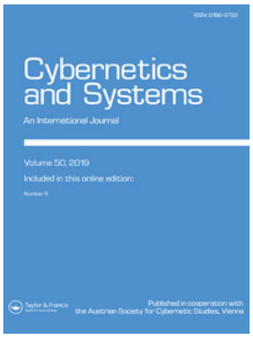

Taylor & Francis
Taylor & Francis Group

Cybernetics and Systems
An International Journal

ISSN: 0196-9722 (Print) 1087-6553 (Online) Journal homepage: https://www.tandfonline.com/loi/ucbs20

Subspace Clustering for Situation Assessment in Aquatic Drones: A Sensitivity Analysis for State-Model Improvement

Alberto Castellini, Manuele Bicego, Domenico Bloisi, Jason Blum, Francesco Masillo, Sergio Peignier & Alessandro Farinelli

To cite this article: Alberto Castellini, Manuele Bicego, Domenico Bloisi, Jason Blum, Francesco Masillo, Sergio Peignier & Alessandro Farinelli (2019) Subspace Clustering for Situation Assessment in Aquatic Drones: A Sensitivity Analysis for State-Model Improvement, Cybernetics and Systems, 50:8, 658-671, DOI: 10.1080/01969722.2019.1677333

To link to this article: https://doi.org/10.1080/01969722.2019.1677333

CYBERNETICS AND SYSTEMS: AN INTERNATIONAL JOURNAL
2019, VOL. 50, NO. 8, 658–671
https://doi.org/10.1080/01969722.2019.1677333

Subspace Clustering for Situation Assessment in Aquatic Drones: A Sensitivity Analysis for State-Model Improvement

Alberto Castellini[a], Manuele Bicego[a], Domenico Bloisi[b], Jason Blum[a], Francesco Masillo[a], Sergio Peignier[c], and Alessandro Farinelli[a]

[a]Department of Computer Science, University of Verona, Verona, Italy; [b]Department of Mathematics, Computer Science, and Economics, University of Basilicata, Italy; [c]INSA-Lyon, Université de Lyon, France

ABSTRACT

In this paper, we propose the use of subspace clustering to detect the states of dynamical systems from sequences of observations. In particular, we generate sparse and interpretable models that relate the states of aquatic drones involved in autonomous water monitoring to the properties (e.g., statistical distribution) of data collected by drone sensors. The subspace clustering algorithm used is called SubCMedians. A quantitative experimental analysis is performed to investigate the connections between i) learning parameters and performance, ii) noise in the data and performance. The clustering obtained with this analysis outperforms those generated by previous approaches.

KEYWORDS

Activity recognition; aquatic drones;autonomous vehicles; model interpretability; sensor data; situation assessment; subspace clustering; time series analysis; unsupervised learning; water monitoring

Introduction

Autonomous vehicles nowadays represent an important support for human activities (Farinelli et al. 2012). These intelligent systems are equipped with multiple sensors that gather large amount of sequential data from the operational environment. This data is used, for instance, by control units that select suitable actions according to the situations the vehicle is facing (Kaelbling and Lozano-Perez 2013). In the field of water monitoring, aquatic drones are increasingly used to acquire real-time data concerning different water parameters, including dissolved oxygen, pH and electrical conductivity, with minimum support of human operators (Bottarelli et al. 2019). The assessment of drone situations in this context is crucial for improving both the online control of the drone and the offline data analysis process, since drone states can affect the acquired data (Endsley 1995).

In this work, we focus on the problem of detecting, modeling and interpreting aquatic drone states from a data-driven point of view. We aim at using statistical learning methods to develop interpretable models of drone

CONTACT Alberto Castellini alberto.castellini@univr.it Department of Computer Science, University of Verona, Strada le Grazie, 15, 37134, Verona, Italy.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ucbs.

states from traces of sensor data acquired during water-monitoring missions. Unsupervised methods, such as clustering and time series segmentation, are ideal tools for detecting data patterns in the considered scenario since they optimize internal performance measures (Arbelaitz et al. 2013). Using these tools, similar observations are grouped together into clusters whose parameters represent the state models. Another advantage of such methods is that they avoid manual labeling of sensor traces that is often expensive, time consuming, and impracticable in some cases. Moreover, models generated by these methods are abstract descriptions of drone states that can be interpreted and validated by experts. Finally, being unsupervised, these techniques allow novelty detection.

Several methods for time series clustering and segmentation are available in the literature (Castellini, Paltrinieri, and Manca 2015). They mainly differ from each other in the assumptions they make on data or model properties. Some techniques are used for sensor-based human activity recognition (Chen et al. 2012), where sensors gather data about human movements from which computational activity models are generated. The main techniques used in this context are k-means (Bishop 2006; Abdallah et al. 2012; Trabelsi et al. 2013; Montanez, Amizadeh, and Laptev 2015; Barták and Vomlelová 2017), Gaussian mixture models (GMM) (Bishop 2006; Trabelsi et al. 2013; Barták and Vomlelová 2017), hierarchical clustering (Bishop 2006; Barták and Vomlelová 2017), hidden Markov models (HMMs) (Fox et al. 2008; Kim, Helal, and Cook 2010; Trabelsi et al. 2013; Barták and Vomlelová 2017), conditional random fields (CRFs) (Vail, Veloso, and Lafferty 2007), Markov random fields (Hallac et al. 2017) and change-point detection methods. However, only few of them were applied to data from drones or autonomous vehicles and none of them on aquatic drones. In our previous work (Castellini et al. 2018a, 2018b) we investigated this application domain using standard clustering techniques. Peculiarities of aquatic environment make activity recognition in this context very challenging. In particular, data is very noisy, it comes from several sources, and strongly depends on unstructured and diversified environments.

Here, we extend our recent results (Castellini et al. 2019) on *subspace clustering* for state-model generation for aquatic drones. Subspace clustering is an adaptation of clustering for high dimensional data (Parsons, Haque, and Liu 2004). This approach tackles two different problems simultaneously, namely, detecting clusters in the dataset and searching a relevant subspace for each cluster. For this reason, it is recognized as more general than traditional clustering. Three major families of subspace clustering approaches are (Sim et al. 2013; Kriegel, Kröger, and Zimek 2009; Madeira and Oliveira 2004): *cell-based* approaches, that search hyper-rectangular

clusters, *density-based* methods, that detect groups of objects separated by low density zones, and *clustering-oriented* approaches, that use distance-based similarity measures to form hyper-spherical shaped clusters.

We use a recent center-based technique, called *SubCMedians* (Peignier et al. 2018), on a dataset containing suitable variables extracted from sensor traces of six data collection campaigns. Results show that the proposed sensibility analysis of clustering performance with respect to SubCMedians learning parameters allows identifying a model that outperforms the one generated in (Castellini et al. 2019). Moreover, a second sensitivity analysis of clustering performance to noise shows that the achieved clustering is informative, since its performance and properties are significantly different from those obtained from noisy (i.e., uninformative) data. Finally, the state-models represented by cluster centroids are investigated and interpreted in terms of drone situations, showing that both known states (e.g., upstream navigation) and novel ones (e.g., drone curves and pitching) can be detected in a completely unsupervised way.

The contribution of this work to the state-of-the-art is three-fold:

- We propose an improved clustering of a real-world dataset collected by aquatic drones, by means of a sensitivity analysis performed to investigate the impact of learning parameters on clustering quality;
- We prove the significance of the proposed solution by a sensitivity analysis of noisy data on clustering performance;
- We provide an informative interpretation of some state-models detected by the proposed approach.

The rest of the manuscript is organized as follows. Section *Material and methods* introduces the drone architecture, the dataset and SubCMedians. Section *Results* presents the outcomes of our sensitivity analyses and the state-models discovered by the optimal clustering. Section *Conclusions* draws some conclusions and ideas for future work.

Material and Methods

In this section, we describe the dataset collected by aquatic drones, the SubCMedians algorithm, and the performance measures used for the evaluation.

Dataset

Data acquisition was performed by aquatic drones developed in the EU-funded Horizon 2020 project INTCATCH[1] (see Figure 1). They are

[1] http://www.intcatch.eu

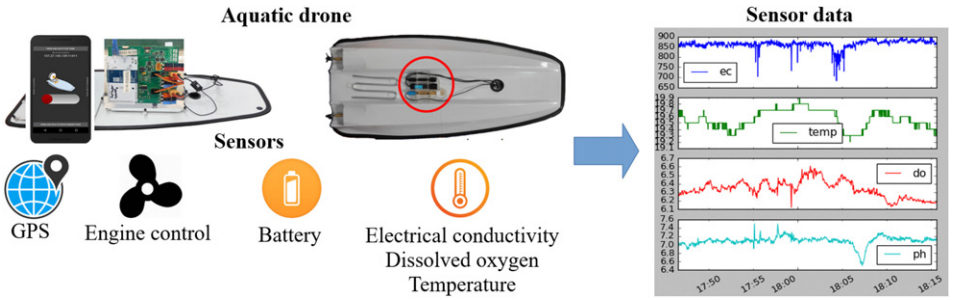

Figure 1. Data acquisition: aquatic drones (on the left) and sensor data acquired by these drones (on the right).

equipped with sensors for temperature, dissolved oxygen, electrical conductivity, GPS position, commands to the two propellers and battery voltage. The dataset used in this work has 20187 observations (about 5.6 hours of navigation with sampling frequency of 1 Hz) taken during six campaigns. We performed feature extraction from sensor signals obtaining the following 27 variables: *i)* instantaneous speed, voltage, acceleration, signal to propellers 0 and 1; *ii)* moving average (with sliding window of 10 seconds) of speed, voltage, acceleration, signal to propellers 0 and 1; *iii)* moving standard deviation (with sliding window of 10 seconds) of speed, voltage, acceleration, electrical conductivity, dissolved oxygen, temperature, signal to propellers 0 and 1, heading; *iv)* variation (value at time *i* minus value at time *i-1*) of speed, voltage, acceleration, electrical conductivity, dissolved oxygen, temperature, signal to propellers 0 and 1. Z-score standardization was performed on each variable. Therefore, the data matrix used to generate the subspace clustering by SubCMedians has 20187 observations (rows) and 27 variables (columns).

Known States and Manual Labeling

We manually labeled our dataset according to seven drone states that are easy to recognize using map plotting, but difficult to discover from sensor data, namely, in water (IW), out water (OW), upstream navigation (US), downstream navigation (DS), no water current (NS), manual drive (MD), and autonomous drive (AD). We used the above listed labeling to evaluate the capability of SubCMedians to detect meaningful states (see Subsection *Performance evaluation*).

SubCMedians Subspace Clustering

SubCMedians (Peignier et al. 2018) is a recent center-based subspace clustering technique based on a K-medians paradigm. It aims at clustering data

points around suitable candidate centers described in their own subspace. Cluster subspace variables are the most informative variables for the cluster, and centroid coordinates along such variables represent the coordinates of the cluster points. In this work, each subspace cluster represents a potential state of the aquatic drone and each cluster centroid represents a state-model.

Let us denote by $X = \{x_1, x_2, \ldots\}$ the set of observations in the dataset, where each point $x \in X$ is a vector of D variables (point coordinates). Let M denote the set of centers built by SubCMedians, such that each center $m_i \in M$ is defined in its own subspace (i.e., subset of variables) $D_i \subseteq D$. The size of a model M, is defined as the sum of the number of variables contained in the subspaces of the model centers, namely $Size(M) = \sum_i |D_i|$, which is intuitively interpreted as the "level of detail" of the model. In SubCMedians, the distance between a point x and a center m_i is an extension of the Manhattan distance that allows to compare points defined in different subspaces, $dist(x, m_i) = \sum_{d \in D_i} |x_d - m_{i,d}| + \sum_{d \in D \setminus D_i} |x_d - \mu_d|$, where $m_{i,d}$ the coordinate of m_i along variable d, and with μ_d the mean of the coordinates of all points in X along variable d. The distance between each point $x \in X$ and its closest center $m_i \in M$ is called the Absolute Error $AE(x, M) = min_{m_i \in M} dist(x, m_i)$. The goal of SubCMedians is to build a set of centers M that minimizes the Sum of Absolute Errors $SAE(X, M) = \sum_{x \in X} AE(x, M)$, and such that $Size(M) \leq SD_{max}$, where SD_{max} is a parameter denoting the maximum Sum of Dimensions used in M to describe all its centers (the number of centers is not constrained).

SubCMedians updates iteratively the coordinates and the subspaces of its centers, using a stochastic hill climbing technique. It takes advantage of a weight-based strategy to guide its local search towards most promising subspace clusters, in order to minimize the Sum of Absolute Errors, while satisfying the maximum model size constraint. The algorithm has three main parameters, namely, SD_{max}, the sample size N (the algorithm considers only N randomly chosen observations at each iteration) and the number of iterations *NbIter*. Following the guidelines in (Peignier et al. 2018) these parameters can be computed from a single parameter, the expected number of clusters *NbExpClust*. The actual number of clusters is then selected automatically by the algorithm at runtime.

Performance Evaluation

We measured the clustering and cluster performance by an internal measure called *silhouette* (Arbelaitz et al. 2013). The silhouette of the *i*-th data point is computed as $S(i) = \frac{b(i) - a(i)}{max\ \{a(i),\ b(i)\}}$, where *a(i)* is the average

dissimilarity of point i with all other data within the same cluster and $b(i)$ is the lowest average dissimilarity of point i to any other cluster, of which i is not a member. The silhouette of a clustering is the mean of the silhouettes of all points in the dataset, while the silhouette of a single cluster is the mean of the silhouettes of points in the cluster. Values range from -1 to 1 where values close to 1 indicate points belonging to perfectly compact and separated clusters and values close to -1 indicate clustering with mixed clusters.

Single clusters (i.e., state-models) were evaluated also by an external measure (which refers to manual labeling) called *precision*. The precision of a cluster c with respect to a label class l measures the extent to which the cluster contains the label class. It is computed by the formula $P(c) = \frac{|c \cap l|}{|c|}$, where $|c \cap l|$ is the number of observations belonging to both the cluster and the label class and $|c|$ is the number of observations in the cluster. The maximum value, namely 1, is obtained when all the observations in cluster c belong also to the label class l, while the minimum value, namely 0, is obtained when no observation in c belong to class label l.

Procedure for Testing the Sensitivity of Clustering Performance to Noise

We considered two kinds of noise distributions, namely, Gaussian and uniform, since they may have a different impact on the cluster subspace selection and the clustering performance. In both cases, we first standardized each variable. For *Gaussian* noise, we added to each variable a Gaussian noise with zero mean and a specific standard deviation $\sigma \in \{0, 0.10, 0.25, 0.50, 0.75, 1, 10, 100\}$, then we standardized again each variable and computed the clustering. For *uniform* noise, we performed the same steps, but we added to each variable a uniform noise in one of the ranges $r \in \{\pm 0, \pm 0.10, \pm 0.25, \pm 0.50, \pm 0.75, \pm 1, \pm 10, \pm 100\}$, then we standardized each variable and generated the clustering models. In both cases (i.e., Gaussian and uniform noise), we finally generated a completely random dataset and computed also its clustering (this test is named "complete noise" in Figure 4).

Results

In this section, we present the results of two kinds of sensitivity analysis that improve the generation of drone state-models by SubCMedians. The first analysis allows to identify an optimal clustering solution across a large set of combinations of learning parameters, the second one allows to evaluate the significance of the performance of the proposed solution with respect to the performance of a clustering generated on a random dataset. Finally, we provide some details about the optimal solution, compare it

Table 1. list of values used to compute the sensitivity of clustering performance to SubCMedians learning parameters.

Parameter	Values
SD_{max}	150, 250, 350, 450
N	200, 400, 600, 800
NbIter	40000, 50000, 60000, 70000

with the one proposed in Castellini et al. (2019) and highlight the improvements achieved by the sensitivity analysis.

Sensitivity of Clustering Performance to Changes in Learning Parameters

The state-models proposed in Castellini et al. (2019) were computed using SubCMedians with learning parameters SD_{max}=270, $N = 500$, $NbIter = 54000$. This setting was derived from the expected number of clusters $NbExpClust = 10$ according to formulas defined in Peignier et al. (2018). The analysis presented here aims to investigate the relationships between SubCMedians' learning parameters SD_{max}, N and *NbIter*, and clustering performance in the specific context of our case study. To discover these relationships, we tested all the combinations of parameter values listed in Table 1. For each combination of parameters, we computed 30 clusterings and kept only the one having the lowest SAE. Then, we computed the silhouette S and the silhouette in the subspaces SS. At the end, we obtained 64 clustering models (from 4 values of SD_{max}, 4 values of N and 4 values of *NbIter*) each of them being evaluated by 3 performance measures (namely, SAE, S and SS).

To quantify the relationships between learning parameters and clustering performance, we computed the correlation between these variables. Figure 2 shows a matrix containing the correlations of all pairs of variables SD_{max}, *N, NbIter, K, SAE,* S and SS. Each cell contains the correlation between the variables in the row and that in the column of the cell itself. Notice that the number of clusters K depends on parameters SD_{max}, N, and *NbIter*. Since it strongly influences clustering performance, we consider it as a learning parameter in our analysis.

The red boxes in the matrix highlight the pairs of variables having correlation greater than 0.5. As expected, the SAE (in the first column), is strongly influenced by K, SD_{max}, and N, but it is not influenced by the number of iterations *NbIter* (at least in the range between 40000 and 70000 that we tested, probably because all these values are large enough). The stronger influence comes from K which is controlled by SD_{max} in SubCMedians, and the negative correlation says that the SAE tends to decrease when SD_{max} and K increase, which makes sense since as the number of clusters increases, the algorithm has greater degrees of freedom to

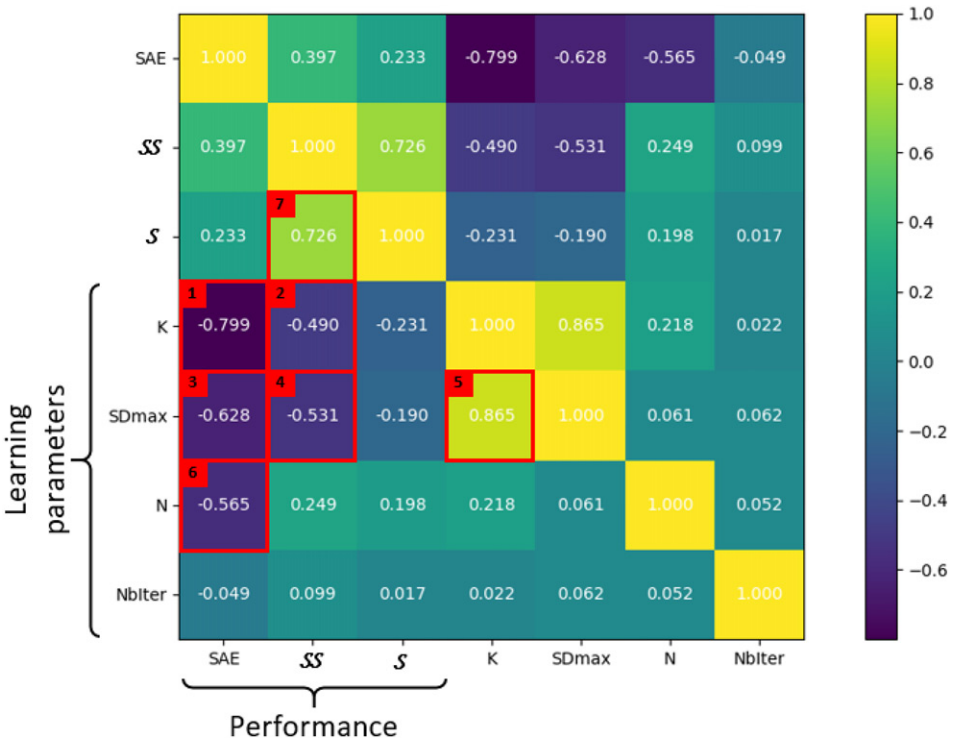

Figure 2. Correlation matrix between learning parameters SD_{max}, N, NbIter, K and performance SAE, S, SS of SubCMedians on our dataset.

improve performance. Similarly, increasing the sample size N leads to a better performance (i.e., decrease of SAE). These trends can be analyzed more precisely in Figure 3, where for each pair *(parameter, performance)* we show a specific scatter plot. In particular, parameters are arranged on the columns (and on the x-axes of the scatter plots) and performance on the rows (and on the y-axes of the scatter plots). Markers color and size represent the number of clusters K of each clustering. The total number of points in each scatter plot is 64, since each point represents a clustering. In all scatter plots a red marker identifies the model presented in Castellini et al. (2019) and a green marker the model selected in this analysis.

The best model was selected by analyzing the scatter plot of the number of clusters K against the clustering silhouette S. We focused on silhouette to be consistent with (Castellini et al. 2019) and because it is recognized as one of the most powerful cluster validity index (Arbelaitz et al. 2013). Since the silhouette tends to decrease when K increases, we looked for the larger clustering having also a large silhouette. Moreover, our goal was to identify an optimal clustering possibly having a number of clusters similar to that found in (Castellini et al. 2019), namely, $K = 26$. The solution we found (see green markers in Figure 3) has 24 clusters, a silhouette of about 0.074, a silhouette in the subspaces of

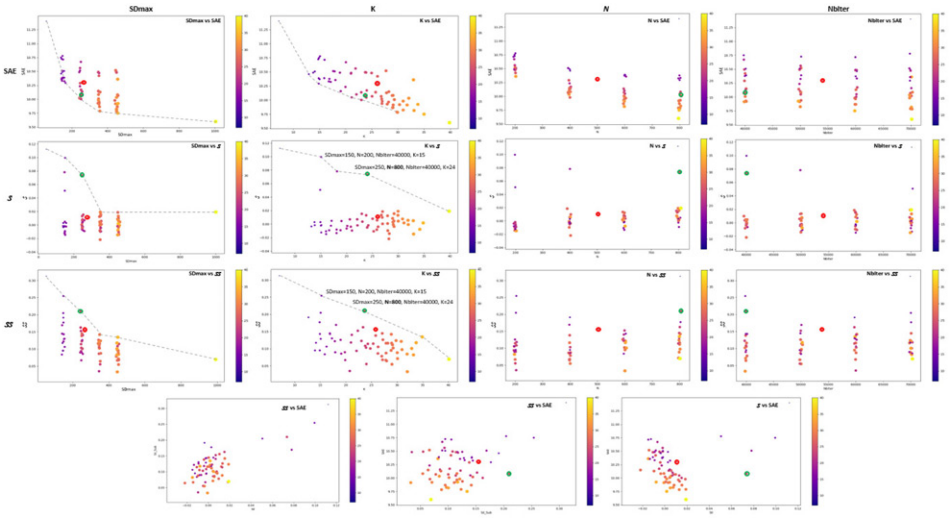

Figure 3. Relationships between learning parameters SD_{max}, K, N, Nblter (columns) and clustering performance SAE, S and SS (rows).

about 0.210 and it was generated by the learning parameters reported in Table 2. In the following we will refer to this optimal solution as C_{opt}. The same table shows also the parameters and performances of the clustering generated in (Castellini et al. 2019) without this sensitivity analysis. The improvement of performance is highlighted by bold values.

Sensitivity of Clustering Performance to Noise

The second part of our sensitivity analysis focuses only on the optimal clustering C_{opt} and aims at proving that the clustering performance of this clustering are significant for the specific dataset under investigation. We added different levels of noise to our dataset, according to the procedure described in section Material and methods (above), and analyzed the differences in related clustering performance. In this way we obtained some baseline values and trends of change for each performance measure that are useful to evaluate the performance of C_{opt}. Figure 4 shows the results of this analysis. The SAE starts from 10.1 (the value of C_{opt}) and grows until about 20 (the value for the completely noisy dataset) with different trends for Gaussian (circle marks) and uniform (square marks) noise. Interestingly, the number of clusters K also increases from 24 to about 51 (on average) for Gaussian noise and to 43 (on average) for uniform noise. This is due to the strategy used by SubCMedians to generate a clustering based on subspaces. When the cluster structure of the data is lost, it is no longer possible for SubCMedians to focus on specific locations along suitable features (i.e., subspace center coordinates) to decrease the SAE. The

Table 2. Learning parameters and performance of clustering C_{opt} and the best clustering generated in (CastelliniSAC2019).

Clustering	SDmax	N	NbIter	K	SAE	*S*	*SS*
C_{opt}	250	800	40000	24	**10.079**	**0.074**	**0.210**
(CastelliniSAC2019)	270	500	54000	26	10.299	0.015	0.155

algorithm therefore builds a higher number of low-dimensional, smaller and unstable clusters having average number of variables tending to about 5, as displayed in Figure 4e, and average number of points per cluster tending to about 400, as shown in Figure 4f. As seen above, this behavior is associated to the increase of the number of clusters K that tends to about 50.

This evidence shows that the structure of clustering C_{opt} (i.e., number of clusters, cluster centroids, etc.) strongly depends on the structure of the data, hence the (number of) clusters selected and the SAE obtained by this set of clusters are also related to the specific grouping of sensor readings present in our dataset. Since we assume that this grouping structure contains information about drone states, we can conclude that clustering C_{opt} is informative with respect to these states. Figure 4e,f shows also that the silhouette S and the silhouette in subspace SS decrease when the level of noise increases. As expected S tends to 0, while SS starts to grow after a first decrease, with values of about 0.20 for completely random datasets. This behavior is affected by the strong decrease of dimensionality (i.e., number of variables in the subspace) of clusters obtained with addiction of strong noise, which makes SS inappropriate for this kind of comparison.

Analysis of Drone State-Models Defined by Clustering c_{opt}

Our final analysis concerns the clusters (i.e., state-models) belonging to clustering C_{opt}. Table 3 provides a list of those state-models, named M_i, $i \in \{1, \ldots, 24\}$, with corresponding properties, namely, subspace dimension D_i, number of observations O_i, silhouette in the subspace SS_i, precision with respect to states in the water P_{IWi}, out of the water P_{OWi}, upstream navigation P_{USi}, downstream navigation $P_{DSi,}$ navigation with no-stream $P_{NSi,}$ manual drive P_{MDi}, and autonomous drive P_{ADi}. The clusters are sorted by SS. The manual labeling of known states allowed us to identify the clusters with the best mapping to those states. In Table 3 these mappings are highlighted in bold and identified by high precision values (i.e., precision close to 1) for related known states. For instance, the three clusters that best represent the state "out of the water" are M_{14}, M_9 and M_{11} that have precision P_{OWi}, 0.860, 0.842 and 0.831, respectively. The

Table 3. List of clusters (i.e., state-models) in C_{opt} with related properties and performance.

Cluster	D_i	O_i	SS_i	P_{IWi}	P_{OWi}	P_{USi}	P_{DSi}	P_{NSi}	P_{MDi}	P_{ADi}
M_9	16	1198	0.553	0.158	**0.842**	0.000	0.000	1.000	0.677	0.323
M_{19}	2	178	0.510	1.000	0.000	0.000	0.018	0.982	0.337	**0.663**
M_{12}	11	959	0.428	0.994	0.006	0.001	0.000	0.999	0.106	**0.894**
M_{11}	15	1610	0.399	0.169	**0.831**	0.000	0.000	1.000	1.000	0.000
M_5	16	5796	0.366	1.000	0.000	0.000	0.005	0.995	0.076	**0.924**
M_7	3	103	0.306	1.000	0.000	0.014	0.014	0.973	0.728	0.272
M_{21}	5	230	0.254	1.000	0.000	0.057	0.207	0.736	0.917	0.083
M_2	3	103	0.225	1.000	0.000	0.000	0.000	1.000	0.631	0.369
M_{20}	3	129	0.225	1.000	0.000	0.013	0.063	0.924	0.713	0.287
M_{24}	3	99	0.219	1.000	0.000	0.000	0.304	0.696	0.909	0.091
M_{23}	1	64	0.206	1.000	0.000	0.000	0.042	0.958	0.817	0.183
M_{13}	11	309	0.168	0.997	0.003	0.059	0.153	0.788	0.874	0.126
M_6	18	1508	0.120	1.000	0.000	**0.610**	0.022	0.367	0.832	0.168
M_{16}	9	124	0.098	0.734	0.266	0.000	0.000	1.000	0.924	0.076
M_{18}	18	866	0.050	0.777	0.223	0.005	**0.995**	0.000	0.950	0.050
M_{22}	13	1037	0.020	0.972	0.028	0.000	0.080	0.920	0.752	0.248
M_{15}	15	712	0.005	1.000	0.000	0.133	0.148	0.719	0.960	0.040
M_4	10	602	−0.016	1.000	0.000	0.042	0.237	0.721	0.847	0.153
M_8	16	943	−0.018	0.992	0.008	0.004	0.049	0.947	0.783	0.217
M_3	17	1887	−0.037	0.422	0.578	0.000	0.000	1.000	0.932	0.068
M_{14}	14	1049	−0.038	0.140	**0.860**	0.053	0.000	0.947	1.000	0.000
M_{10}	14	413	−0.065	0.562	0.438	0.000	0.053	0.947	0.922	0.078
M_1	1	144	−0.079	1.000	0.000	0.018	0.018	0.963	0.507	0.493
M_{17}	6	118	−0.094	1.000	0.000	0.189	0.149	0.662	0.873	0.127

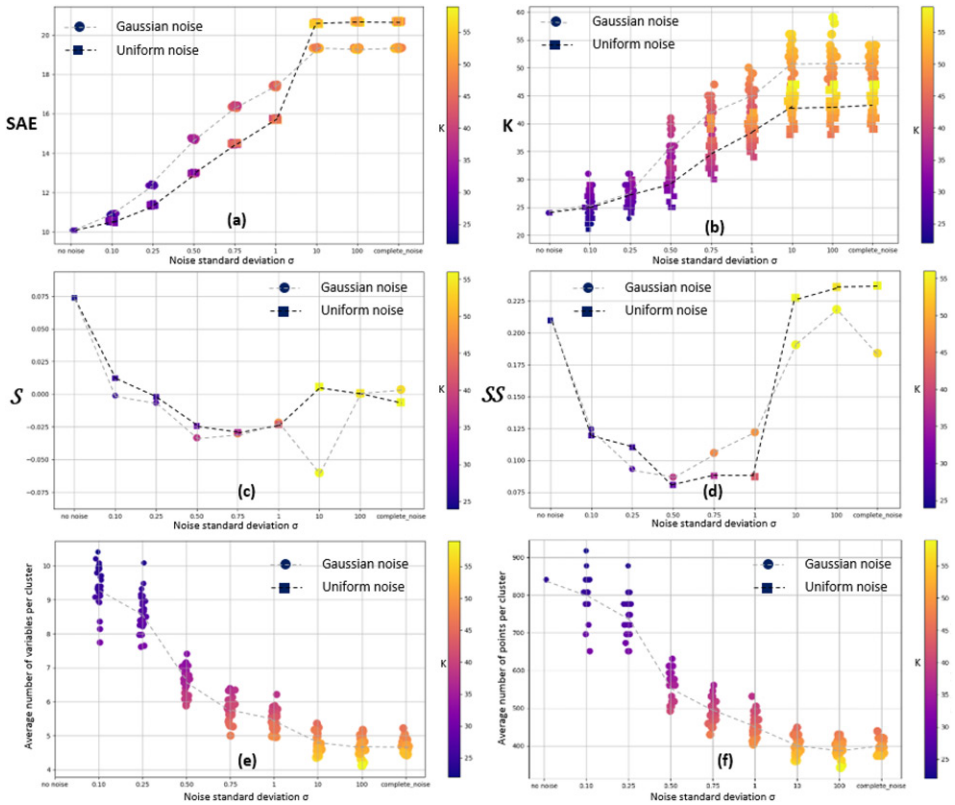

Figure 4. Sensitivity of clustering performance to noise. x-axes represent the level of noise (in terms of standard deviation) and the y-axes represent different performance measure.

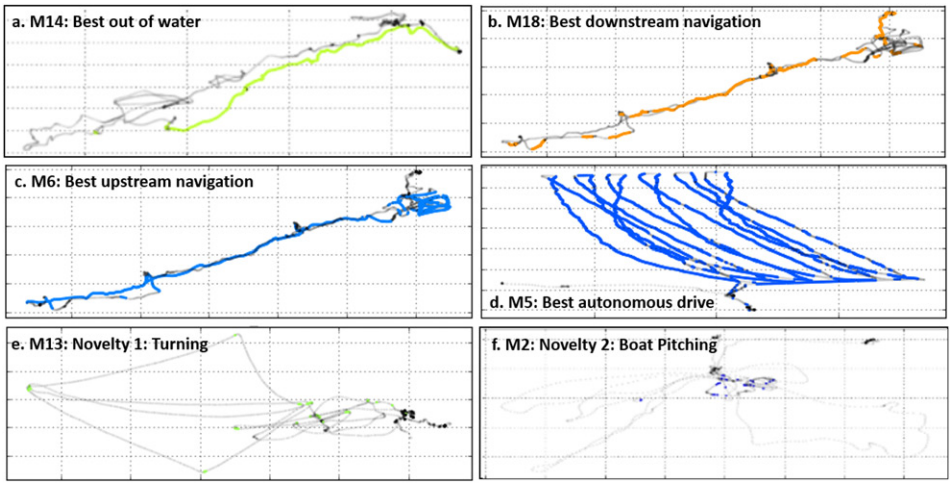

Figure 5. Geo-localization of six clusters (i.e., states). The caption inside each map provides an interpretation for the state represented by the clusters.

geo-localization of cluster M_{14} is partially displayed in Figure 5a (this map focuses only on experiment ESP2, but the cluster has points also in every other experiment) and shows that the cluster actually represents situations in which the drone is out of the water.

It is interesting to focus on four specific clusters representing known states and two states representing unknown states. The last ones demonstrate the novelty detection capabilities of the proposed approach. State model M_{18} has high precision on downstream navigation (see $P_{DSi} = 0.995$ in Table 3) and it actually covers a section of the path related to downstream navigation, as displayed in Figure 5. State-model M_6 is the best cluster for upstream navigation ($P_{DSi} = 0.610$) and it clearly covers the opposite direction of the path previously associated to downstream navigation (see Figure 5c). The main variables of centroids M_{18} and M_6, namely, signal to propellers, enable a clear interpretation of these models, since they have low values for M_{18} (downstream navigation requires low power) and high values for M_6 (upstream navigation requires high power). State-model M_5 has high precision for autonomous drive ($P_{ADi} = 0.924$), it actually corresponds to a long section of the autonomous campaign performed in experiment ITA1 (see Figure 5d), and the most informative variables (selected by SubCMedians) of the corresponding centroid are the signal to propellers, with very small values, that are typical of the stable controller which manages autonomous drive.

Regarding the discovery of novel states, we discuss two clusters of interest. Cluster M_{13}, displayed in Figure 5e, identifies (in a completely unsupervised way) the left curves performed by the drone, and its centroids correspondingly have an high positive value for signal to propeller 0 (left propeller) and a high

negative value for signal to propeller 1 (right propeller). Cluster M_2, displayed in Figure 5f, identifies a state with very high variation of dissolved oxygen which can be possibly associated to situations of "boat pitching", where the sensor of dissolved oxygen is quickly pulled off the water by waves or a swift movement of the drone. Although this state-model needs further investigation, it is of strong interest for data filtering.

Conclusions

The sensitivity analysis presented in this paper allows to identify a subspace clustering model that outperforms the one generated in previous work. Moreover, it provides useful knowledge about the relationships between SubCMedian parameters and clustering performance in the specific context of the considered dataset. Finally, it defines a baseline that allows to evaluate the significance and the informativeness of the proposed model by comparing it with models generated from noisy data. Ongoing research on this topic focuses on the comparison of the performance of other clustering methodologies and the analysis of the effect of different performance indices on the identification of drone states of interest.

References

Abdallah, Z. S., M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. 2012. CBARS: Cluster based classification for activity recognition systems. In *Proc. AMLTA*, 82–91. Berlin: Springer.

Arbelaitz, Olatz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. 2013. An extensive comparative study of cluster validity indices. *Pattern Recognition* 46 (1): 243–56. doi:10.1016/j.patcog.2012.07.021.

Barták, R., and M. Vomlelová. 2017. Using machine learning to identify activities of a flying drone from sensor readings. In *Proc. Flairs,* 436–41. AAAI Press.

Bishop, C. M. 2006. *Pattern recognition and machine learning (information science and statistics)*. Secaucus, NJ, USA: Springer-Verlag, Inc.

Bottarelli, L., M. Bicego, J. Blum, and A. Farinelli. 2019. Orienteering-based informative path planning for environmental monitoring. *Engineering Applications of Artificial Intelligence* 77:46–58. doi:10.1016/j.engappai.2018.09.015.

Castellini, A., G. Beltrame, M. Bicego, J. Blum, M. Denitto, and A. Farinelli. 2018a. Unsupervised activity recognition for autonomous water drones. In *Proceedings of the 33rd Symposium on Applied Computing, SAC*, 840–2. ACM. doi:10.1145/3167132.3167396

Castellini, A., G. Beltrame, M. Bicego, D. Bloisi, J. Blum, M. Denitto, and A. Farinelli. 2018b. Activity recognition for autonomous water drones based on unsupervised learning methods. In *Proc. AIRO - AI*IA*., 16–21. CEUR

Castellini, A., F. Masillo, M. Bicego, D. Bloisi, J. Blum, A. Farinelli, and S. Peigner. 2019. Subspace clustering for situation assessment in aquatic drones. In *Proceedings of the 34th Symposium on Applied Computing, SAC*., 930–7. ACM. doi:10.1145/3297280.3297372

Castellini, A., D. Paltrinieri, and V. Manca. 2015. MP-GeneticSynth: Inferring biological network regulations from time series. *Bioinformatics* 31: 785–787. doi:10.1093/bioinformatics/btu694

Chen, L., J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. 2012. Sensor-based activity recognition. *IEEE Transaction on Systems, Man, and Cybernetics, Part C* 42 (6):790–808.

Endsley, M. R. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37 (1):32–64. doi:10.1518/001872095779049543.

Farinelli, A., D. Nardi, R. Pigliacampo, M. Rossi, and G. P. Settembre. 2012. Cooperative situation assessment in a maritime scenario. *International Journal of Intelligent Systems* 27 (5):477–501. doi:10.1002/int.21532.

Fox, E. B., E. B. Sudderth, M. I. Jordan, and A. S. Willsky. 2008. An HDP-HMM for systems with state persistence. In *Proc. ICML*., 312–9. ACM

Hallac, D., S. Vare, S. Boyd, and J. Leskovec. 2017. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proc. ACM SIGKDD*, 215–23. ACM

Kaelbling, L. P., and T. Lozano-Perez. 2013. Integrated task and motion planning in belief space. *The International Journal of Robotics Research* 32 (9-10):1194–227. doi:10.1177/0278364913484072.

Kim, E., S. Helal, and D. Cook. 2010. Human activity recognition and pattern discovery. *IEEE Pervasive Computing* 9 (1):48–53. doi:10.1109/MPRV.2010.7.

Kriegel, H. P., P. Kröger, and A. Zimek. 2009. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data* 3 (1):1:1–:58. doi:10.1145/1497577.1497578.

Madeira, S. C., and A. L. Oliveira. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1 (1): 24–45. doi:10.1109/TCBB.2004.2.

Montanez, G., S. Amizadeh, and N. Laptev. 2015. Inertial hidden Markov models: Modeling change in multivariate time series. In *Proc. Aaai.*, 1819–25. AAAI

Parsons, L., E. Haque, and H. Liu. 2004. Subspace clustering for high dimensional data: A review. *Acm Sigkdd Explorations Newsletter* 6 (1):90–105. doi:10.1145/1007730.1007731.

Peignier, S., C. Rigotti, A. Rossi, and G. Beslon. 2018. Weight-based search to find clusters around medians in subspaces. In *Proc. SAC*, 471–80. ACM.

Sim, K., V. Gopalkrishnan, A. Zimek, and G. Cong. 2013. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery* 26 (2):332–97. doi:10.1007/s10618-012-0258-x.

Trabelsi, D., S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. 2013. An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *IEEE Transactions on Automation Science and Engineering* 10 (3): 829–35. doi:10.1109/TASE.2013.2256349.

Vail, D. L., M. M. Veloso, and J. D. Lafferty. 2007. Conditional random fields for activity recognition. In *Proc. AAMAS*, 235:1–235:8. ACM