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)/iMinds Future Health Department, SCD, 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 KKT optimality conditions for 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} concentration registered during a pollution episode occurred in Northern Europe in January 2010. We observe that our model is able to precisely recognize the dynamic 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} concentration.

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, and object tracking in computer vision represent just few examples. Therefore researchers perceived the need of developing clustering methods that can model the complex dynamic 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 [5, 7, 15], self-organizing time map [21], incremental K-means [6] constitute some examples. Here we focus our attention on the family of the Spectral Clustering (SC) approaches [19, 24, 8], 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 [20] and more recently in [9], 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, [2]). KSC is an LS-SVM formulation [22] of Spectral Clustering with two main advantages: an organized

model-selection procedure based on several criteria (BLF, Modularity, AMS, [2, 14, 16]) 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 [18]. 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 [20] and [9], can be used to accurately cluster stationary data-streams. 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. It is then to solve this issue that we introduce the Incremental Kernel Spectral Clustering Algorithm (IKSC). The IKSC method takes advantage of the work presented in [3] to continuously adjust the initial KSC model over-time, in order to learn the complex dynamic 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 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 manuscript.

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. 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 [2]:

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

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

where $e^{(l)} = [e_1^{(l)}, \ldots, e_N^{(l)}]^T$ are the projections, $l = 1, \ldots, k-1$ indicate the number 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 clustering model is expressed by:

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

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high-dimensional feature space, b_l are bias terms, with $l = 1, \ldots, k-1$. The projections $e_i^{(l)}$ represent the latent variables of a group of $k-1$ binary clustering indicators given by $\text{sign}(e_i^{(l)})$. The set of binary indicators 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. The Lagrangian associated

with the primal problem 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:

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

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 graph degree matrix which is diagonal with positive elements $D_{ii} = \sum_j \Omega_{ij}$, M_D is a centering matrix defined as $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}$, the $\alpha^{(l)}$ are dual variables, $\lambda_l = \frac{N}{\gamma_l}$. $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the kernel function and captures the similarity between the data-points.

In the first two the synthetic experiments (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 [17]. Thus $K(x_i, x_j) = \exp(-||x_i - x_j||_{cd}^2/\sigma^2)$, where the correlation distance is defined as $cd = \sqrt{\frac{1}{2}(1 - R_{ij})}$, 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 fulfill the positive definitiveness property.

The out-of-sample extension is performed by an Error Correcting Output Codes (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:

$$\text{sign}(e_{\text{test}}^{(l)}) = \text{sign}(\Omega_{\text{test}} \alpha^{(l)} + b_l 1_{\text{Ntest}}) \tag{5}$$

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

3. Incremental Kernel Spectral Clustering (IKSC)

3.1. Model-based update

In contrast with other techniques that compute approximate eigenvectors of large matrices or for dynamic data like the Nyström method [25], the work presented in [12] or the above-mentioned algorithms [9] and [20], the eigen-approximation we use to evolve the initial model is model-based [3]. This means that based on a training set (in our case the cluster centroids) out-of-sample eigenvectors are calculated using eq. (6). 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 [2, 23] (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. (5) and the ECOC decoding procedure. However, if the data source is non-stationary, this scheme fails since the initial model is not representative anymore 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 of 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:

$$\alpha^{(l)}_{\text{test}} = \frac{1}{\lambda_l} D^{-1}_{\text{test}} e^{(l)}_{\text{test}} \tag{6}$$

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 [3]. 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.(5), 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 (6)
- assign the new points to the closest centroids in the eigenspace and update them accordingly
- 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 [13]:

$$C_{\text{new}} = C_{\text{old}} + \frac{x_{\text{new}} - C_{\text{old}}}{n_{\text{old}}} \tag{7}$$

where n_{old} is the number of samples previously assigned to the cluster center C_{old}. A schematic visualization of this procedure is depicted in Fig. 1. Finally, here we sketch the complete IKSC algorithm:

Algorithm IKSC Incremental Kernel Spectral Clustering algorithm

---

Input: Training set $\mathcal{D} = \{x_i\}_{i=1}^N$, 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: Clusters $\{\mathcal{A}_1, \ldots, \mathcal{A}_p\}$, cluster centroids in the input space $C_1, \ldots, C_k$, cluster centroids in the eigenspace $C_1^\alpha, \ldots, C_k^\alpha$.

Initialization:

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

Online IKSC:

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

1. compute the out-of-sample eigenvectors using eq. (6)
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. (7)
4. update centroids in the input space $C_1, \ldots, C_k$ according to eq. (7)
5. new cluster check
6. merge check
7. cluster death

endfor

Outlier elimination.

The new cluster check allows to dynamically create a new cluster if necessary. For every new point the related degree is calculated. If it is less than 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. Throughout the merge check, if two centroids become too similar they are merged into one center. Moreover, if the centroid of a cluster is not updated anymore the algorithm consider that cluster as disappeared (cluster death). Finally, if one cluster is formed by less than 5 points is considered as outlier and is eliminated in the end of the data-stream acquisition.

3.3. Computational complexity

Let us assume that at the beginning of the data-stream acquisition the training set $\mathcal{D} = \{x_i\}_{i=1}^k, x_i \in \mathbb{R}^d$ is formed by k centroids in the input space $C_1, \ldots, C_k$ and the initial clustering model is expressed by the centroids in the eigenspace $C_1^\alpha, \ldots, C_k^\alpha$, with $C_i^\alpha \in \mathbb{R}^{k-1}$. 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. (6) and the update of both the model and the training set (that is the clusters centers in the input space). The main contribution to the computational complexity is due to the out-of-sample extension part:

$$e_{\text{test}}^{(l)} = \Omega_{\text{test}} \alpha^{(l)} + b_l 1_{\text{Ntest}}, l = 1, \ldots, k-1. \tag{8}$$

The evaluation of the kernel matrix Ω_{test} needs $O(k^2 d)$ operations to be performed. The calculation of the score variable $e_{\text{test}}^{(l)}$ takes then $O(k^2 d + k^2 + k)$ time. This operation has to be repeated for the N data-points of the data-stream, so the overall time complexity is $O(N(k^2 d + k^2 + k))$. This can become linear with respect to the number of data-points ($O(N)$) when $k \ll N$ and $d \ll N$, which is the case in many applications. This is comparable with other eigen-updating algorithms for spectral clustering like [20] and [9].

4. Data-sets

4.1. Artificial data

Three simulations are performed: the first and the second by reproducing the experiments described in [4], 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. 2. 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. 3.

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. 4. 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 the second type at time step $t = 150$. Moreover when these signals get back to their original frequency at time step $t = 300$ a 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 [10]. 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 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.5). 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 behavior of evolving patterns of non-stationary data.

To evaluate the outcomes of the model, two cluster quality measures are computed [11]: the average cumulated 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 [9]. 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. 6. In the initialization phase 30 points are used to construct the model. The IKSC algorithm can perfectly model the two drifting distributions: the average cumulated 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. 7. Concerning the second artificial dataset, similar comments can be drawn. The cluster quality stays high over time, in almost perfect agreement with the-ground truth (see Figs. 8, 9 and 10). Moreover at time-step $t = 6926$ the two moving Gaussian clouds, as expected, are merged. Only in this case, as observed also in [4], there is a small increase in the average cumulated ARI error. The small cluster at the bottom left side of Fig. 3 is detected as outlier after the data acquisition.

Finally, in Figs. 11, 12 and 13 the results of IKSC on the synthetic time-series experiment is illustrated. 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 $t = 223$. Moreover the second change point is detected at time step $t = 382$, when a merging of two clusters is performed. 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. 14 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 amazing: 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 perturbated. 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. 15. 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 model selection criterion [16]. We find $k = 2$ and $\sigma^2 = 0.05$ as optimal parameters, like depicted in Fig. 16. The initial model, based on these parameters, is illustrated in Fig. 17. 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 correspond 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. 18. Later on these three clusters evolve until a merge of two of them occurs at time step $t = 251$ (see Fig. 19). If we analyze 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). Afterward 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 dynamic of the pollution cloud at this level of accuracy is somehow 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 [6].

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 Silhouette.

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 modeling the cluster evolution over-time (drifting, merging, outlier elimination etc.). Then we analyzed a real-world data-set consisting of PM_{10} concentration during an heavy pollution episode taking place for a period of 12 days on January 2010 in Northern Europe. 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 dynamic of PM_{10} concentration.

Acknowledgements

This work was supported by Research Council KUL: ERC Adv. 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., 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

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

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

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

[12] 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.

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

[14] 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.

[15] 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.

[16] Langone, R., Mall, R., Suykens, J. A. K., 2013. Soft kernel spectral clustering. In: IJCNN 2013. In press.

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

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

[25] 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.

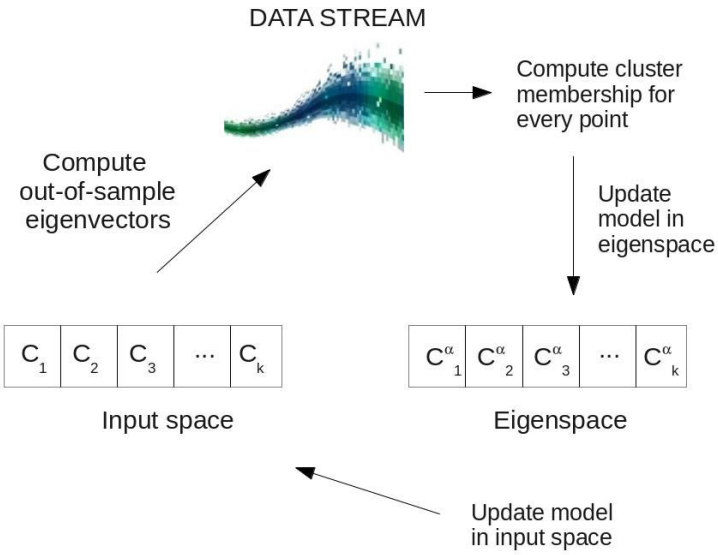


Figure 1: **IKSC update scheme**

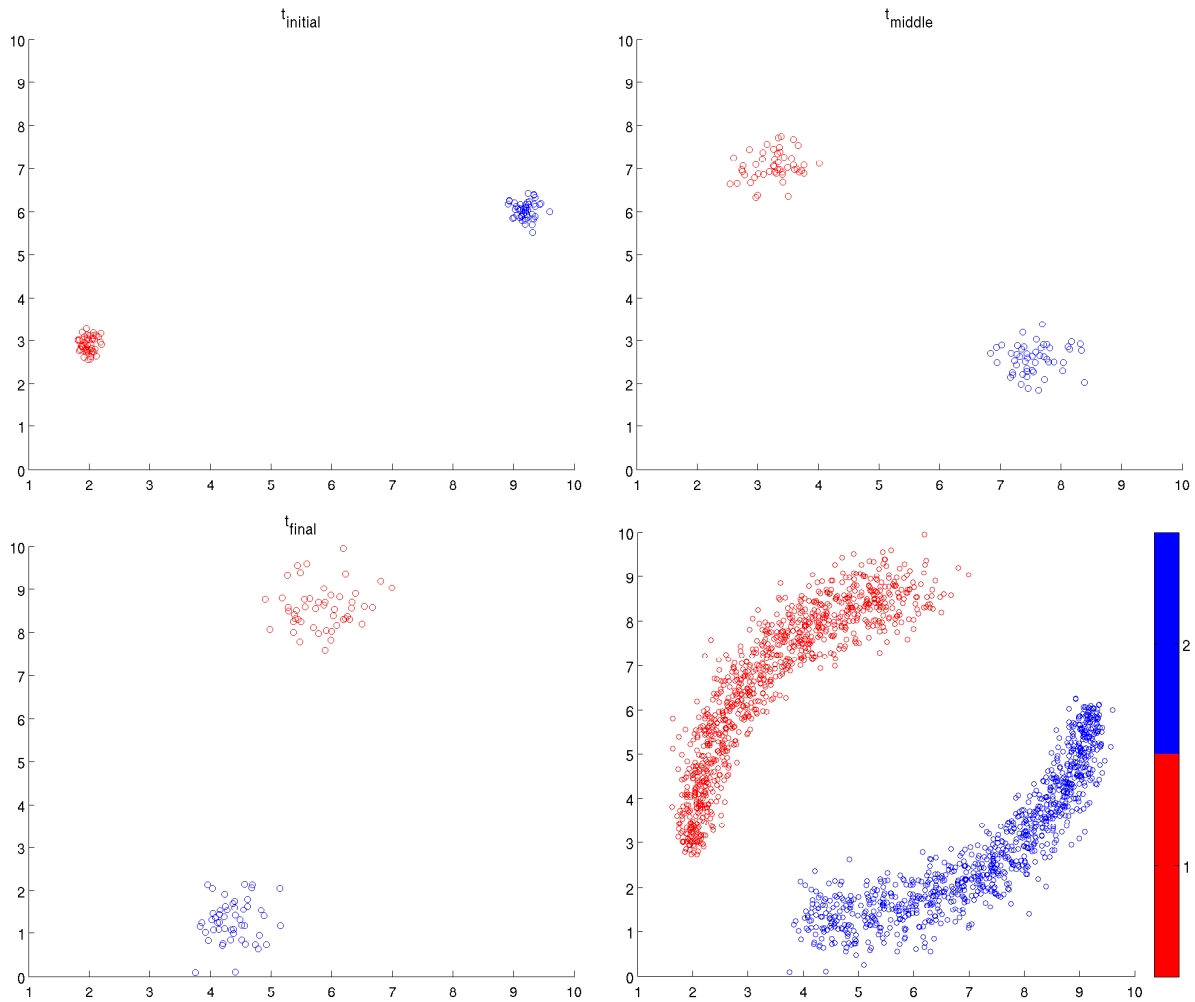


Figure 2: **Drifting Gaussian distributions.** Some snapshots of the evolution of the distributions (top and bottom left), and the whole data all at once (bottom right).

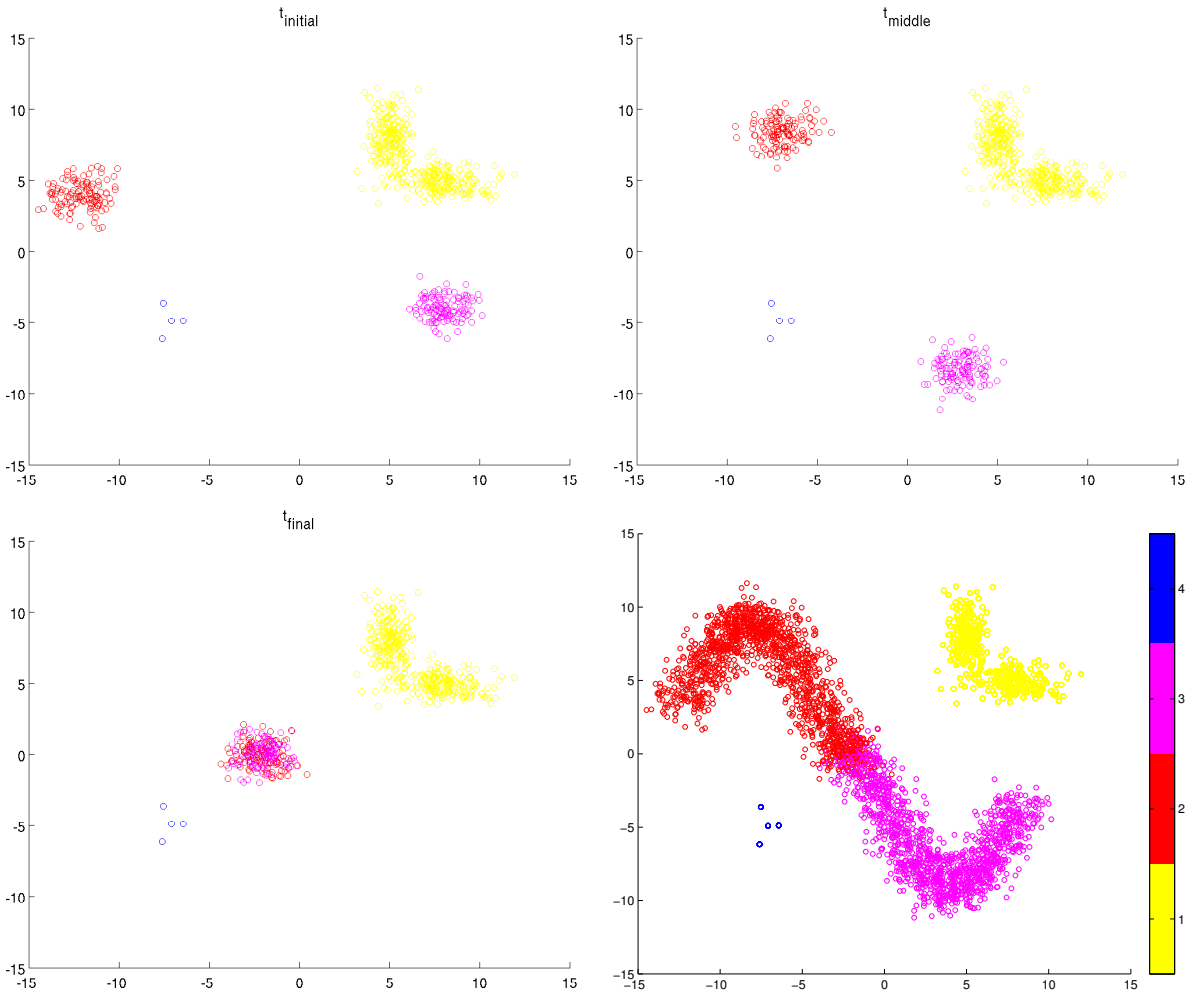


Figure 3: **Merging Gaussian distributions.** Some snapshots of the evolution of the distributions (top and bottom left), and the whole data all at once (bottom right).

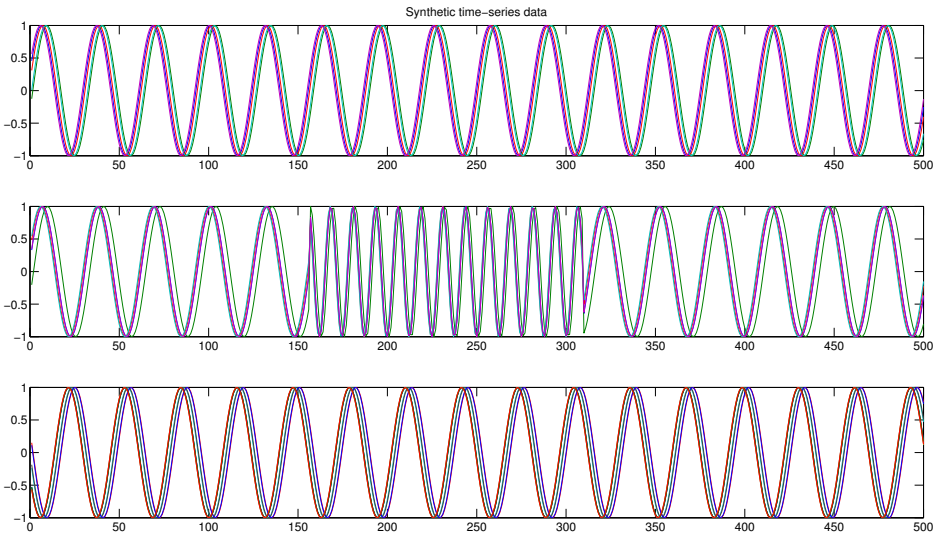


Figure 4: **Synthetic time-series.**

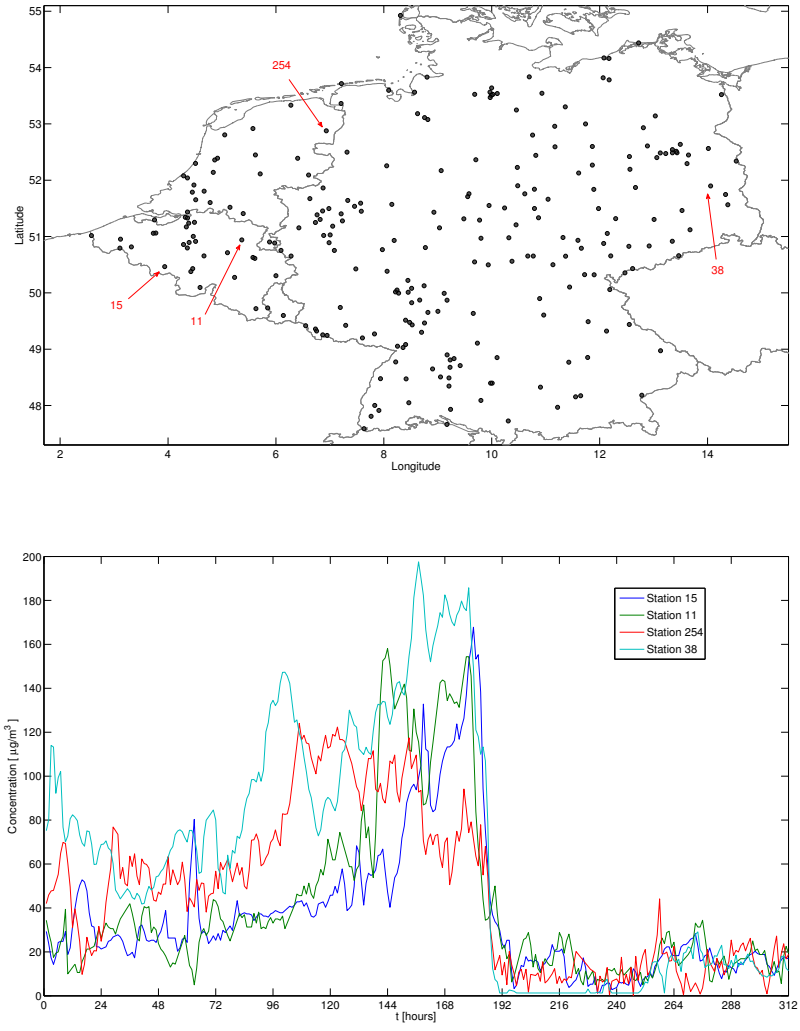


Figure 5: **$\mathbf{PM_{10}}$ data. Top:** AirBase monitoring stations. **Bottom:** Some time-series of PM_{10} concentrations for the whole period under investigation.

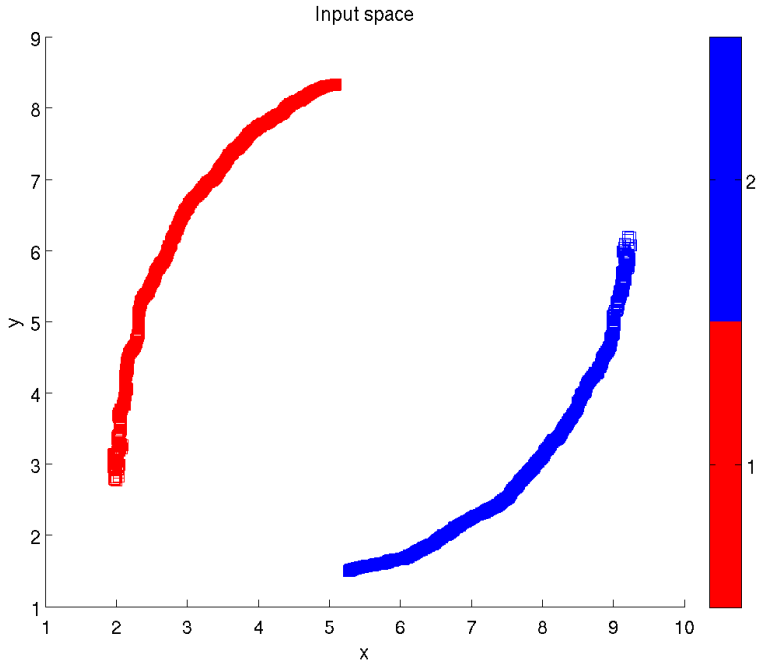


Figure 6: **Results of IKSC on the drifting Gaussian distributions.** Evolution of the centroids in the input space. We can notice that the IKSC model can recognize the drifting targets without errors. A video of the simulation is present in the supplementary material of this paper.

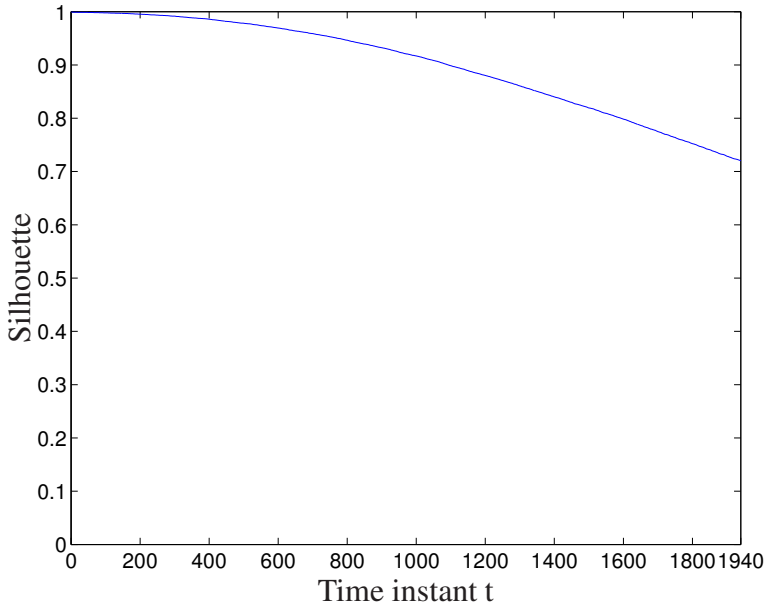


Figure 7: **Silhouette for drifting Gaussian distributions.** The mean silhouette value related to the clusters detected by IKSC stays high over time.

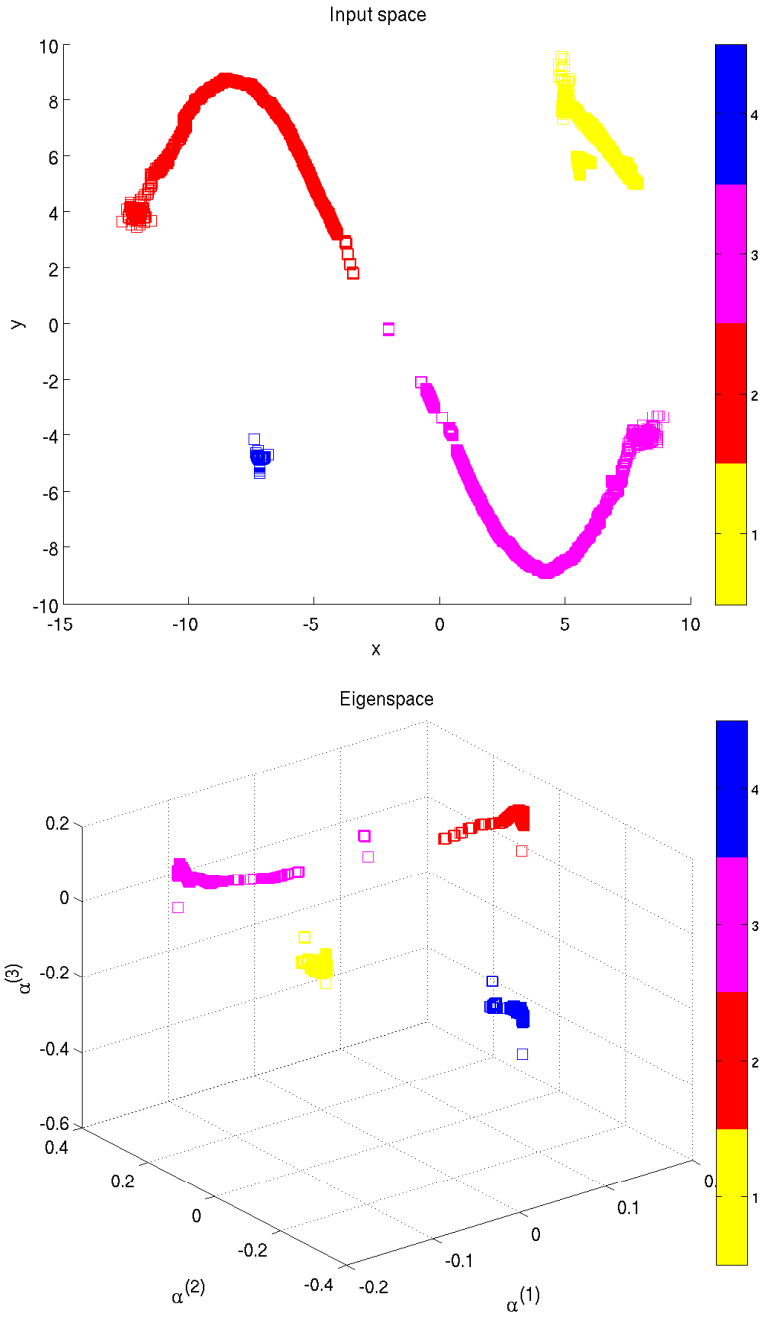


Figure 8: **Results of IKSC on the merging Gaussian distributions. Top:** Evolution of the centroids in the input space. **Bottom:** Model evolution in the eigenspace. A video of the simulation is provided as supplementary material of this paper.

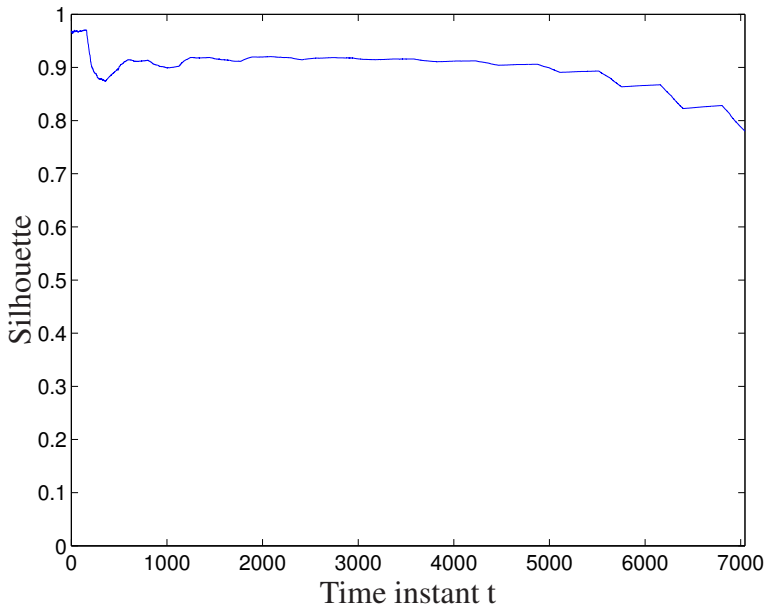


Figure 9: **Silhouette for the merging Gaussian distributions experiment.** The silhouette value related to the clusters detected by IKSC remains high over time.

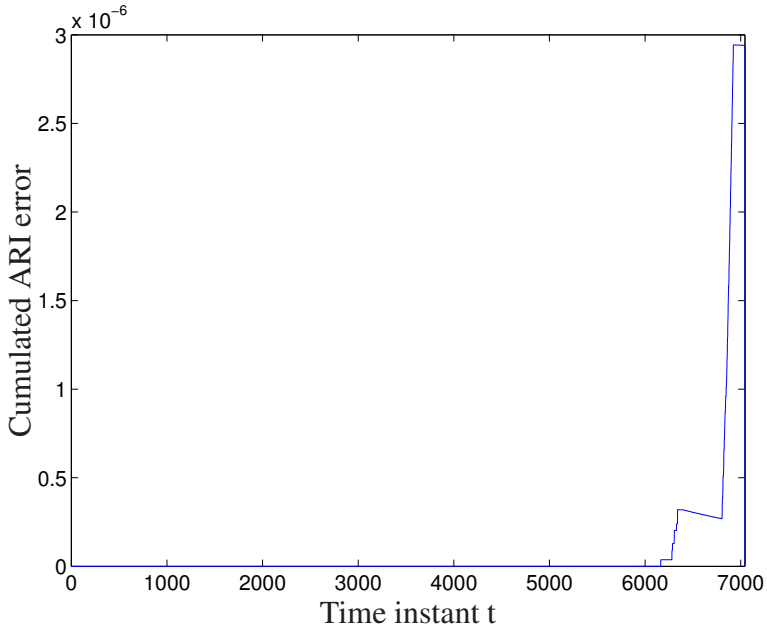


Figure 10: **ARI error-Merging Gaussian distributions.** The average cumulated ARI error related to the clusters detected by IKSC is very small over time, with a peak around the merging step at time $t = 6926$, in agreement with what was observed also in [4].

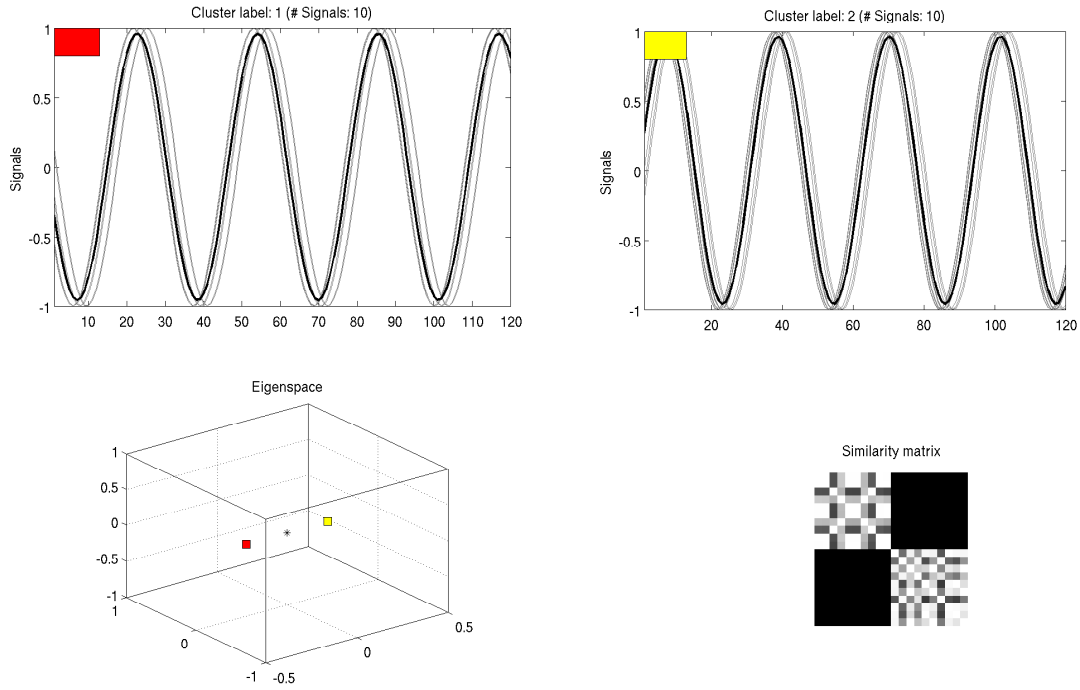


Figure 11: **Synthetic time-series initial clustering model. Top:** signals of the two starting clusters. **Bottom left**: data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right**: kernel matrix with a clear block diagonal structure.

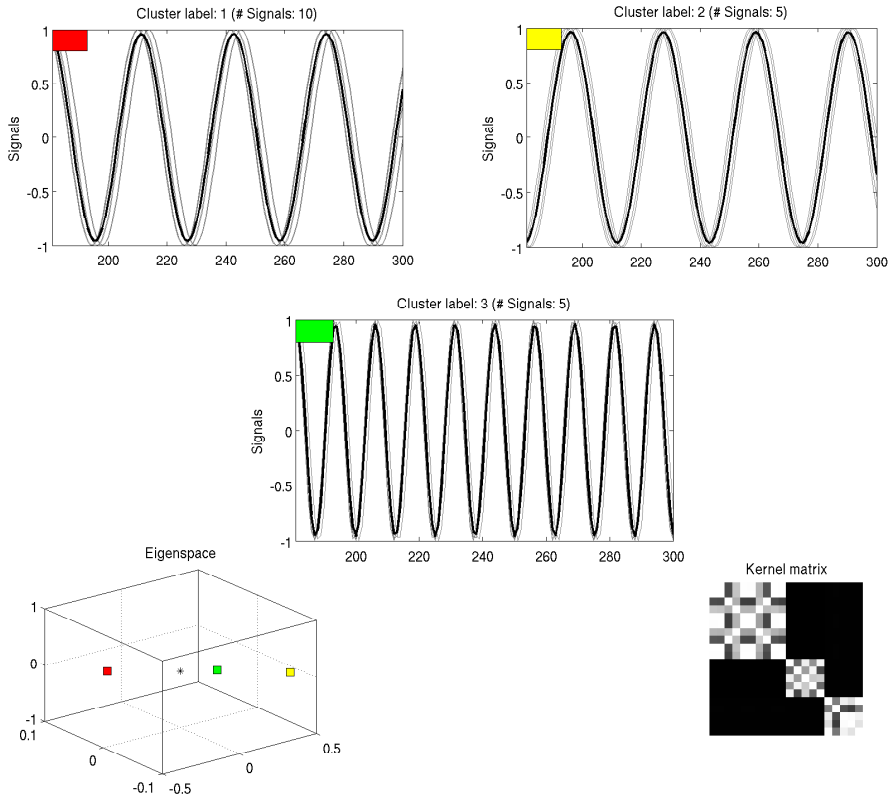


Figure 12: **Synthetic time-series clusters after creation. Top and center:** signals of the three clusters after the creation event. **Bottom left** data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right:** kernel matrix. A video of the entire simulation is present in the supplementary material of the paper.

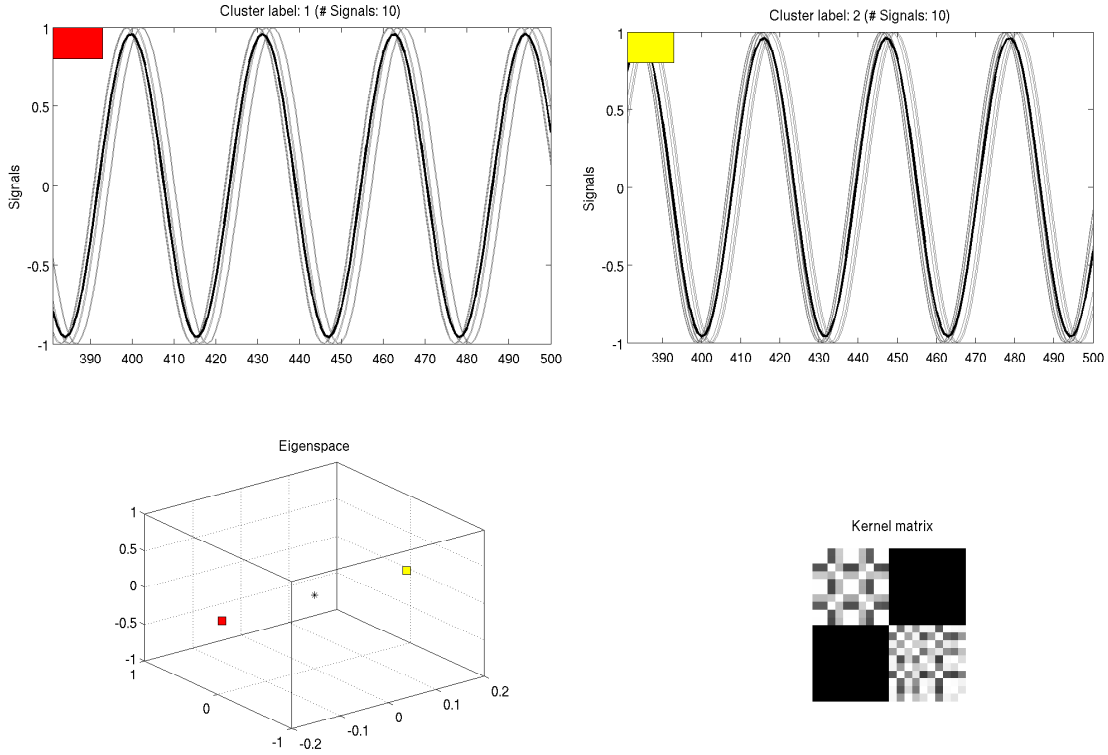


Figure 13: **Synthetic time-series final clustering model. Top:** two final clusters after the merging event. **Bottom left:** clustered data in the eigenspace (the points are mapped in the same location as the related centroids, since the eigenvectors are perfectly piece-wise constant). **Bottom right:** kernel matrix.

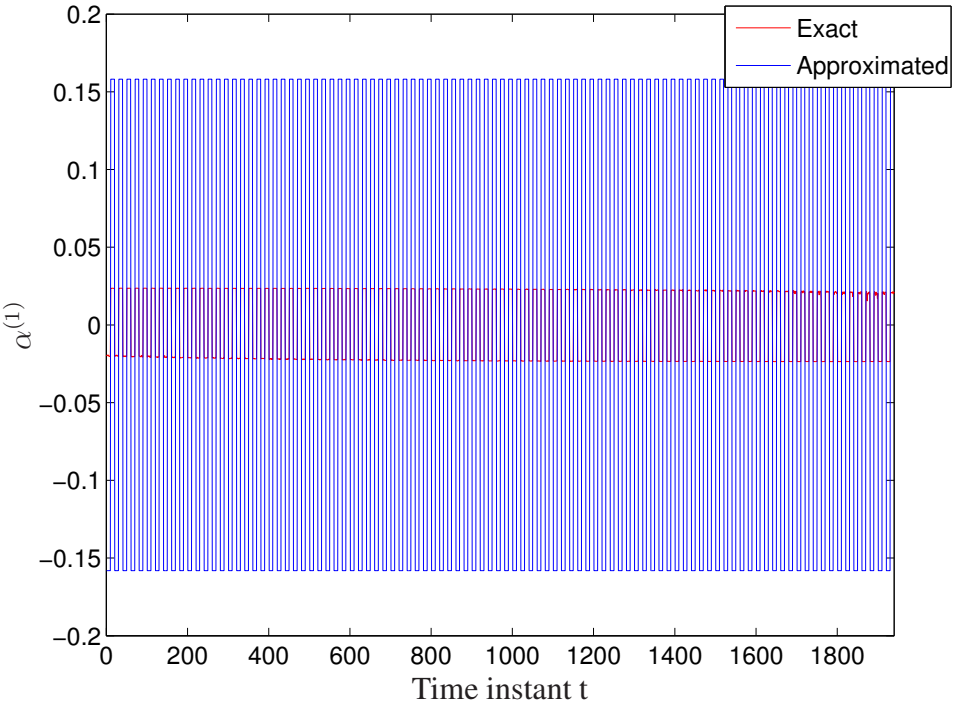


Figure 14: **Eigenvector-Drifting Gaussian distributions.** Exact and approximated eigenvector corresponding to the largest eigenvalue of the problem (4), for the first synthetic example.

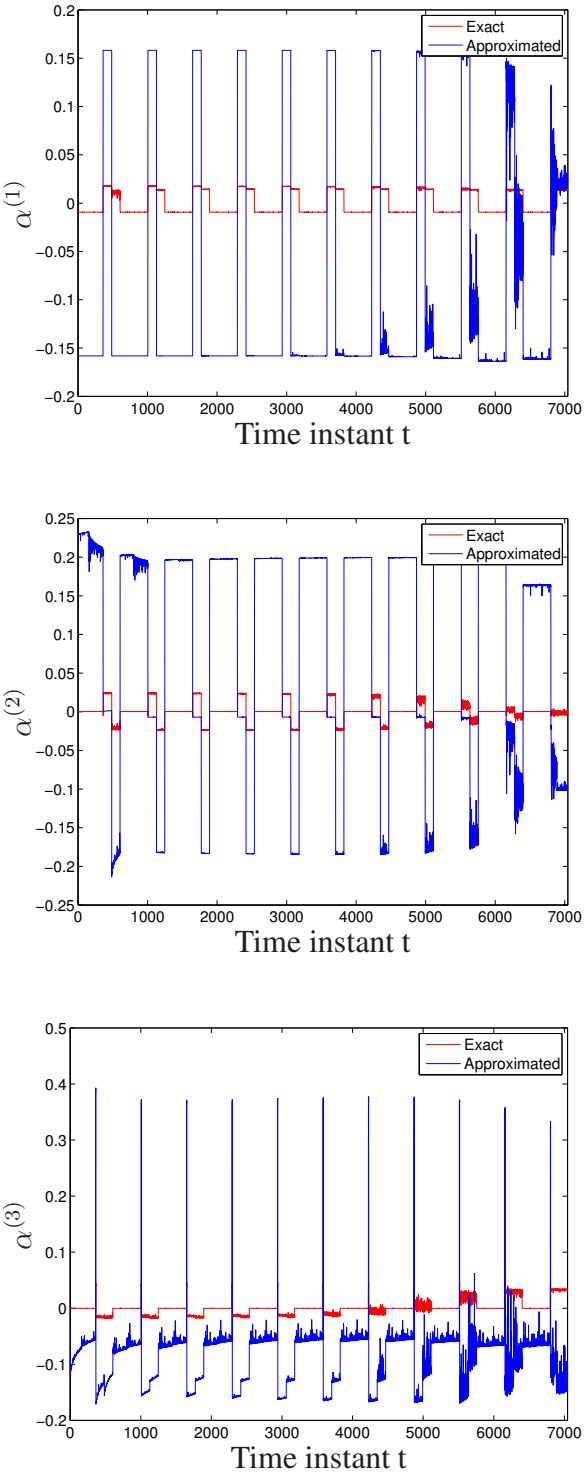


Figure 15: **Eigenvectors-Merging Gaussian distributions.** Exact and approximated eigenvectors corresponding to the 3 largest eigenvalues of the problem (4), for the second virtual experiment.

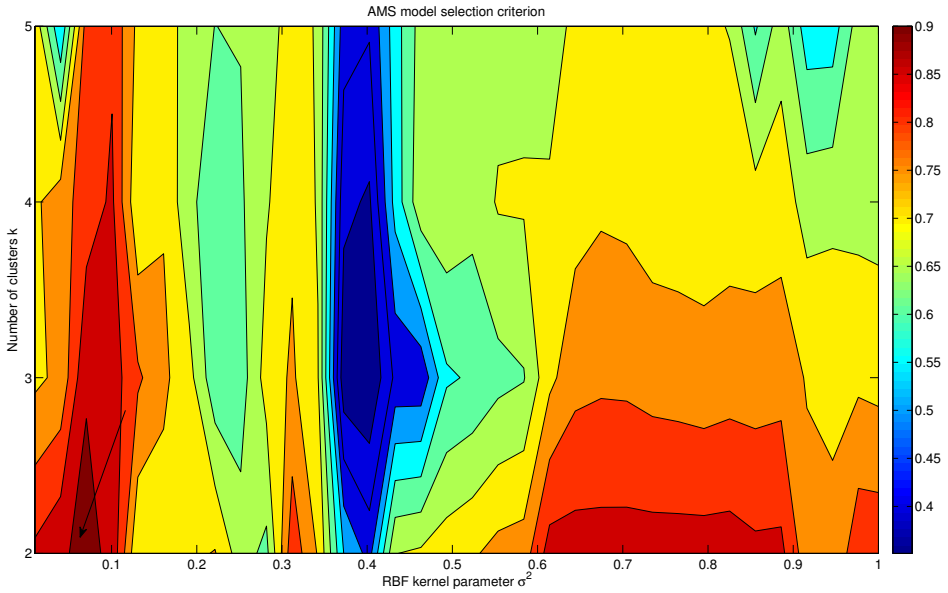


Figure 16: **Model selection.** Tuning of the number of clusters and the bandwidth of the RBF kernel in the initialization phase of IKSC for the analysis of the PM_{10} data.

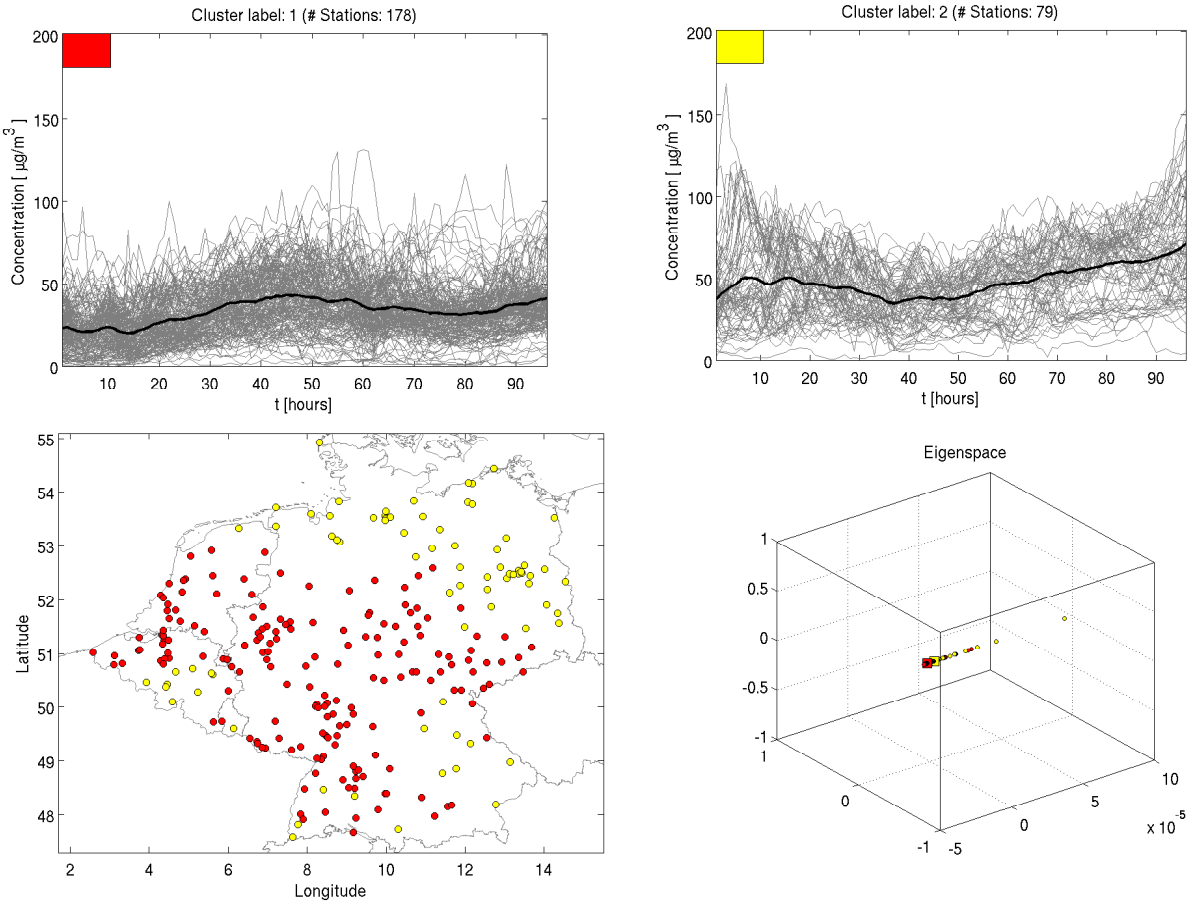


Figure 17: **Initial clustering model for the PM_{10} monitoring stations. Top:** signals for the two starting clusters. **Bottom left:** Spatial distribution of the clusters. **Bottom right:** data mapped in the eigenspace. A video showing the whole simulation can be found in the supplementary material of the paper.

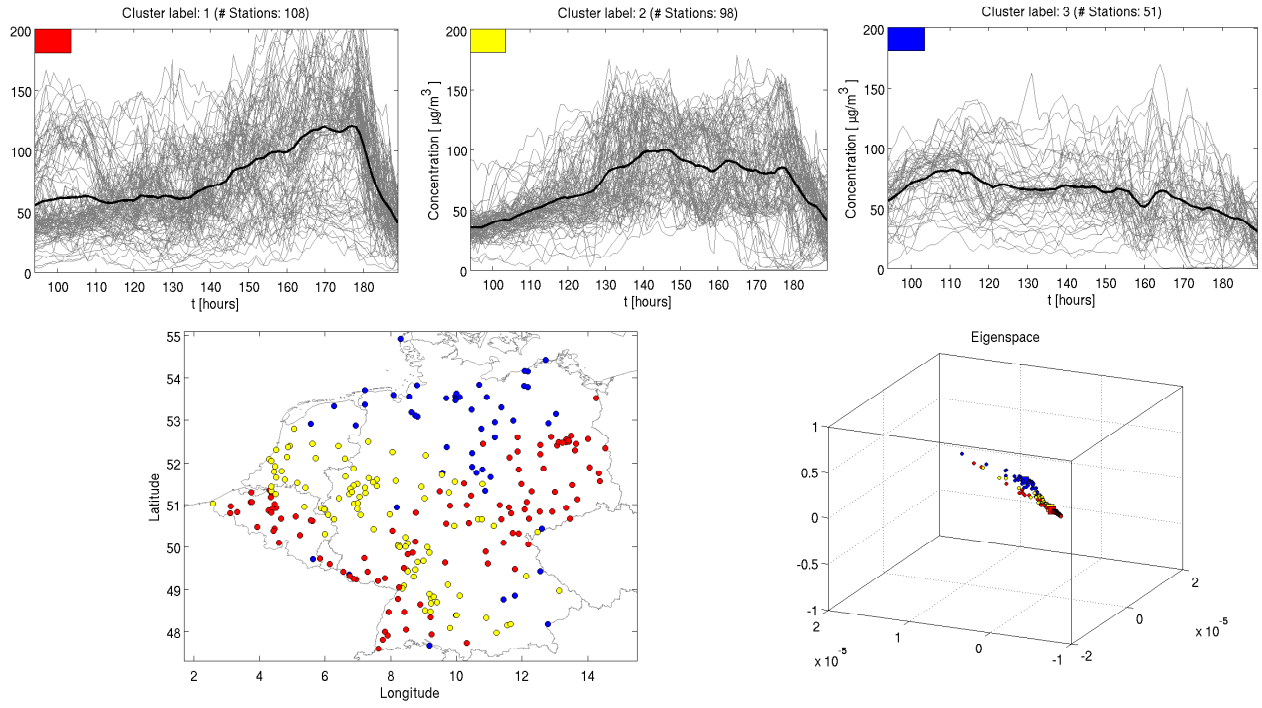


Figure 18: **PM_{10} clusters after creation. Top:** signals for the three clusters after the creation event. **Bottom left:** Spatial distribution of the clusters. **Bottom right:** data in the eigenspace.

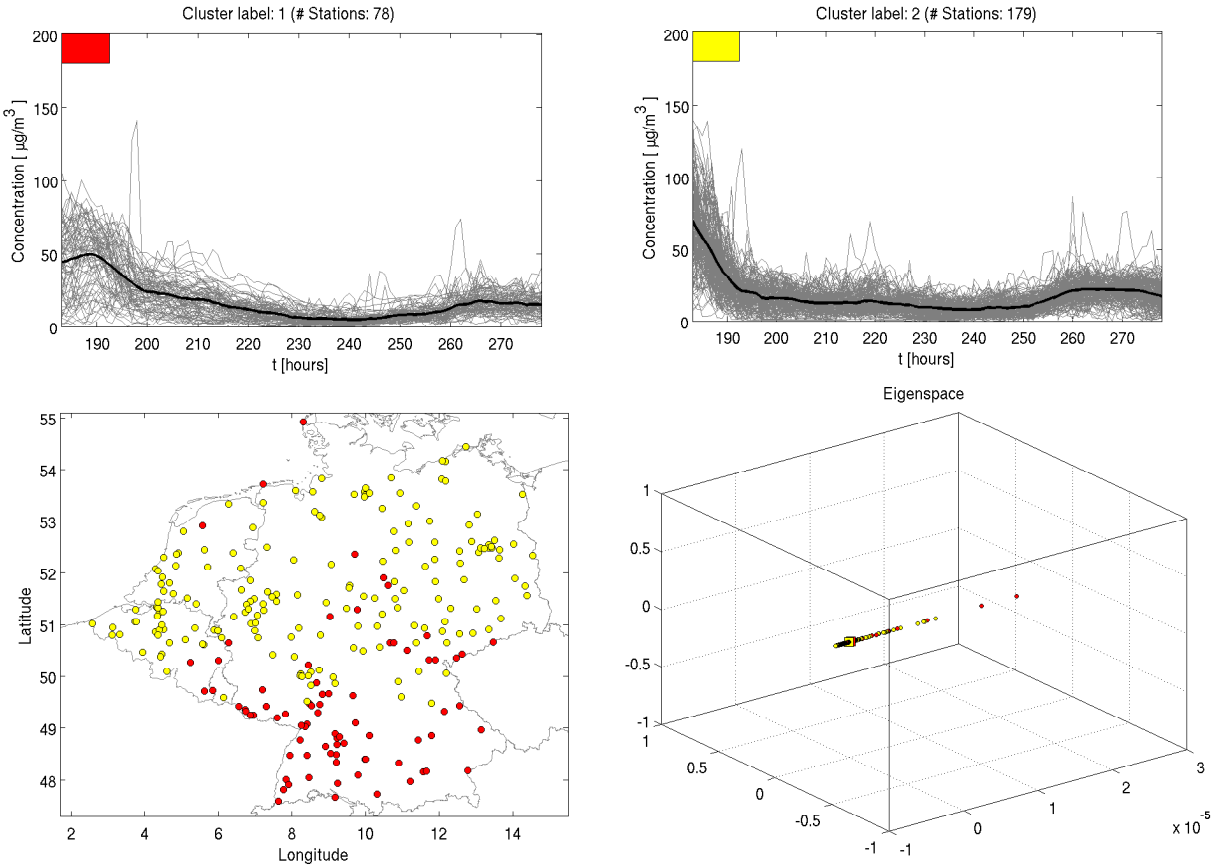


Figure 19: **Clustering model of PM_{10} stations after merging. Top:** two clusters left after the merging event occurred at time step $t = 251$. **Bottom left:** Spatial distribution of the clusters. **Bottom right:** data in the eigenspace.