(8), 2362–2374 (2010)
2. Rodrigues, M., Gama, J., Ferreira, C.A.: Identifying relationships in transactional data. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) *IBERAMIA 2012*. LNCS, vol. 7637, pp. 81–90. Springer, Heidelberg (2012)
3. Zhang, D.Q., Lin, C.Y., Chang, S.F., Smith, J.R.: Semantic video clustering across sources using bipartite spectral clustering. In: *2004 IEEE International Conference on Multimedia and Expo, ICME 2004*, vol. 1, pp. 117–120. IEEE (2004)
4. Takács, B., Butler, S., Demiris, Y.: Multi-agent behaviour segmentation via spectral clustering. In: *Proceedings of the AAAI-2007, PAIR Workshop*, pp. 74–81 (2007)
5. Keshet, J., Bengio, S.: *Automatic speech and speaker recognition: Large margin and kernel methods*. Wiley (2009)
6. Maestri, M.L., Cassanello, M.C., Horowitz, G.I.: Kernel pca performance in processes with multiple operation modes. *Chemical Product and Process Modeling* 4(5), 7 (2009)
7. Langone, R., Alzate, C., Suykens, J.A.: Kernel spectral clustering with memory effect. *Physica A: Statistical Mechanics and its Applications* (2013)
8. Alzate, C., Suykens, J.A.K.: A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In: *International Joint Conference on Neural Networks, IJCNN 2006*, pp. 138–144. IEEE (2006)
9. 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(2), 335–347 (2010)
10. Huang, H.C., Chuang, Y.Y., Chen, C.S.: Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems* 20(1), 120–134 (2012)
11. 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, 1855–1887 (2005)
12. Peluffo-Ordóñez, D., García-Vega, S., Castellanos-Domínguez, C.G.: Kernel spectral clustering for motion tracking: A first approach. In: Ferrández Vicente, J.M., Álvarez Sánchez, J.R., de la Paz López, F., Toledo Moreo, F. J. (eds.) *IWINAC 2013, Part I*. LNCS, vol. 7930, pp. 264–273. Springer, Heidelberg (2013)
13. Guo, C., Zheng, S., Xie, Y., Hao, W.: A survey on spectral clustering. In: *World Automation Congress (WAC)*, pp. 53–56. IEEE (2012)
14. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
15. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* 1(2), 193–218 (1985)
16. Zelnik-manor, L., Perona, P.: Self-tuning spectral clustering. In: *Advances in Neural Information Processing Systems 17*, pp. 1601–1608. MIT Press (2004)