Incremental kernel spectral clustering for online learning of non-stationary data

Rocco Langone*, Oscar Mauricio Agudelo*, Bart De Moor* and Johan A. K. Suykens*

*Department of Electrical Engineering (ESAT)-STADIUS/iMinds Future Health Department, KU Leuven, B-3001 Leuven Belgium

Email: {rocco.langone,mauricio.agudelo,bart.demoor,johan.suykens}@esat.kuleuven.be

Abstract

In this work a new model for online clustering named Incremental Kernel Spectral Clustering (IKSC) is presented. It is based on Kernel Spectral Clustering (KSC), a model designed in the Least Squares Support Vector Machines (LS-SVMs) framework, with primal-dual setting. The IKSC model is developed to quickly adapt itself to a changing environment, in order to learn evolving clusters with high accuracy. In contrast with other existing incremental spectral clustering approaches, the eigen-updating is performed in a model-based manner, by exploiting one of the Karush-Kuhn-Tucker (KKT) optimality conditions of the KSC problem. We test the capacities of IKSC with some experiments conducted on computer-generated data and a real-world data-set of PM_{10} concentrations registered during a pollution episode occurred in Northern Europe in January 2010. We observe that our model is able to precisely recognize the dynamics of shifting patterns in a non-stationary context.

Keywords: incremental kernel spectral clustering, out-of-sample eigenvectors, LS-SVMs, online clustering, non-stationary data, PM_{10} concentrations.

1. Introduction

In many real-life applications we face the ambitious challenge of online clustering of non-stationary data. Voice and face recognition, community detection of evolving networks such as the World Wide Web or the metabolic pathways in biological cell, object tracking in computer vision, represent just few examples. Therefore researchers perceived the need of developing clustering methods that can model the complex dynamics of evolving patterns in a real-time fashion. Indeed, in the recent past many adaptive clustering models with different inspiration have been proposed: evolutionary spectral clustering techniques [7, 9, 18, 20], self-organizing time map [28], dynamic clustering via multiple kernel learning [27], incremental K-means [8] constitute some examples. Here we focus our attention on the family of the Spectral Clustering (SC) approaches [25, 31, 10], which has shown its practical success in many application domains. SC is an off-line algorithm, and the above-cited attempts to make it applicable to dynamic data-sets, although quite appealing, are at the moment not very computationally efficient. In [26] and more recently in [11], the authors propose some incremental eigenvalue solutions to continuously update the initial eigenvectors found by SC. In this paper, we follow this direction, but with an important difference. The incremental eigen-update we introduce is model-based and cast in a machine learning framework, since our core

model is Kernel Spectral Clustering (KSC, [3]). KSC is an LS-SVM formulation [29] of Spectral Clustering with two main advantages: an organized model-selection procedure based on several criteria (BLF, Modularity, AMS, [3, 17, 19]) and the extension of the clustering model to out-of-sample data. Moreover, it can scale to large data as it has been shown in [23, 24] and very sparse models can be constructed [22, 2]. In KSC a clustering model can be trained on a subset of the data and then applied to the rest of the data in a learning framework. The out-of-sample extension allows then to predict the memberships of a new point thanks to the previously learned model. The out-of-sample extension alone, without the need of ad-hoc eigen-approximation techniques like the ones proposed in [26] and [11], can be used to accurately cluster stationary data-streams. For instance, in [16], KSC has been applied for online fault detection of an industrial machine. In this work KSC was trained offline to recognize two main working regimes, namely good and faulty state. Then it was used in an online fashion via the out-of-sample extension to raise an early warning when necessary.

However, if the data are generated according to some distribution which change over time (i.e. non-stationary), the initial KSC model must be updated. In order to solve this issue we introduce the Incremental Kernel Spectral Clustering Algorithm (IKSC). The IKSC method takes advantage of the work presented in [4] to continuously adjust the initial KSC model over-time, in order to learn the complex dynamics characterizing the non-stationary data.

The remainder of this paper is structured as follows: in Section 2 we briefly recall the KSC model. Section 3 introduces the new IKSC algorithm. Section 4 describes the data-sets used in the experiments. In Section 5 we discuss the simulation results and we compare our method with incremental K-means (IKM).To better understand our technique and the experimental findings we advice the readers to take a look at the demonstrative videos present in the supplementary material of this paper. Finally, Section 6 concludes the article.

2. Kernel Spectral Clustering (KSC)

Spectral clustering methods use the eigenvectors of the graph Laplacian to unfold the data manifold and properly group the data-points. In contrast with classical spectral clustering, KSC is considered in a learning framework. This allows the out-of-sample extension of the clustering model to test points in a straightforward way. With training data $\mathcal{D} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$ and the number of clusters k, the kernel spectral clustering optimization problem can be stated in the following way [3]:

$$\min_{w^{(l)},e^{(l)},b_l} \frac{1}{2}\sum_{l=1}^{k-1} w^{(l)^T} w^{(l)} - \frac{1}{2N}\sum_{l=1}^{k-1} \gamma_l e^{(l)^T} D^{-1} e^{(l)} \tag{1}$$

$$\text{such that } e^{(l)} = \Phi w^{(l)} + b_l 1_N. \tag{2}$$

This is a weighted kernel PCA formulation, being the weighting matrix equal to the degree matrix D associated to the training kernel matrix. The objective consists of minimizing the regularization terms and maximizing the weighted variance of the projections of the data points in the feature space. The score variables[1] are named $e^{(l)} = [e_1^{(l)}, \ldots, e_N^{(l)}]^T, l = 1, \ldots, k-1$ indicates the number

[1] We use interchangeably the terms projections, score variables, latent variables to name the $e^{(l)}$.

of score variables needed to encode the k clusters to find, $D^{-1} \in \mathbb{R}^{N \times N}$ is the inverse of the degree matrix D, Φ is the $N \times d_h$ feature matrix $\Phi = [\varphi(x_1)^T; \ldots; \varphi(x_N)^T]$ and $\gamma_l \in \mathbb{R}^+$ are regularization constants. The multiway clustering model in the primal space is expressed by a set of $k-1$ binary problems, which are combined in an Error Correcting Output Code (ECOC) encoding scheme:

$$e_i^{(l)} = w^{(l)^T} \varphi(x_i) + b_l, i = 1, \ldots, N, l = 1, \ldots, k-1. \tag{3}$$

where $w^{(l)} \in \mathbb{R}^{d_h}$ is the parameter vector in the primal space associated with the l-th binary clustering, b_l are bias terms, $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping of the input points x_i into a high-dimensional feature space of dimension d_h. The projections $e_i^{(l)}$ represent the latent variables of the group of $k-1$ binary clustering indicators given by sign$(e_i^{(l)})$. Thus every point x_i is associated with a latent variable $[e_i^{(1)}, \ldots, e_i^{(k-1)}]$ which lives in the low-dimensional space spanned by $w^{(l)}$. The set of binary indicators sign$(e_i^{(l)}), i = 1, \ldots, N, l = 1, \ldots, k-1$ form a code-book $\mathcal{CB} = \{c_p\}_{p=1}^k$, where each code-word is a binary word of length $k-1$ representing a cluster.

As for all the kernel-based methods, since an explicit formula of the feature map $\varphi(\cdot)$ is in general unknown, the dual of problem (1) is derived. As a consequence, we go from the parametric representation of the clustering model expressed by eq. (3) to a non-parametric representation in the dual space denoted by (5). Here only dot products between the mapped points in $\varphi(\cdot)$ appear, which can be easily computed using the kernel trick derived by the Mercer theorem: $\varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$. In Fig. 1 for the sake of clarity we illustrate, in the case of a synthetic dataset consisting of three intertwined spirals, the points mapped in the space of the eigenvectors $\alpha^{(l)}$ and the space of the latent variables $e^{(l)}$.

The Lagrangian associated with the primal problem, written in matrix form, is:

$$\mathcal{L}(w^{(l)}, e^{(l)}, b_l, \alpha^{(l)}) = \frac{1}{2} \sum_{l=1}^{k-1} w^{(l)^T} w^{(l)} - \frac{1}{2N} \sum_{l=1}^{k-1} \gamma_l e^{(l)^T} D^{-1} e^{(l)} - \sum_{l=1}^{k-1} \alpha^{(l)^T} (e^{(l)} - \Phi w^{(l)} - b_l 1_N)$$

where $\alpha^{(l)}$ are the Lagrange multipliers. The KKT optimality conditions are:

$\frac{\partial \mathcal{L}}{\partial w^{(l)}} = 0 \to w^{(l)} = \Phi^T \alpha^{(l)}$,
$\frac{\partial \mathcal{L}}{\partial e^{(l)}} = 0 \to \alpha^{(l)} = \frac{\gamma_l}{N} D^{-1} e^{(l)}$,
$\frac{\partial \mathcal{L}}{\partial b_l} = 0 \to 1_N^T \alpha^{(l)} = 0$,
$\frac{\partial \mathcal{L}}{\partial \alpha^{(l)}} = 0 \to e^{(l)} - \Phi w^{(l)} - b_l 1_N = 0$.

Once we have solved the KKT conditions for optimality, we can derive the following dual problem:

$$D^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)} \tag{4}$$

where Ω is the kernel matrix with ij-th entry $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$, D is the related graph degree matrix which is diagonal with positive elements $D_{ii} = \sum_j \Omega_{ij}$, M_D is a centering matrix defined as $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}$, $\alpha^{(l)}$ are the dual variables, $\lambda_l = \frac{N}{\gamma_l}$ and K :

$\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the kernel function and captures the similarity between the data-points. The clustering model in the dual space evaluated on training data becomes:

$$e^{(l)} = \Omega\alpha^{(l)} + b_l 1_N, l = 1, \ldots, k-1. \quad (5)$$

The eigenvectors $\alpha^{(l)}$ express an embedding of the input data that reveals the underlying clustering structure. They are linked to the $w^{(l)}$ through the first KKT condition.

In order to cope with truly non stationary data arriving over time, the initial $\alpha^{(l)}$ must be modified in response to the new inputs. This issue is tackled by means of the incremental kernel spectral clustering algorithm, which will be explained in detail in the next section.

The out-of-sample extension is performed by the ECOC decoding scheme. In the decoding process the cluster indicators found in the validation/test stage are compared with the code-book and the nearest code-word indicated by the Hamming distance is selected. The cluster indicators are the results of binarizing the score variables for test points:

$$\text{sign}(e^{(l)}_{\text{test}}) = \text{sign}(\Omega_{\text{test}}\alpha^{(l)} + b_l 1_{\text{Ntest}}) \quad (6)$$

with $l = 1, \ldots, k-1$. Ω_{test} is the $N_{\text{test}} \times N$ kernel matrix evaluated using the test points with entries $\Omega_{\text{test},ri} = K(x^{\text{test}}_r, x_i)$, $r = 1, \ldots, N_{\text{test}}$, $i = 1, \ldots, N$.

In the first two synthetic experiments that will be presented in section 4.1.1 (Drifting Gaussians and Merging Gaussians) we use the RBF kernel function defined by $K(x_i, x_j) = \exp(-||x_i - x_j||^2_2/\sigma^2)$. The symbol σ indicates the bandwidth parameter and x_i is the i-th data point. In the analysis of the third synthetic data (Synthetic time-series) and the PM_{10} data, x_i represents the i-th time-series. In this case to better capture the similarity between the time-series we use the RBF kernel with the correlation distance [21]. Thus $K(x_i, x_j) = \exp(-||x_i - x_j||^2_{\text{cd}}/\sigma^2)$, where $||x_i - x_j||_{\text{cd}} = \sqrt{\frac{1}{2}(1 - R_{ij})}$, with R_{ij} indicating the Pearson correlation coefficient between time-series x_i and x_j. By means of extensive experiments we empirically observed that this kernel is positive definite. Moreover the RBF kernel with Euclidean distance has been mathematically proven to fulfil the positive definitiveness property.

3. Incremental Kernel Spectral Clustering (IKSC)

3.1. Model-based update

In contrast with other techniques that compute approximate eigenvectors of large matrices like the Nyström method [32], the work presented in [14] or the above-mentioned algorithms [11] and [26], the eigen-approximation we use to evolve the initial model is model-based [4]. This means that based on a training set (in our case the cluster centroids) out-of-sample eigenvectors are calculated using eq. (7). These approximate eigenvectors are then used to adapt the initial clustering model over-time. In principle, if the training model has been properly constructed, this guarantees high accuracy of the approximated eigenvectors due to the good generalization ability of KSC and LS-SVMs in general [3, 30] (see also the discussion in Section 5.2).

3.2. The algorithm

One big advantage of a model-based clustering tool like KSC is that we can use it online in a straightforward way. Indeed, once we built-up our optimal model during the training phase, we

can estimate the cluster membership for every new test point by simply applying eq. (6) and the ECOC decoding procedure. However, if the data source is non-stationary, this scheme fails since the initial model is not representative any more of the new data distribution. Therefore to cope with non-stationary data the starting code-book must be adjusted accordingly. Here, instead of using the code-book and the ECOC procedure, we propose to express our model in terms of the centroids in the eigenspace and to compute the cluster memberships as measured by the euclidean distance from these centers. In this way it is possible to continuously update the model in response to the new data-stream. In order to calculate the projection in the eigenspace for every new point, we can exploit the second KKT condition for optimality which links the eigenvectors and the score variables for training data:

$$\alpha^{(l)}_{\text{test}} = \frac{1}{\lambda_l} D^{-1}_{\text{test}} e^{(l)}_{\text{test}} \tag{7}$$

with $D^{-1}_{\text{test}} = \text{diag}(1/\text{deg}(x^{\text{test}}_1), \ldots, 1/\text{deg}(x^{\text{test}}_{\text{Ntest}})) \in \mathbb{R}^{\text{Ntest}} \times \mathbb{R}^{\text{Ntest}}$ indicating the inverse degree matrix for test data. The out-of-sample eigenvectors $\alpha^{(l)}_{\text{test}}$ represent the model-based eigen-approximation with the same properties as the original eigenvectors $\alpha^{(l)}$ for training data. With the term eigen-approximation we mean that these eigenvectors are not the solution of an eigenvalue problem, but they are estimated by means of a model built during the training phase of KSC [4]. To summarize, once one or more new points belonging to a data-stream are collected, we update the IKSC model as follows:

- calculate the out-of-sample extension using eq.(6), where the training points x_i are the centroids in the input space $C_1, \ldots, C_k$, and the $\alpha^{(l)}$ are the centroids in the eigenspace $C^{\alpha}_1, \ldots, C^{\alpha}_k$
- calculate the out-of-sample eigenvectors by means of eq. (7)
- assign the new points to the closest centroids in the eigenspace
- update the centroids in the eigenspace
- update the centroids in the input space

To update online a centroid C_{old} given a new sample x_{new}, we can use the following formula [15]:

$$C_{\text{new}} = C_{\text{old}} + \frac{x_{\text{new}} - C_{\text{old}}}{n_{\text{old}}} \tag{8}$$

where n_{old} is the number of samples previously assigned to the cluster center C_{old}. The same procedure can be used to update the cluster centers in the eigenspace: in this way the initial $\alpha^{(l)}$ provided by KSC are changed over time to model the non-stationary behaviour of the system. A schematic visualization of this procedure is depicted in Fig. 2. Finally, here we sketch the complete IKSC algorithm:

Algorithm IKSC Incremental Kernel Spectral Clustering algorithm

---

Input: Training set $\mathcal{D} = \{x_i\}^N_{i=1}$ for the initialization stage, initial centroids in the input space $C_1, \ldots, C_k$ (training set online stage), initial centroids in the eigenspace $C^{\alpha}_1, \ldots, C^{\alpha}_k$ (initial clustering model), kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ positive definite and localized ($K(x_i, x_j) \rightarrow 0$

if x_i and x_j belong to different clusters), kernel parameters (if any), number of clusters k.
Output: Updated clusters $\{\mathcal{A}_1, \ldots, \mathcal{A}_p\}$, cluster centroids in the input space (training set online stage) $C_1, \ldots, C_k$, cluster centroids in the eigenspace (clustering model) $C_1^\alpha, \ldots, C_k^\alpha$.

Initialization:

1. Acquire N points.
2. Train the KSC model by solving eq. (4).
3. Obtain the initial centroids in the input space $C_1, \ldots, C_k$ and the initial centroids in the eigenspace $C_1^\alpha, \ldots, C_k^\alpha$.

Online IKSC:

for *i=N+1* to the end of the data-stream

1. compute the out-of-sample eigenvectors using eq. (7)
2. calculate cluster membership for the new point (or the new batch of points) according to the distance between the out-of-sample eigenvectors and the centroids $C_1^\alpha, \ldots, C_k^\alpha$
3. update centroids in the eigenspace $C_1^\alpha, \ldots, C_k^\alpha$ using eq. (8)
4. update centroids in the input space $C_1, \ldots, C_k$ according to eq. (8)
5. new cluster check
6. merge check
7. cluster death

endfor

Outlier elimination.

The adaptation to non-stationarities relates to identifying changes in the number of clusters occurring over time by means of some inspections:

- the new cluster check allows to dynamically create a new cluster if necessary. For every new point the related degree d_i^{test} is calculated. If $d_i^{\text{test}} < \epsilon$ where ϵ is a user-defined threshold, it means that the point is dissimilar to the actual centroids. Therefore it becomes the centroid of a new cluster and it is added to the model. Moreover, the old eigenspace is updated in the following way. If at time t a new cluster is created, the number of cluster centers increases from k_{old} to k_{new}. Then a kernel matrix involving only the centroids of dimension $k_{\text{new}} \times k_{\text{new}}$ is created and problem (4) with $k = k_{\text{new}}$ is solved. In this way the cluster prototypes are now represented in a k_{new} dimensional eigenspace, and the same applies for the next points of the data stream.

- throughout the merge check, if two centroids become too similar they are merged into one center, and the number of clusters is decreased. In this case the dimension of the eigenspace is not changed.

- if the centroid of a cluster is not updated any more the algorithm considers that cluster as disappeared (cluster death).

Finally, if one cluster is formed by less than 5 points it is considered as outlier and it is eliminated in the end of the data-stream acquisition.

3.3. Computational complexity

In the initialization stage we have to solve the eigenvalue problem (4) involving an $N \times N$ matrix, which has quadratic complexity if we use fast solvers like the Lanczos algorithm [6]. Then before the data-stream acquisition we compute the k initial centroids in the input space and the corresponding centroids in the eigenspace. During the online stage involving the data-stream processing, we consider as training set only the k centers in the input space $C_1, \ldots, C_k$, while the k centers in the eigenspace $C_1^\alpha, \ldots, C_k^\alpha$ represent the clustering model. For every new point of the data-stream, as explained in the previous section, we have to compute the out-of-sample extension, the corresponding out-of-sample eigenvectors by means of eq. (7) and the update of both the model and the training set[2]. In this case the main contribution to the computational complexity is due to the out-of-sample extension part:

$$e_{\text{test}}^{(l)} = \Omega_{\text{test}} \alpha^{(l)} + b_l 1_{\text{Ntest}}, l = 1, \ldots, k-1. \tag{9}$$

The evaluation of the kernel matrix Ω_{test} needs $O(k^2 d)$ operations to be performed. The calculation of the score variables $e_{\text{test}}^{(l)}$ takes then $O(k^2 d + k^2 + k)$ time. This operation has to be repeated for the N_{test} data-points of the data-stream, so the overall time complexity is $O(N_{\text{test}}(k^2 d + k^2 + k))$. This can become linear with respect to the number of data-points ($O(N_{\text{test}})$) when $k \ll N_{\text{test}}$ and $d \ll N_{\text{test}}$, which is the case in many applications. This is comparable with other eigen-updating algorithms for spectral clustering like [26] and [11].

4. Data-sets

4.1. Artificial data

Three simulations are performed: the first and the second by reproducing the experiments described in [5], and the third with some computer-generated time-series.

4.1.1. Gaussian clouds

In the first simulation two Gaussian distributions evolving over time are created. These two clouds of points drift toward each other with increasing dispersal, as illustrated in Fig. 3. In the second virtual experiment a multi-cluster non-stationary environment is created. In particular, there are two drifting Gaussian clouds that come to merge, some isolated data forming an outlier cluster of 4 points and a static cluster consisting of a bi-modal distribution. This second data-set is depicted in Fig. 4.

4.1.2. Synthetic time-series

In order to test the ability of IKSC to dynamically cluster time-series rather than data-points, we generated 20 time-series of three types as depicted in Fig. 5. The idea behind this experiment is that if we cluster in an online fashion the time-series with a moving window approach, we should be able to detect the appearance of a new cluster given the increase in frequency of the signals of

[2] In this paper we assume that the training set during the online stage consists of k points, where k is the number of clusters. In some situations it could happen that such a small number of training points is not enough to define a proper mapping. Nevertheless, by considering more training points N such that $N \ll N_{\text{test}}$ the overall complexity of the algorithm does not change.

the second type at time step $t_1 = 150$. Moreover, when these signals get back to their original frequency at time step $t_2 = 300$, the clustering algorithm must detect this change.

4.2. The PM_{10} data-set

Particulate Matter (PM) is the term used for solid or liquid particles found in the air. In particular PM_{10} refers to those particles whose size is up to 10 micrometers in aerodynamic diameter. The inhalation of these particles is dangerous for human health since it can cause asthma, lung cancer, cardiovascular issues, etc. Accurate measurements and estimation of PM is then of vital importance by the health care point of view. To this aim the European Environmental Agency manages a publicly available database called AirBase [12]. This air-quality database contains validated air quality monitoring information of several pollutants for more than 30 participating countries throughout Europe.

In this paper we analyze the PM_{10} data registered by 259 background stations during a heavy pollution episode that took place between January 20th, 2010 and February 1st, 2010. We focus on an area comprising four countries: Belgium, Netherlands, Germany and Luxembourg (see Fig.6). The experts attributed this episode to the import of PM originating in Eastern Europe [1].

5. Experimental results

In this section we show how the proposed IKSC model, thanks to its capacity of adapting to a changing environment, is able to model the complex behaviour of evolving patterns of non-stationary data.

To evaluate the outcomes of the model, two cluster quality measures are computed [13]: the average cumulative adjusted rand index (ARI) error and the instantaneous silhouette criterion. The ARI is an external evaluation criterion and measures the agreement between two partitions (ARI $= 0$ means complete disagreement and ARI $= 1$ indicates a perfect match). The ARI error is defined then as 1 -ARI, as in [11]. Silhouette is an internal criterion taking a value in the range $[-1, 1]$ and measures how tightly grouped all the data in the clusters are.

5.1. Artificial data

The results of testing the IKSC algorithm on the first synthetic example is presented in Fig. 7. In the initialization phase 30 points are used to construct the model. The IKSC algorithm can perfectly model the two drifting distributions: the average cumulative ARI error is equal to 0. Moreover the quality of the predicted clusters remains very high over time, as demonstrated by the trend of the average silhouette index depicted in Fig. 8. The results of the simulation related to the second artificial data-set are depicted in Fig. 9. Similarly to the first artificial experiment, the cluster quality stays high over time as shown in Fig. 10, and the partitions found by IKSC are in almost perfect agreement with the ground truth (small ARI error) for the whole duration of the simulation (see Fig.11). Moreover at time-step $t = 6926$ the two moving Gaussian clouds are merged, as expected. Only in this case, as observed also in [5], there is a small increase in the average cumulative ARI error. The small cluster at the bottom left side of Fig. 4 is detected as outlier after the data acquisition.

Finally, we discuss the results of IKSC on the synthetic time-series experiment. In the initialization phase the algorithm recognize 2 clusters, which are shown in Fig. 12. After some time, we notice that IKSC successfully detects the first change in frequency of the signals of the second type

(see Section 4.1.2) by creating a new cluster at time step $t = 223$, as depicted in Fig.13. Moreover the second change point is detected at time step $t = 382$, when a merging of two clusters is performed, as illustrated in Fig.14. A video of this simulation is also present in the supplementary material of the paper.

5.2. The approximated model-based eigenvectors

Here we discuss on the quality of our model-based eigen-updating for kernel spectral clustering. In Fig. 15 the exact and the approximated eigenvector related to the largest eigenvalue of (4) for the drifting Gaussians example are shown. We notice that the model-based eigenvectors are less noisy with respect to the exact eigenvectors and a multiplicative bias is present. The first property is quite surprising: basically we are able to recover the perfect separation between the two clusters even when this is somehow masked by the data. This occurs mainly in the end of the simulation when the two Gaussian clouds approach each other. In this case the exact eigenvector is not exactly piecewise constant due to a small overlap, while the model-based eigenvector is much less perturbed. The multiplicative bias is probably due to the fact that the out-of-sample eigenvectors are computed using an ultra-sparse training set (only the two cluster centroids). The latter allows to process the data-stream very quickly, but lacks of the information related to the spread of the data-points, which may cause the bias. Similar considerations can be done for the second synthetic experiment, i.e. the merging Gaussians. The three eigenvectors corresponding to the largest eigenvalues of (4) are represented in Fig. 16. In the third approximated eigenvector we can notice 4 levels, which are not present in the exact eigenvector. Once again this testifies the tight relation between the clustering model of IKSC (the 4 centroids) and the approximated eigenvectors, which is a unique property of our framework.

5.3. PM_{10} data

In the initialization phase our data-set consists of a time-series of 96 time steps (i.e. four days) for each station. In order to build-up an initial clustering model we tune the number of clusters k and the proper σ for the RBF kernel by using the AMS (Average Membership Strength) model selection criterion [19]. In the cited work a method to obtain soft cluster memberships from KSC has been introduced. Based on this soft assignment technique a new model selection method has been derived. It works by computing a kind of mean membership per cluster indicating the average degree of belonging of the points to that cluster. By repeating the same procedure for every cluster and taking the mean, we obtain the AMS criterion. Unlike previously proposed model selection criteria as BLF [3] and Modularity [17], AMS works fine with overlapping clusters and can be used for large scale data analysis.

After tuning we find $k = 2$ and $\sigma^2 = 0.05$ as optimal parameters, as depicted in Fig. 17. The initial model, based on these parameters, is illustrated in Fig. 18. In this case the 2 centroids in the input space are the time-series representing the two clusters, while in the eigenspace they are points of dimension $k - 1$ (anyway for visualization purposes we always use a 3D plot).

During the online stage, by adopting a moving window approach, our data-set at time t corresponds to the PM_{10} concentrations measured from time $t - 96$ to time t. In this way we are able to track the evolution of the pollutants over-time. In fact, after some time the IKSC model creates a new cluster, as depicted in Fig. 19. Later on these three clusters evolve until a merge of two of them occurs at time step $t = 251$ (see Fig. 20). If we analyse more in details the clustering results (see video in the supplementary material), we can notice how the new cluster (represented

in blue) is concentrated mainly in the Northern region of Germany. Moreover the creation occurs at time step $t = 143$, when the window describes the start of the pollution episode in Germany (see Section 4.2). Afterwards, the new cluster starts expanding in direction South-West. Basically, IKSC is detecting the arrival of the pollution episode originated in Eastern Europe and driven by the wind toward the West. This ability of our clustering model of detecting the dynamics of the pollution cloud at this level of accuracy is rather unexpected. Indeed, IKSC does not have any information about the spatial localization of the stations and the meteorological conditions. At time step $t = 251$ two clusters are merged. This can be explained by the fact that the window covers the unusually high PM_{10} concentrations as well as the end of the episode, registered by many of the stations.

5.4. Comparison with Incremental K-means (IKM)

One of the most popular data clustering methods in many scientific domains is K-means clustering because of its simplicity and computational efficiency. K-means clustering works by choosing some random initial centers and then iteratively moves the centers to minimize the total within cluster variance. In its incremental variant, the K-means clustering algorithm is applied online to a data stream. At each time-step Incremental K-means (IKM) uses the previous centroids to find the new cluster centers, instead of rerunning the K-means algorithm from scratch [8].

In Table 1 a summary of the results regarding all the experiments is presented. The performance of IKSC and IKM are compared in terms of mean ARI and mean Silhouette index over time. Concerning the experiments with the Gaussian clouds IKSC achieves better cluster accuracy (higher ARI), with a slightly worse Silhouette value with respect to IKM. In the case of the synthetic time-series and the PM_{10} data IKSC outperforms IKM in terms of the Silhouette index.

6. Conclusions

In this work an adaptive clustering model called Incremental Kernel Spectral Clustering (IKSC) has been introduced. IKSC takes advantage of the out-of-sample property of kernel spectral clustering (KSC) to adjust the initial model over time. Thus, in contrast with other existing incremental spectral clustering techniques, we propose a model-based eigen-update, which guarantees high accuracy. On some toy-data we have shown the effectiveness of IKSC in modelling the cluster evolution over-time (drifting, merging, outlier elimination etc.). Then we analysed a real-world data-set consisting of PM_{10} concentrations registered during a heavy pollution episode that took place in Northern Europe in January 2010. Also in this case IKSC was able to recognize some interesting patterns and track their evolution over-time, in spite of dealing with the complex dynamics of PM_{10} concentration.

Acknowledgements

This work was supported by Research Council KUL: ERC AdG. A-DATADRIVE-B, GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine), research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC), IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization,

2007-2011); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); Contract Research: AMINAL; Other:Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger. Oscar M. Agudelo is a post-doctoral fellow at the KU Leuven, Belgium. Bart De Moor is Full Professor at the KU Leuven, Belgium. Johan Suykens is Professor at the KU Leuven, Belgium. The scientific responsibility is assumed by its authors.

References

[1] Agudelo, O. M., Viaene, P., Blyth, L., De Moor, B., 2012. Application of data assimilation techniques to the air quality model AURORA. Internal Report 12-134, ESAT-SISTA, KU Leuven (Leuven, Belgium).

[2] Alzate, C., Suykens, J. A. K., 2010. Highly sparse kernel spectral clustering with predictive out-of-sample extensions. In: Proc. of the 18th European Symposium on Artificial Neural Networks (ESANN 2010). pp. 235–240.

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

[4] Alzate, C., Suykens, J. A. K., 2011. Out-of-sample eigenvectors in kernel spectral clustering. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2011). pp. 2349–2356.

[5] Boubacar, H. A., Lecoeuche, S., Maouche, S., 2008. Sakm: Self-adaptive kernel machine a kernel-based algorithm for online clustering. Neural Networks 21 (9), 1287 – 1301.

[6] C., L., 1950. Iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of Standards.

[7] Chakrabarti, D., Kumar, R., Tomkins, A., 2006. Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '06. ACM, New York, NY, USA, pp. 554–560.

[8] Chakraborty, S., Nagwani, N., 2011. Analysis and study of incremental k-means clustering algorithm. In: High Performance Architecture and Grid Computing. Vol. 169 of Communications in Computer and Information Science. pp. 338–341.

[9] Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B. L., 2007. Evolutionary spectral clustering by incorporating temporal smoothness. In: KDD. pp. 153–162.

[10] Chung, F. R. K., 1997. Spectral Graph Theory. American Mathematical Society.

[11] Dhanjal, C., Gaudel, R., Clemenccon, S., 2013. Efficient eigen-updating for spectral graph clustering. arXiv/1301.1318.

[12] Eionet, 2011. European topic centre on air and climate change. [online] `http://air-climate.eionet.europa.eu/databases/airbase`.

[13] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. Journal of Intelligent Information Systems 17, 107–145.

[14] Hoegaerts, L., De Lathauwer, L., Goethals, I., Suykens, J. A. K., Vandewalle, J., De Moor, B., 2007. Efficiently updating and tracking the dominant kernel principal components. Neural Networks 20 (2), 220–229.

[15] Knuth, D. E., 1997. The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[16] Langone, R., Alzate, C., De Ketelaere, B., Suykens, J. A. K., 2013. Kernel spectral clustering for predicting maintenance of industrial machines. In: IEEE Symposium Series on Computational Intelligence (SSCI) 2013. pp. 39–45.

[17] Langone, R., Alzate, C., Suykens, J. A. K., 2011. Modularity-based model selection for kernel spectral clustering. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2011). pp. 1849–1856.

[18] Langone, R., Alzate, C., Suykens, J. A. K., 2013. Kernel spectral clustering with memory effect. Physica A: Statistical Mechanics and its Applications 392 (10), 2588 – 2606.

[19] Langone, R., Mall, R., Suykens, J. A. K., 2013. Soft kernel spectral clustering. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2013). pp. 1028 – 1035.

[20] Langone, R., Suykens, J. A. K., 2013. Community detection using kernel spectral clustering with memory. Journal of Physics: Conference Series 410 (1), 012100.

[21] Liao, T. W., 2005. Clustering of time series data - a survey. Pattern Recognition 38 (11), 1857 – 1874.

[22] Mall, R., Langone, R., Suykens, J., 2013. Highly Sparse Reductions to Kernel Spectral Clustering. In: 5th International Conference on Pattern Recognition and Machine Intelligence (PREMI).

[23] Mall, R., Langone, R., Suykens, J. A. K., 2013. Kernel spectral clustering for big data networks. Entropy 15 (5), 1567–1586.

[24] Mall, R., Langone, R., Suykens, J. A. K., 2013. Self-Tuned Kernel Spectral Clustering for Large Scale Networks. In: IEEE International Conference on Big Data (2013).

[25] Ng, A. Y., Jordan, M. I., Weiss, Y., 2002. On spectral clustering: Analysis and an algorithm. In: Dietterich, T. G., Becker, S., Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems 14. MIT Press, Cambridge, MA, pp. 849–856.

[26] Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T. S., Jan. 2010. Incremental spectral clustering by efficiently updating the eigen-system. Pattern Recogn. 43 (1), 113–127.

[27] Peluffo, D., Garcia, S., Langone, R., Suykens, J., Castellanos, G., 2013. Kernel spectral clustering for dynamic data using multiple kernel learning. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2013). pp. 1085 – 1090.

[28] Sarlin, P., 2013. Self-organizing time map: An abstraction of temporal multivariate patterns. Neurocomputing 99, 496–508.

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

[30] Van Gestel, T., Suykens, J. A., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., de Moor, B., Vandewalle, J., 2004. Benchmarking least squares support vector machine classifiers. Machine Learning 54 (1), 5–32.

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

[32] Williams, C. K. I., Seeger, M., 2001. Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13. MIT Press.

Experiment	Algorithm	Silhouette	ARI
Drifting Gaussians	**IKM**	**0.89**	1
	IKSC	0.88	1
Merging Gaussians	**IKM**	**0.91**	0.95
	IKSC	0.90	**0.99**
Synthetic time-series	**IKM**	0.90	–
	IKSC	**0.92**	–
PM_{10} data	**IKM**	0.27	–
	IKSC	**0.32**	–

Table 1: **Cluster quality evaluation.**. Average ARI and/or mean Silhouette index over time for all the experiments described in this paper. In the case of the synthetic time-series and the PM_{10} only Silhouette is computed since the true partition is unknown.

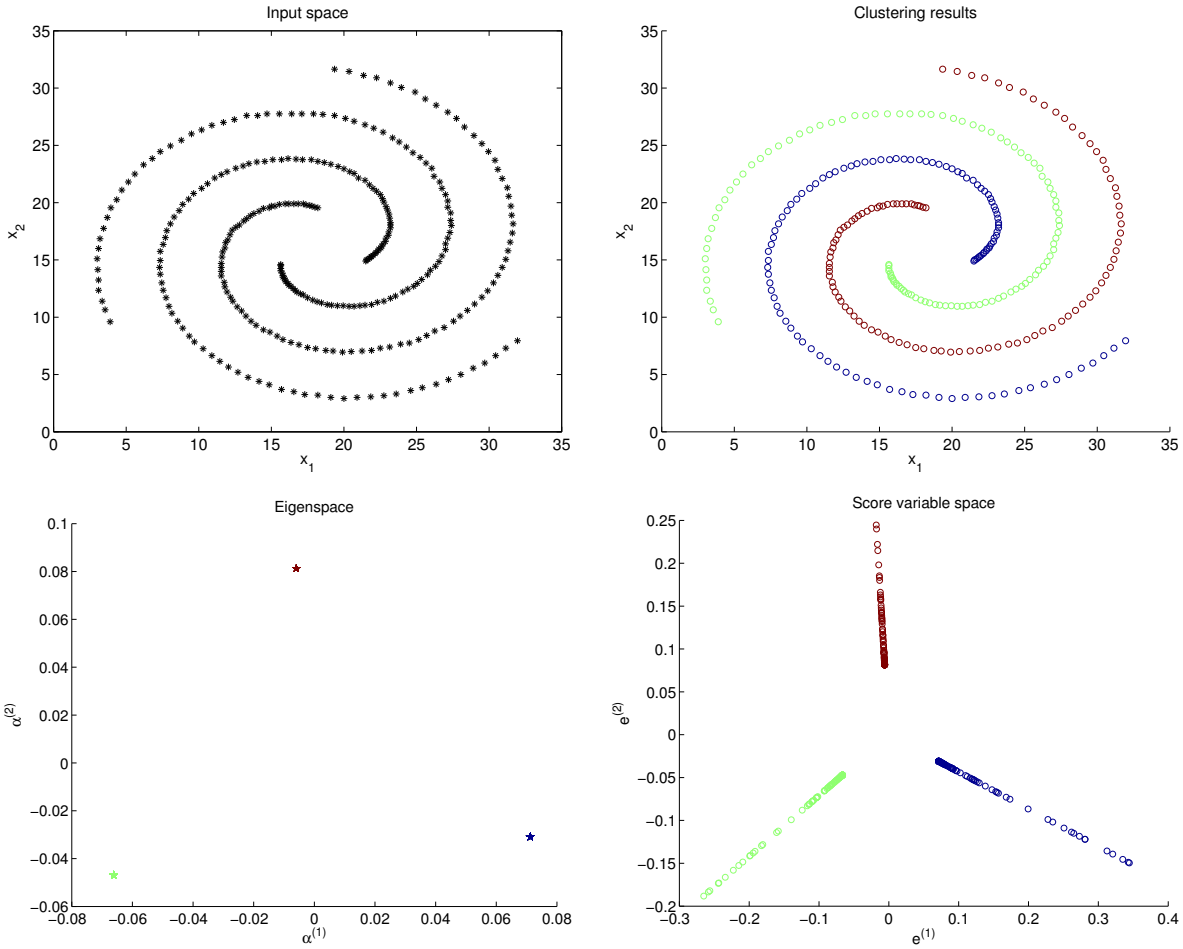


Figure 1: **Schematic illustration of KSC main variable spaces for the 2D three spiral dataset.** The original data $\mathcal{D} = \{x_i\}_{i=1}^N$ are mapped into a high dimensional Reproducing Kernel Hilbert Space (RKHS) by means of the feature map $\varphi(\cdot)$. In the feature space a linear model succeeds in separating the clusters, resulting in a non-linear clustering boundary in the input space. **Top left**: original data. **Top right**: clustering results. **Bottom left**: eigenspace. **Bottom right**: projection space.

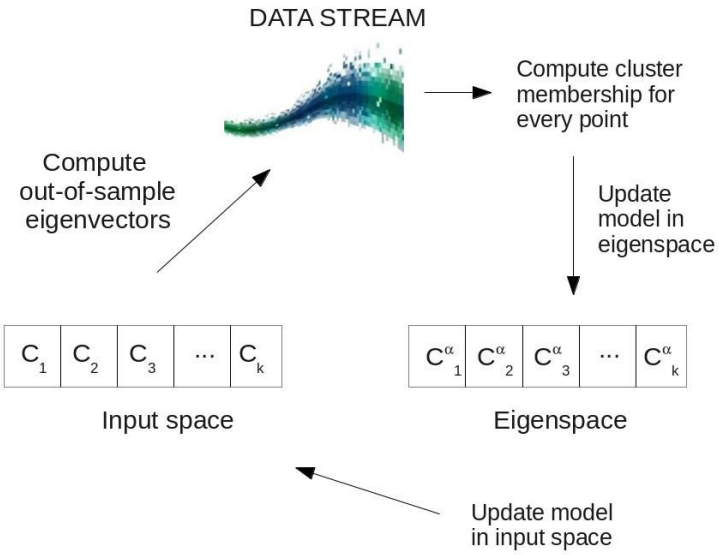


Figure 2: **IKSC update scheme.**

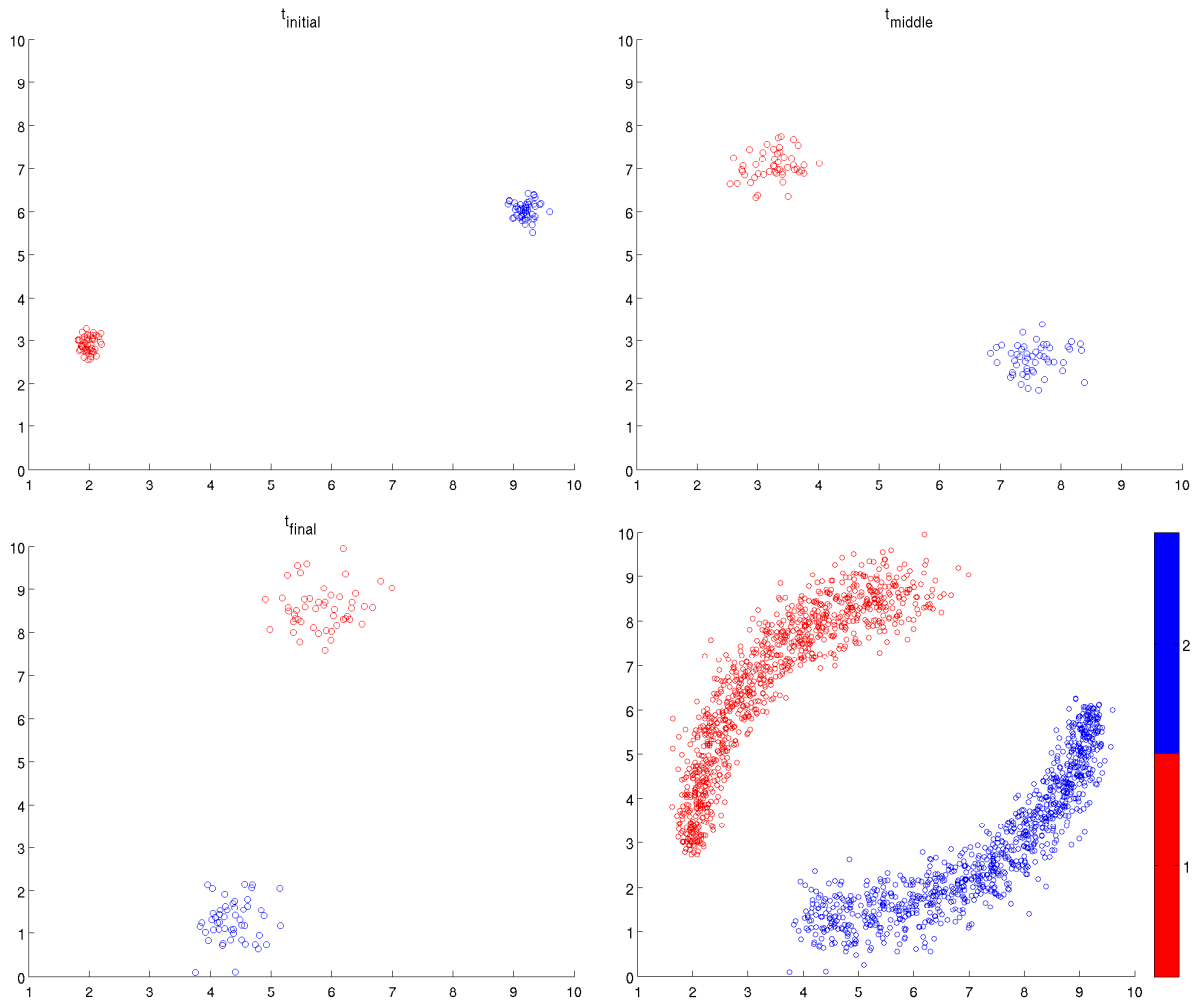


Figure 3: **Drifting Gaussian distributions.** Some snapshots of the evolution of the distributions (top and bottom left), and the whole data all at once (bottom right).

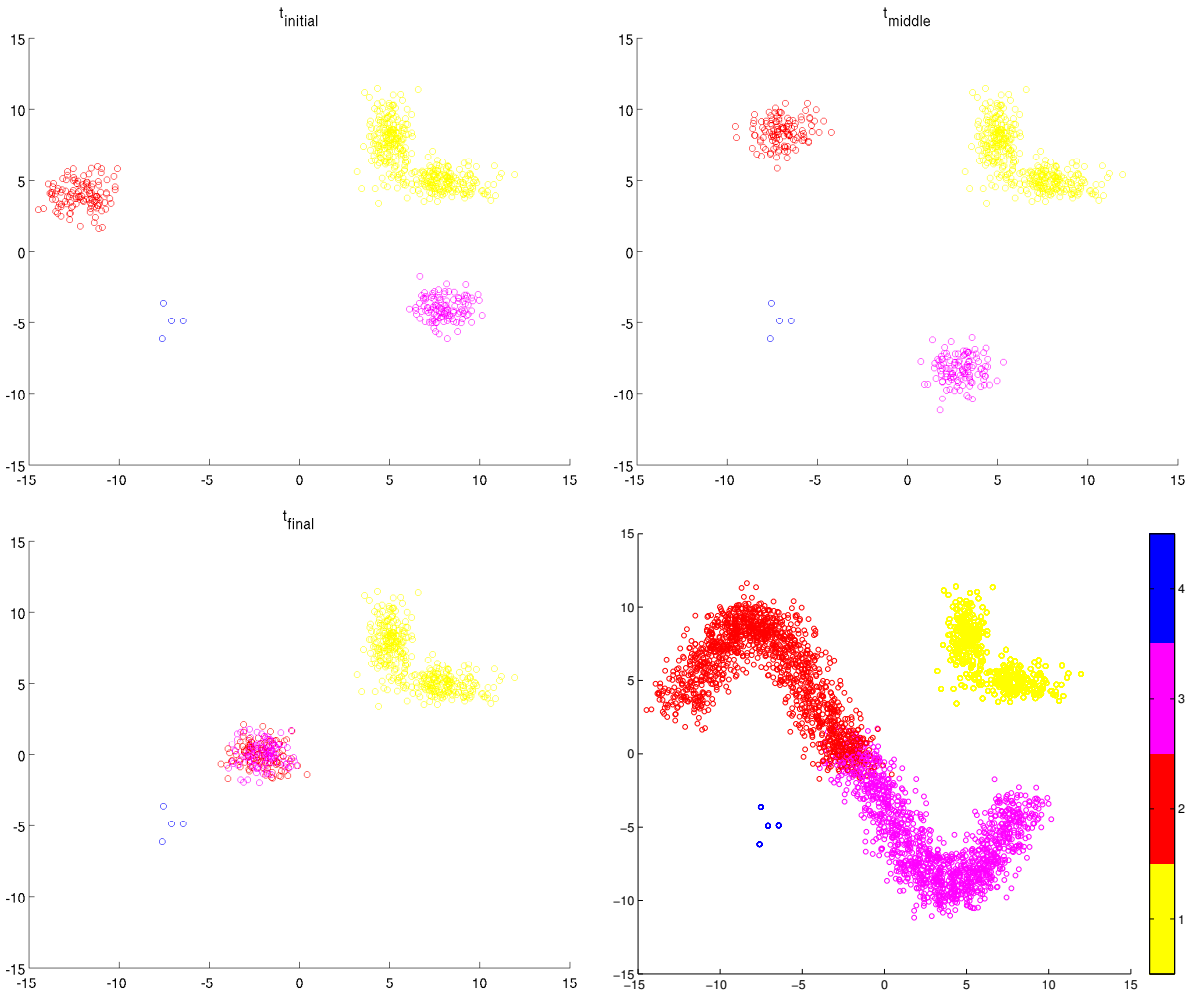


Figure 4: **Merging Gaussian distributions.** Some snapshots of the evolution of the distributions (top and bottom left), and the whole data all at once (bottom right).

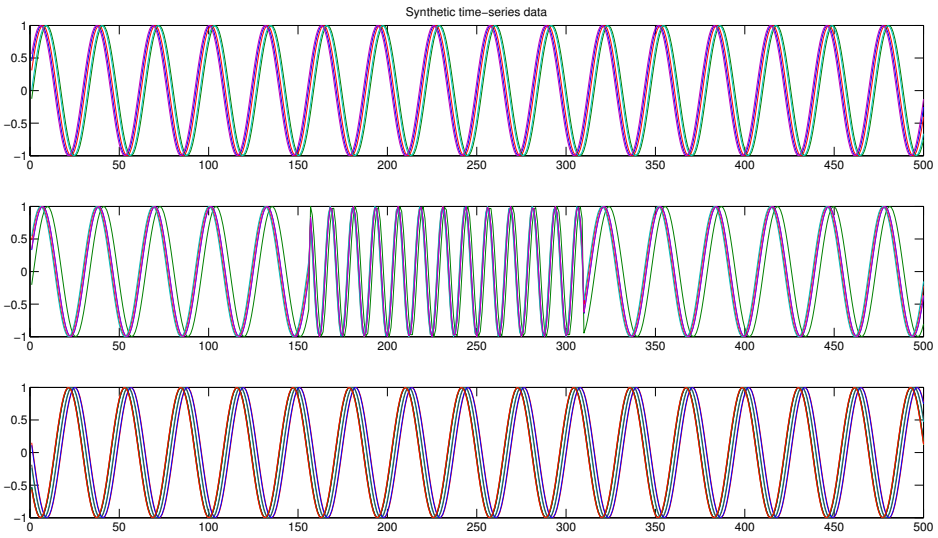


Figure 5: **Synthetic time-series.** At $t_1 = 150$ and $t_2 = 300$ two change points (change in frequency) can be observed.

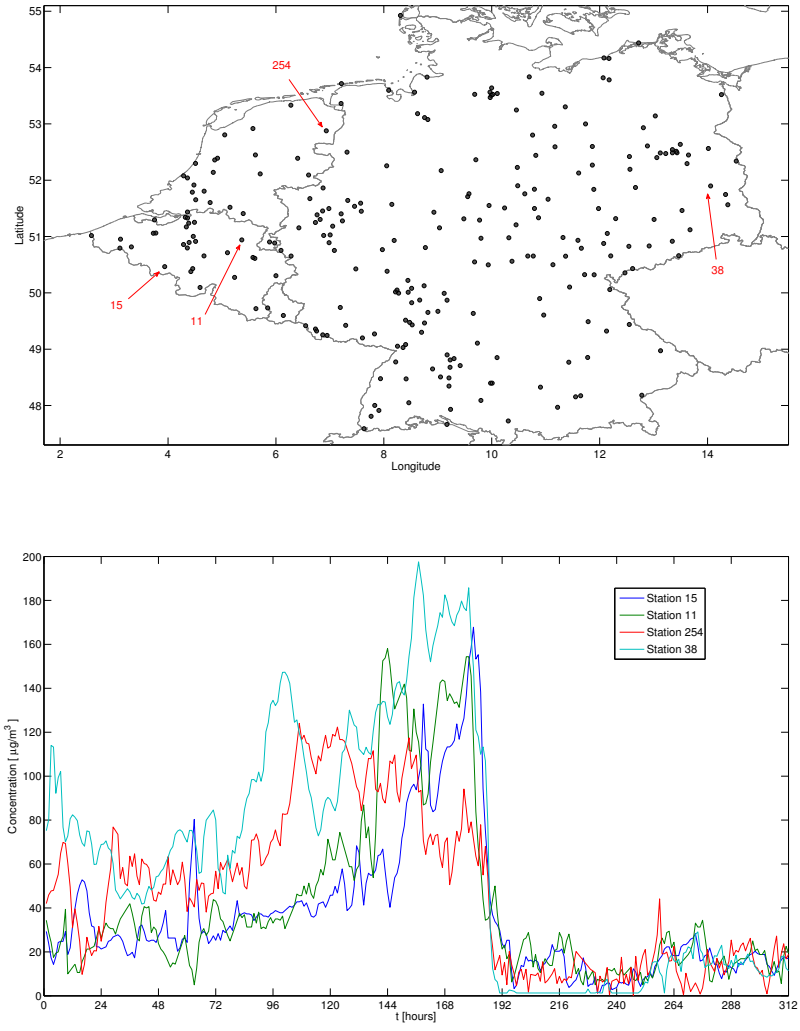


Figure 6: $\mathbf{PM_{10}}$ **data. Top:** AirBase monitoring stations. **Bottom:** Some representative time-series of PM_{10} concentrations for the whole period under investigation.

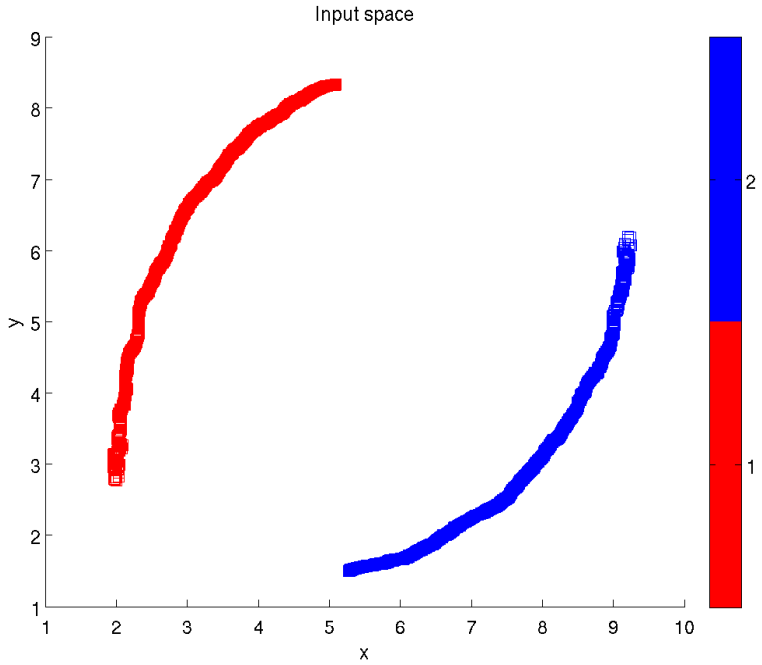


Figure 7: **Results of IKSC on the drifting Gaussian distributions.** Evolution of the centroids in the input space. We can notice that the IKSC model can recognize the drifting targets without errors. A video of the simulation is present in the supplementary material of this paper.

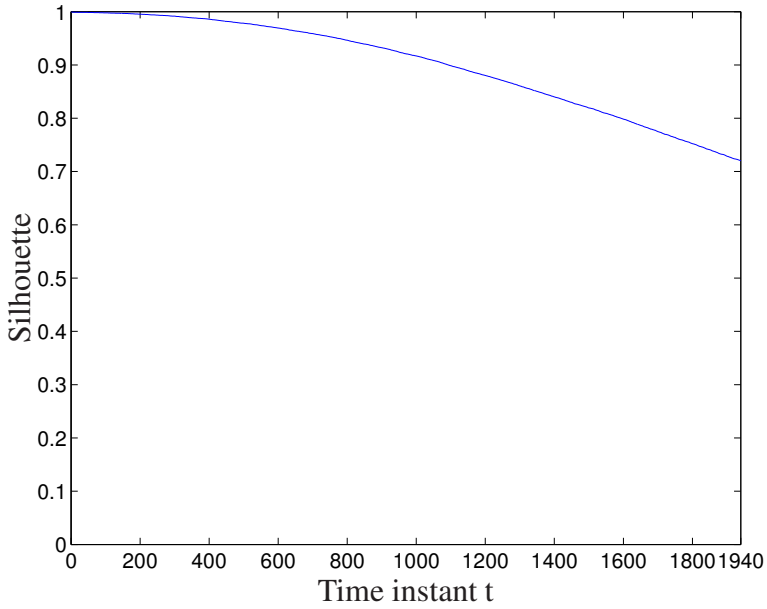


Figure 8: **Silhouette for drifting Gaussian distributions.** The mean silhouette value related to the clusters detected by IKSC stays high over time, meaning that our method is able to model the drift of the distributions.

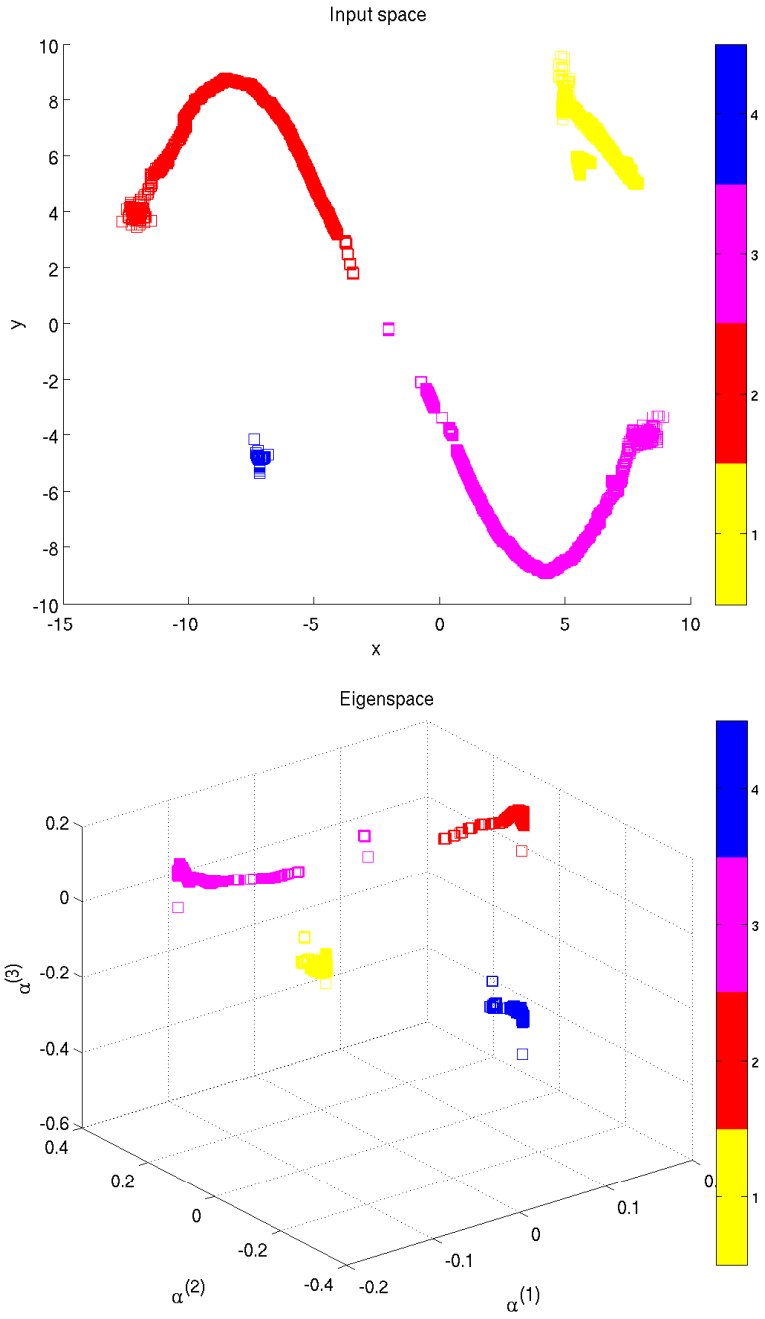


Figure 9: **Results of IKSC on the merging Gaussian distributions. Top:** Evolution of the centroids in the input space. **Bottom:** Model evolution in the eigenspace. A video of the simulation is provided as supplementary material of this paper.

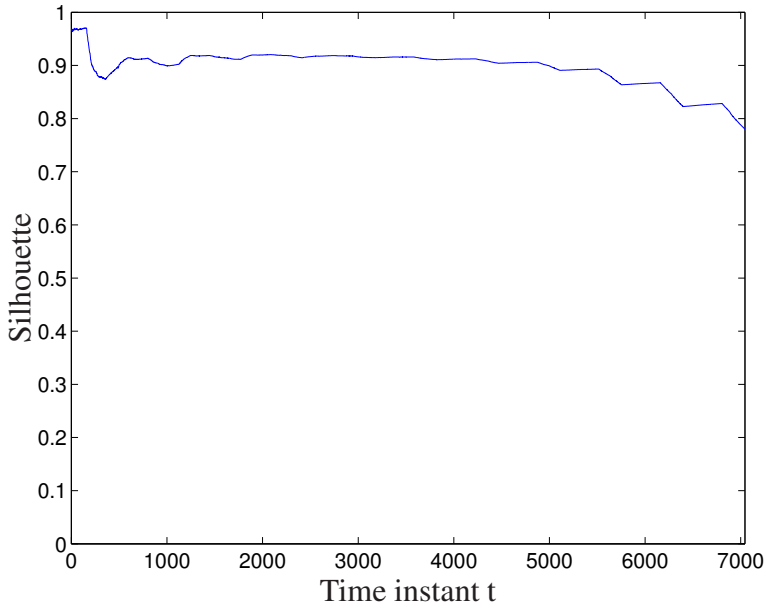


Figure 10: **Silhouette for the merging Gaussian distributions experiment.** The silhouette value related to the clusters detected by IKSC remains high over time. Thus, also in this case IKSC manages to properly follow the non-stationary behaviour of the clusters for the whole duration of the experiment.

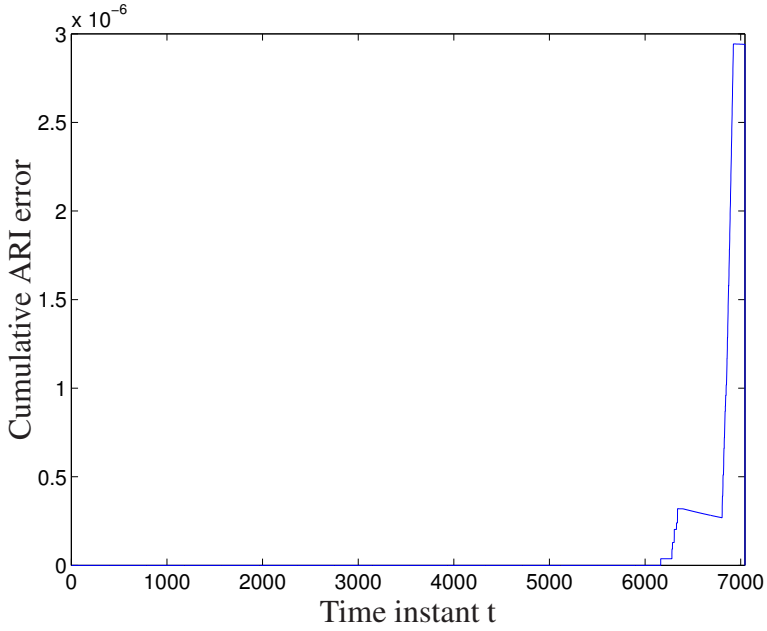


Figure 11: **ARI error-Merging Gaussian distributions.** The average cumulative ARI error related to the clusters detected by IKSC is very small over time, with a peak around the merging step at time $t = 6926$, in agreement with what was observed also in [5].

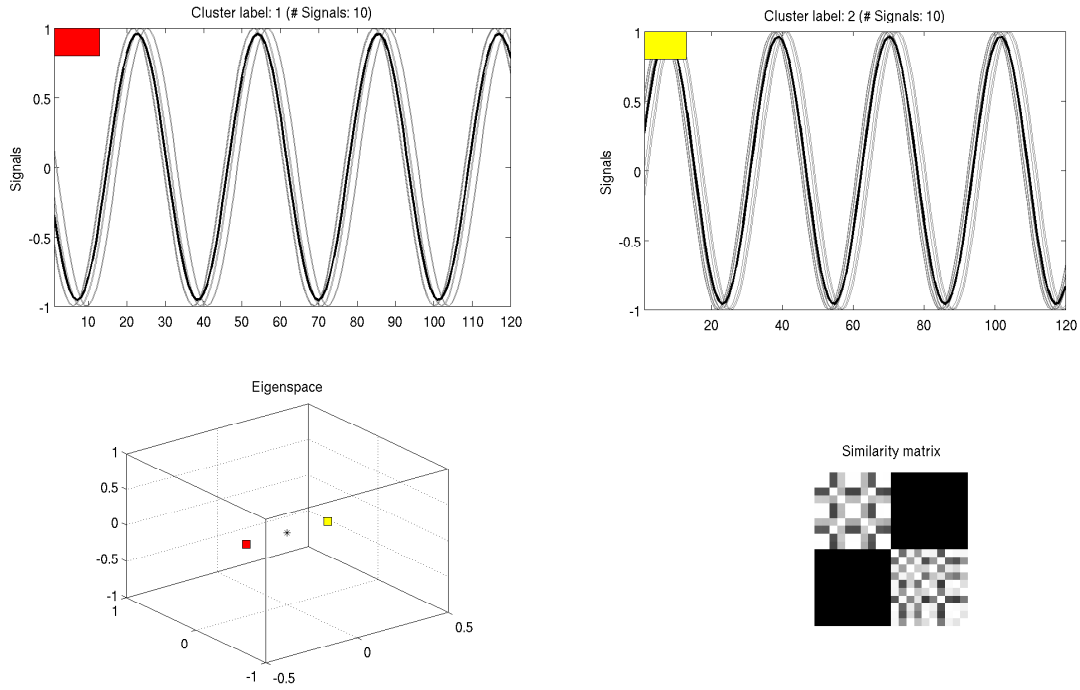


Figure 12: **Synthetic time-series initial clustering model. Top:** signals of the two starting clusters. **Bottom left**: data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right**: kernel matrix with a clear block diagonal structure.

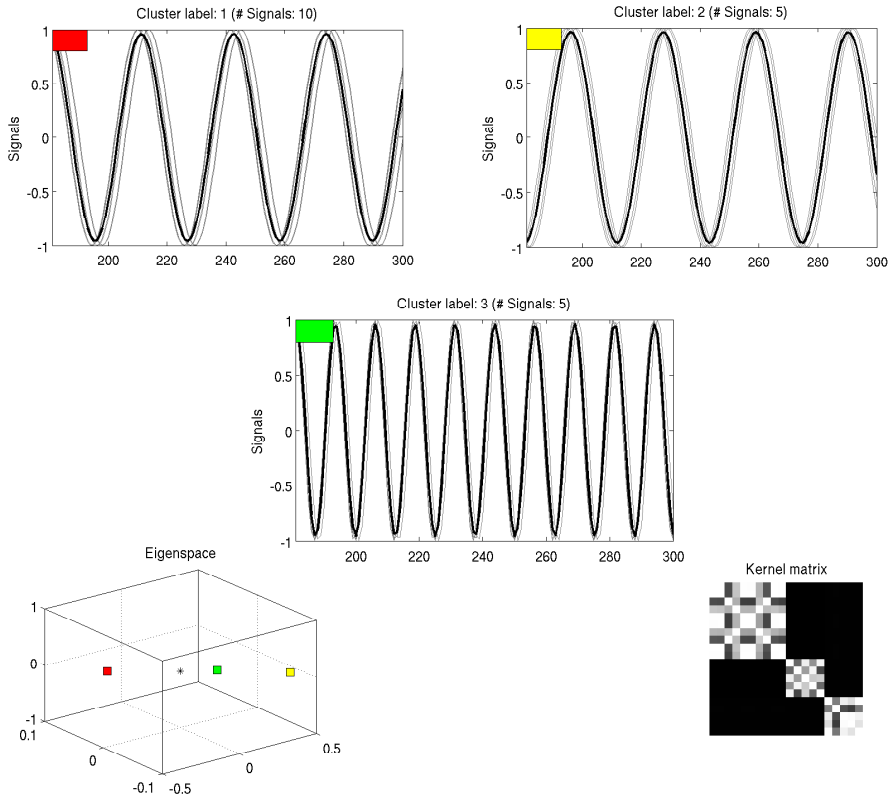


Figure 13: **Synthetic time-series clusters after creation. Top and center:** signals of the three clusters after the creation event. **Bottom left** data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right:** kernel matrix. A video of the entire simulation is present in the supplementary material of the paper.

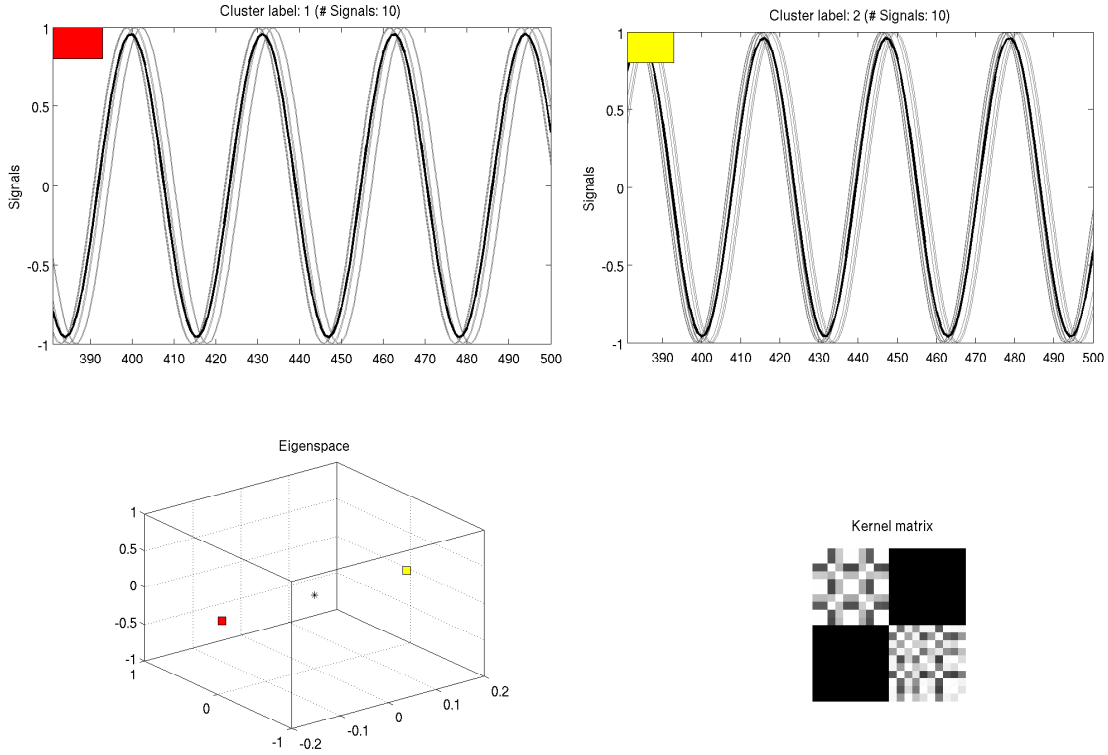


Figure 14: **Synthetic time-series final clustering model. Top:** two final clusters after the merging event. **Bottom left:** clustered data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right:** kernel matrix.

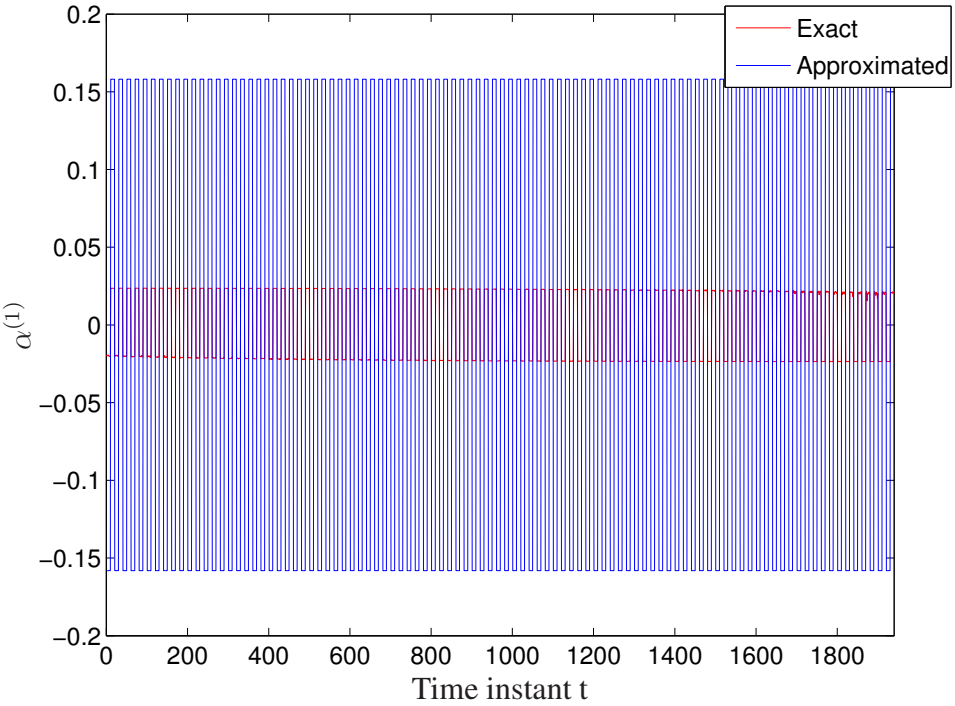


Figure 15: **Eigenvector-Drifting Gaussian distributions.** Exact and approximated eigenvector corresponding to the largest eigenvalue of the problem (4), for the first synthetic example.

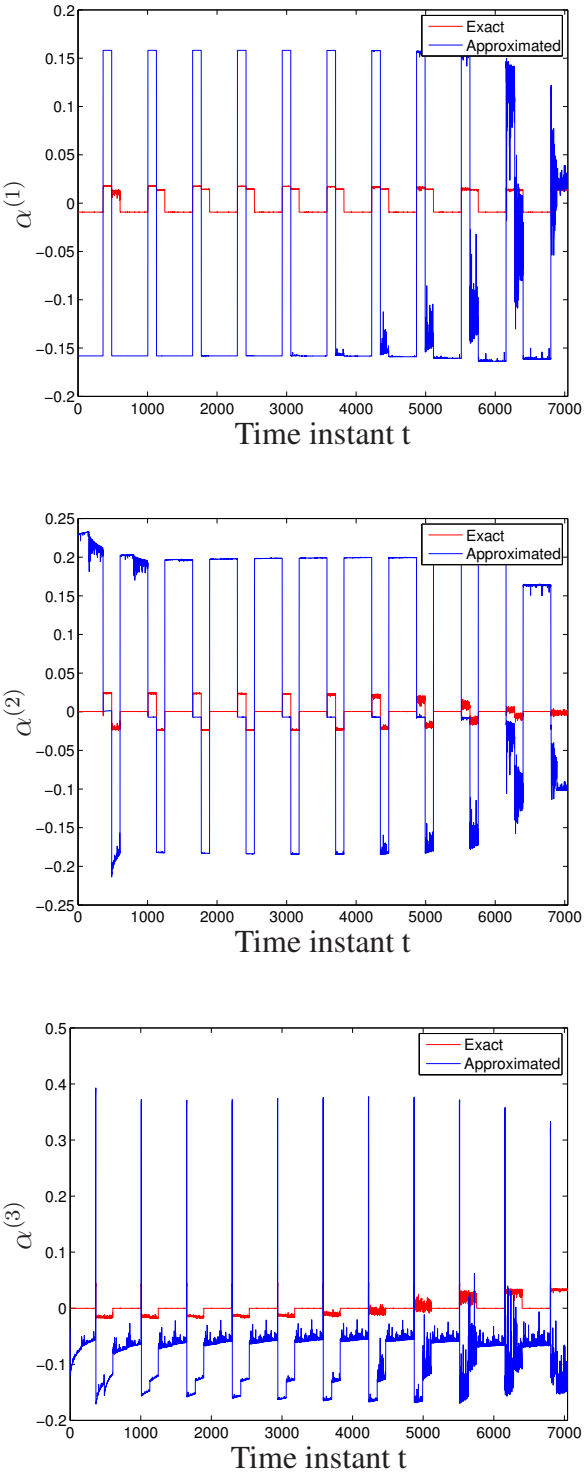


Figure 16: **Eigenvectors-Merging Gaussian distributions.** Exact and approximated eigenvectors corresponding to the 3 largest eigenvalues of the problem (4), for the second synthetic experiment.

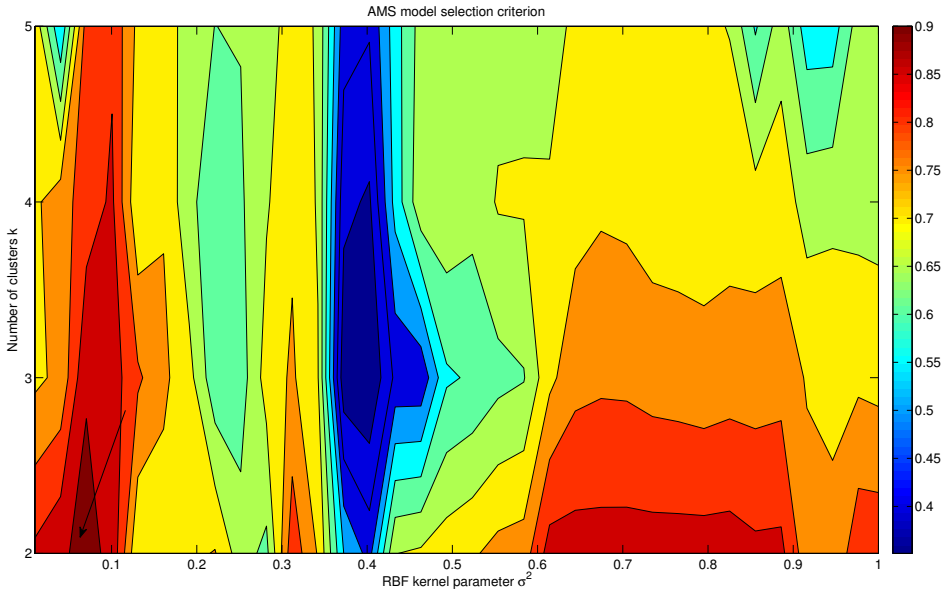


Figure 17: **Model selection.** Tuning of the number of clusters and the bandwidth of the RBF kernel in the initialization phase of IKSC for the analysis of the PM_{10} data.

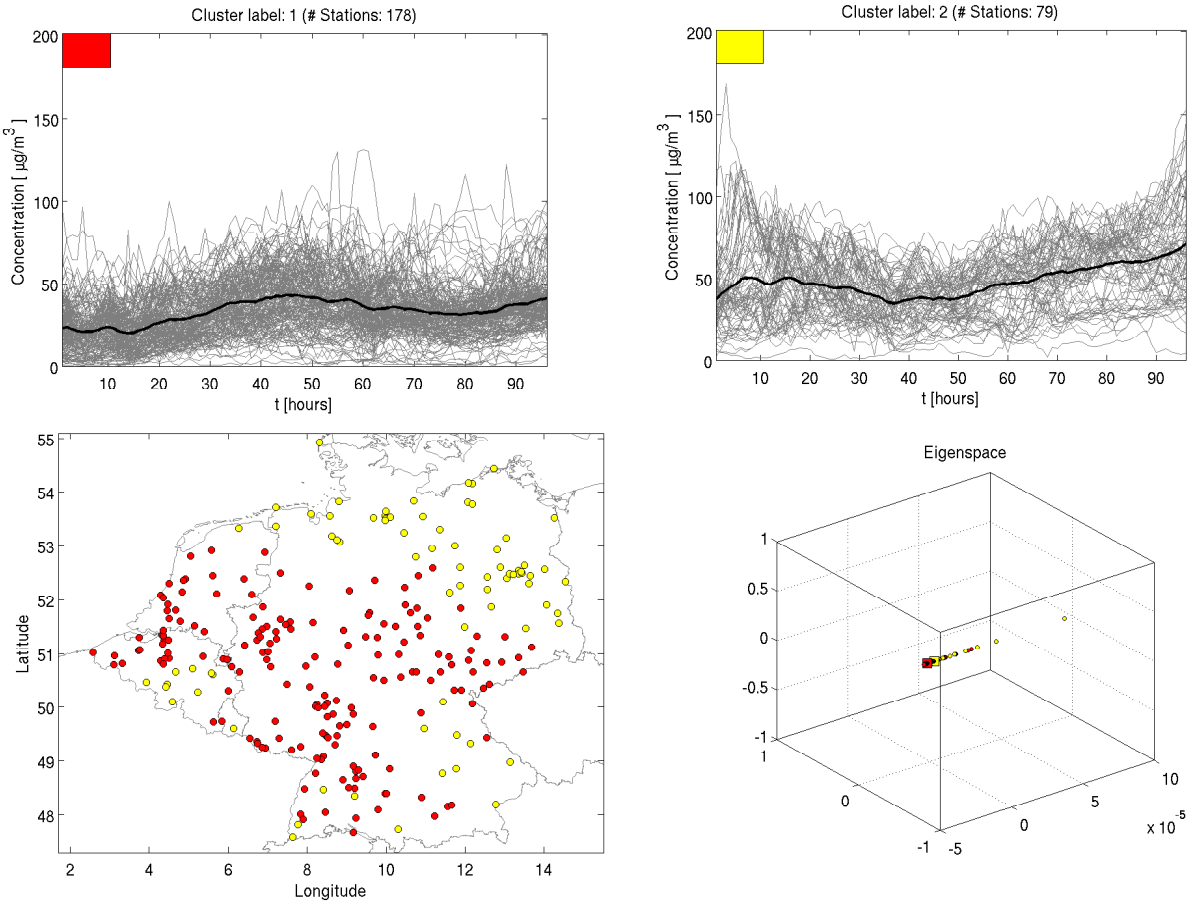


Figure 18: **Initial clustering model for the PM_{10} monitoring stations. Top:** signals for the two starting clusters. **Bottom left:** Spatial distribution of the clusters. **Bottom right:** data mapped in the eigenspace. A video showing the whole simulation can be found in the supplementary material of the paper.

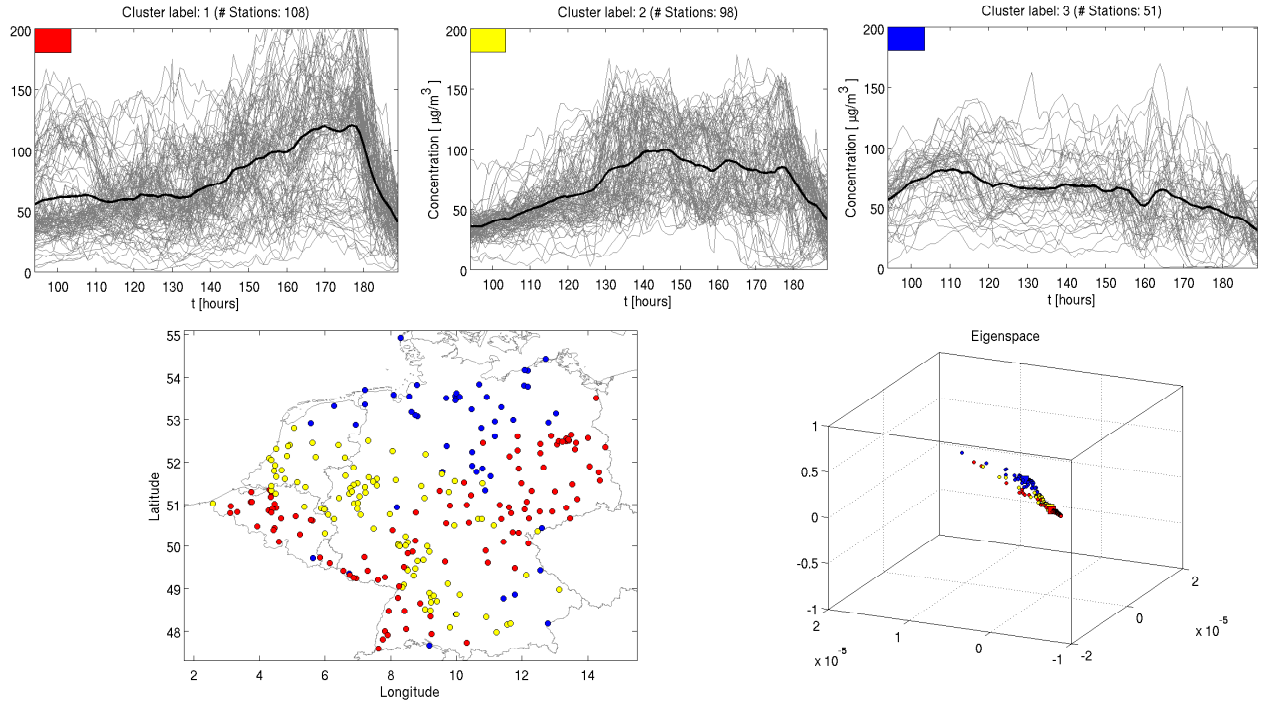


Figure 19: **PM_{10} clusters after creation. Top:** signals for the three clusters after the creation event. **Bottom left:** Spatial distribution of the clusters. Interestingly, the new cluster comprises stations located in the North-East part of Germany, which is the area where the pollutants coming from Eastern Europe started to spread during the heavy pollution episode of January 2010. **Bottom right:** data in the eigenspace.

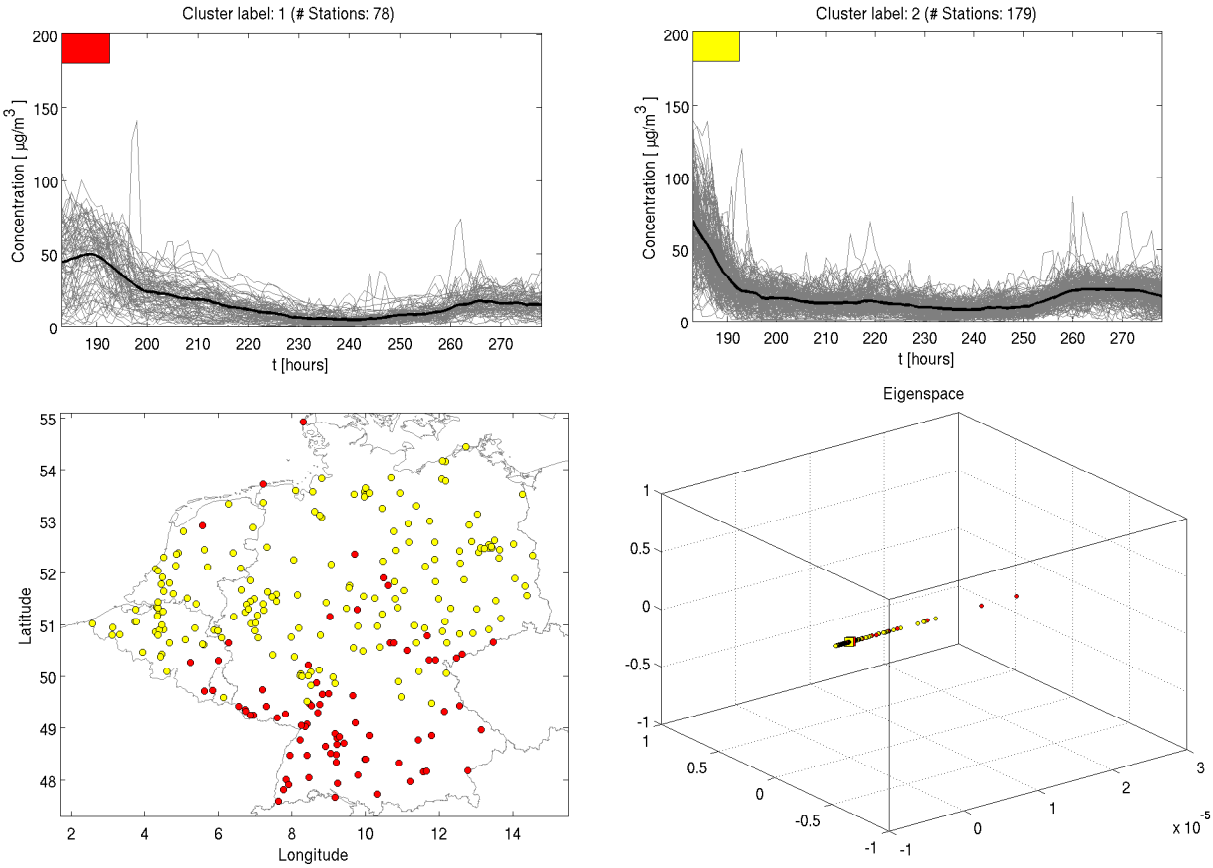


Figure 20: **Clustering model of PM_{10} stations after merging. Top:** two clusters left after the merging event occurred at time step $t = 251$. **Bottom left:** Spatial distribution of the clusters. **Bottom right:** data in the eigenspace.