Subspace clustering for situation assessment in aquatic drones

Alberto Castellini
Verona University
alberto.castellini@univr.it

Francesco Masillo
Verona University
francesco.masillo@studenti.univr.it

Manuele Bicego
Verona University
manuele.bicego@univr.it

Domenico Bloisi
Verona University
domenico.bloisi@univr.it

Jason Blum
Verona University
jason.blum@univr.it

Sergio Peigner
Université de Lyon, INSA-Lyon, INRA
sergio.peignier@insa-lyon.fr

Alessandro Farinelli
Verona University
alessandro.farinelli@univr.it

ABSTRACT

Accepted version of the manuscript. Please refer to https://dl.acm.org/citation.cfm?id=3297372 for the published version. We propose a novel methodology based on subspace clustering for detecting, modeling and interpreting aquatic drone states in the context of autonomous water monitoring. It enables both more informative and focused analysis of the large amounts of data collected by the drone, and enhanced situation awareness, which can be exploited by operators and drones to improve decision making and autonomy. The approach is completely data-driven and unsupervised. It takes unlabeled sensor traces from several water monitoring missions and returns both a set of sparse drone state models and a clustering of data samples according to these models. We tested the methodology on a real dataset containing data of six different missions, two rivers and four lakes in different countries, for about 5.5 hours of navigation. Results show that the methodology is able to recognize known states "in/out of the water", "upstream/downstream navigation" and "manual/autonomous drive", and to discover meaningful unknown states from their data-based properties, enabling novelty detection.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**;

KEYWORDS

Situation assessment, activity recognition, subspace clustering, autonomous vessels, aquatic drones, water monitoring, unsupervised learning, model interpretability, sensor data, time series analysis.

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297372

ACM Reference Format:
Alberto Castellini, Francesco Masillo, Manuele Bicego, Domenico Bloisi, Jason Blum, Sergio Peigner, and Alessandro Farinelli. 2019. Subspace clustering for situation assessment in aquatic drones. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus,* Theo D'Hondt and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3297280.3297372

1 INTRODUCTION

Autonomous robots are nowadays used in various application domains, such as, surveillance, monitoring and rescuing [12]. These intelligent systems are able to collect large amounts of information, providing crucial support to human operations. Aquatic drones are increasingly used in this context for autonomous monitoring of catchments, in which robotic boats must navigate rivers and lakes to acquire real-time data concerning important water parameters, such as dissolved oxygen and electrical conductivity. While human operators are usually involved in such data collection activities, direct tele-operation of the drones is often not an option for an entire mission, hence autonomous navigation capabilities are required. In particular, aquatic drones must maximize the information content of data acquired during missions [6] while adapting to anomalous conditions of internal devices and environment, using a minimal number of sensors to reduce the cost of equipment.

A key factor for the success of autonomous data acquisition campaigns is *mission awareness* [4, 11], which is composed of three main elements: knowledge of mission objectives, internal self-situational awareness, and external self-situational awareness. In this work we specifically focus on the problem of detecting, modeling and interpreting aquatic drone states from a data-driven point of view, which is an aspect of self-situational awareness. In other words, we aim at developing interpretable models of drone states from traces of sensor data acquired during water-monitoring missions, by means of statistical learning methods. Maintaining such a set of drone state models is important for two reasons, first it can strongly improve the autonomy of the drone by providing key information to *online decision making* [2, 18], second it can support *offline data analysis* by improving the extraction of knowledge from the large dataset of available sensor traces.

As drone states in this work we intend classes of observations having similar statistical/informational properties within the entire dataset of sensor traces. Unsupervised methods, such as clustering

and time series segmentation, are ideal tools for detecting such kind of patterns since they usually optimize internal performance measures [15]. 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 which is often expansive, time consuming, and even impracticable in some cases. Moreover, the data-driven models generated by some of these methods provide abstract descriptions of drone states that can be interpreted and validated by experts, and incrementally updated as new data become available. Finally, new states could also be discovered from sets of observations with homogeneous and statistically coherent patterns, promoting the process of novelty detection [30].

The literature provides several methods for clustering and segmenting multivariate time series which mainly differ from each other in the assumptions they make on data or model properties. Some of these techniques are also used in contexts similar to ours, such as sensor-based human activity recognition [9, 21], in which sensors are used to acquire data about human movements with the aim to create computational activity models and infer human activities. To the best of our knowledge the main techniques used in this field are: k-means [1, 3, 5, 21, 32], Gaussian mixture models (GMM) [3, 5, 21, 32], hierarchical clustering [3, 5, 21], hidden Markov models (HMMs) [3, 13, 19, 26, 32], conditional random fields (CRFs) [33], Markov random fields [14] and change-point detection methods [22]. However, only a very few of them [3, 14] were applied to data from drones or vehicles and none of them on aquatic drones. The usefulness of some of these techniques in discovering activities in the aquatic drone scenario was first investigated in [7, 8] where performance of k-means, GMM, HMM and affinity propagation were compared on a real dataset. Some peculiarities of aquatic drones datasets make activity recognition in this context very challenging. In particular, these kinds of data are very noisy, since they come from several sources, and are strongly influenced by unstructured and diversified environments (e.g., rivers and lakes in different parts of the world). Moreover, aquatic drones collect data related to both movement and water parameters, and both sources of information can be used to assess the state of the drone.

In this work we use *subspace clustering* to improve the performance of standard clustering methods in terms of both goodness of fit (here measured by clustering silhouette) and cluster interpretability. Subspace clustering is an adaptation of clustering for high dimensional data [28]. This approach is recognized as more general than traditional clustering, since it tackles two different problems simultaneously: detecting clusters in the dataset and searching a relevant subspace for each cluster. Different approaches have been proposed to address this problem, using different paradigms [31]. Three major families of approaches have been identified in the literature. The *cell-based* approach which searches hyper-rectangular clusters, that contain more than a given number of objects. The *density-based* approach aims at detecting arbitrarily shaped dense groups of objects, separated from other groups by low density zones. The *clustering-oriented* approach tends to form hyper-spherical shaped clusters, using distance-based similarity measures, and some properties of the targeted clustering model (e.g., number of clusters). The reader is referred for instance to [20, 27, 28, 31] for reviews and comparisons of state-of-the-art methods and major categories, and to [10, 23, 25, 29] for some recent subspace clustering algorithms proposed so far in the literature.

We use, in particular, a recent center-based technique called *Sub-CMedians* [29], on a dataset containing suitable variables extracted from sensor traces of 6 concatenated missions. The methodology generated 26 state models that we prove to contain information about meaningful situations, such as upstrem/downstream navigation and manual/autonomous drive. Then we analyze some of the models by means of a novel software tool called *eXplainable Modeling* (XM) and interpret their parameters (i.e., cluster centroids) and properties (e.g., geolocation and distributions), showing that the analysis framework has enhanced capabilities in terms of interpretability and novelty detection.

The main contributions of this paper to the state-of-the-art can be summarized in the following points:

- we generated sparse models of aquatic drone states by subspace clustering and shown that they have improved goodness-of-fit and interpretability with respect to those computed by standard clustering methods;
- we proved the capability of the proposed methodology to recognize meaningful states of the drone, which motivates its usage in discovering unknown states (i.e., novelty detection).

The following of the manuscript presents the data acquisition platform, the dataset, the subspace clustering methodology and the XM tool in Section 2. The results and related performance are analyzed in Section 3, and some conclusions and directions for future work are described in Section 4.

2 MATERIAL AND METHODS

In this section we formalize the problem, describe the available dataset and introduce the subspace clustering method employed to generate the results.

2.1 System overview and problem definition

Data acquisition campaigns are performed by the aquatic drones shown in Figure 1. Drones are equipped with sensors for GPS position, water properties (i.e., temperature, dissolved oxygen and electrical conductivity), commands to propellers and battery voltage. Drone operators define paths by setting waypoints on a map in a tablet (autonomous drive) or they manually drive the drone using an RC controller. Sensor traces are stored in log files which are preprocessed to obtain a matrix of multivariate time series (displayed in the center of Figure 1), where rows represent variables (i.e., sensor signals) and columns represent time (in seconds). The methodology proposed in this work aims at detecting different drone states from that data matrix, and to generate interpretable mathematical models of those states. States are represented by colored square labels in the picture, since they involve subsets of time points (columns) and subsets of variables (rows). Interpreting states in terms of situations in which the drone finds itself is useful for improving both data analysis and decision making.

2.2 Dataset

Sensor traces of six missions were analyzed. Table 1 shows their duration, number of samples and type of catchment (i.e., river or lake). We concatenated these traces, obtaining a single dataset with

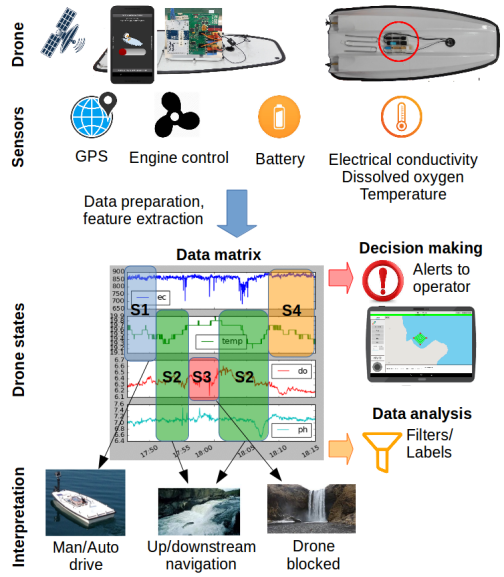


Figure 1: Main elements of the proposed situation assessment system for autonomous water drones.

20187 observations and about 5.6 hours of navigation (the sampling interval is of 1 second). Variables available in the raw datasets were time, latitude, longitude, altitude, speed, electrical conductivity, dissolved oxygen, temperature, battery voltage, heading, acceleration and command to propeller 0 and 1. Using only raw sensor data for generating state models has the drawback to often generate data-splitting due to naive differences in environmental parameters among different missions. For instance, since the dissolved oxygen assumes quite different values in different missions, the clustering methodology tends to generate independent models for each mission. On the other hand, our aim is to discover cross-mission states, namely states that are possibly observable in several missions. Feature extraction was employed to this end. We removed some of raw-sensor variables and generated two kinds of new variables, namely, *moving means and standard deviations* over a sliding windows of 10 seconds and *variations* between couples of consecutive observations. The list of 27 variables in the final (CONCAT) dataset is reported in Table 2. As suggested in [29] Z-score standardization was performed on each variable and the obtained CONCAT dataset (see Table 1) was provided to the SubCMedians algorithm.

#	Name	Samples	Duration	Lake/River
1	ESP2	2814	47'	R
2	ESP5	3601	60'	R
3	ESP4	2374	39'	L
4	GARDA3	2451	40'	L
5	ITA1	7243	121'	L
6	ITA6	1704	28'	L
-	CONCAT	20187	335'	-

Table 1: Analyzed datasets by mission.

Symbol	Description
s, v, a	Instantaneous speed, voltage, acceleration
m_0, m_1	Instantaneous signal to propeller 0 and 1
$\bar{s}$, $\bar{v}$, $\bar{a}$	Moving average mean of speed, voltage, acceleration
$\bar{m_0}$, $\bar{m_1}$	Moving average mean of signal to propeller 0 and 1
$\hat{s}$, $\hat{v}$, $\hat{a}$	Moving average std of speed, voltage, acceleration
$\hat{ec}$, $\hat{do}$, $\hat{T}$	Moving average std of electrical conductivity, dissolved oxygen, temperature
$\hat{m_0}$, $\hat{m_1}$	Moving average std of signal to propeller 0 and 1
$\hat{h}$	Moving average std of heading
$\widetilde{s}$, $\widetilde{a}$, $\widetilde{v}$	Variation of speed, voltage, acceleration
$\widetilde{m_0}$, $\widetilde{m_1}$	Variation of signal to propeller 0 and 1
$\widetilde{ec}$, $\widetilde{do}$, $\widetilde{h}$	Variation of electrical conductivity, dissolved oxygen, temperature

Table 2: Variables in dataset CONCAT.

2.3 Drone states

By drone states here we mean situations or activities whose awareness can improve the autonomy of the drone or the process of analysis of the data that it collects. Since our approach is data-driven, we can detect only states that influence somehow the sensor traces. In order to check if there is a real connection between meaningful states and modifications of data properties, we manually labeled our concatenated dataset according to seven simple but meaningful drone states, namely, in water (IW), out water (OW), upstream navigation (US), downstream navigation (DS), no water current (NS), manual drive (MD), and autonomous drive (AD). Then we checked if our methodology was able to identify such known states. To this end we computed cluster purity, a performance measure defined in Section 2.6, that evaluates the extent to which a cluster represents a single drone state. The positive result on this test (see Figure 2.a) proves the capability of our method to detect meaningful states and supports our confidence in its capabilities of novelty detection, which are also confirmed by results presented in Section 3.

2.4 Subspace clustering by SubCMedians

The methodology for drone state identification presented in this work relies on a recent center-based subspace clustering technique called SubCMedians [29]. This algorithm is based on a K-medians paradigm [17] and it aims at clustering data points around suitable candidate centers, each one described in its own subspace. In the context of the center-based subspace clustering paradigm, a cluster center defined in a given subspace, represents informally a "summary" of the cluster points, its subspace contains the most informative variables for the given cluster, and the center 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.

The clustering problem tackled by SubCMedians can be stated more formally as follows. Let a set of points $X = \{x_1, x_2, \dots\}$ denote a dataset, and each point $x \in X$ is described in $\mathbb{R}^D$ by D variables (point coordinates). Here, each variable is represented by a number ranging from 1 to D, and the set of all variables is denoted $\mathcal{D} = \{1, \dots, D\}$. Let $\mathcal{M}$ denote the set of centers built by SubCMedians, such that each center $m_i \in \mathcal{M}$ is defined in its own

subspace (i.e., subset of variables) $\mathcal{D}_i \subseteq \mathcal{D}$. The size of a model $\mathcal{M}$, is defined as the sum of the number of variables contained in the subspaces of the model centers: $Size(\mathcal{M}) = \sum_i |\mathcal{D}_i|$. This value is intuitively interpreted as the "level of detail" of the model.

In SubCMedians, the distance $dist(x, m_i)$ 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: $dist(x, m_i) = \sum_{d \in \mathcal{D}_i} |x_d - m_{i,d}| + \sum_{d \in \mathcal{D} \setminus \mathcal{D}_i} |x_d - \mu_d|$, with $m_{i,d}$ the coordinate of m_i along variable d, and with μ_d the mean of the coordinates of all points in X along d. For a dimension $d \notin \mathcal{D}_i$, the intended meaning is that, along d, the points of the cluster are simply distributed around the barycenter of the full dataset. The distance between each point $x \in X$ and its closest center $m_i \in \mathcal{M}$ is called the Absolute Error $AE(x, \mathcal{M}) = min_{m_i \in \mathcal{M}} dist(x, m_i)$. The goal of SubCMedians is to build a set of centers $\mathcal{M}$, so as to minimize the Sum of Absolute Errors $SAE(X, \mathcal{M}) = \sum_{x \in X} AE(x, \mathcal{M})$, and such that $Size(\mathcal{M}) \leq SD_{max}$, where SD_{max} is a parameter denoting the maximum Sum of Dimensions used in $\mathcal{M}$ to describe all its centers (the number of centers is not constrained).

In practice SubCMedians updates iteratively the coordinates and the subspaces of its centers, using a stochastic hill climbing technique. Moreover SubCMedians 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 $NbIter$. In [29], the authors provide easy default parameter setting guidelines, that allowed SubCMedians to obtain competitive results compared to state-of-the-art algorithms on benchmark datasets. Following these guidelines the user only needs to provide a suggested number of clusters, termed $NbExpClust$, from which the other parameters are computed. The actual number of clusters is then selected automatically by the algorithm at runtime. We used $NbExpClust = 10$, since the expected number of clusters was around 10, and slightly changed the standard parameter settings proposed in [29] to $SD_{max} = D \times NbExpClust = 270$, $N = 50 \times NbExpClust = 500$ and $NbIter = 20 \times SD_{max} \times NbExpClust = 54000$. The algorithms needs less than one minute to compute the clustering on a Intel CORE i7 with 8GB of RAM. We run the algorithm 20 times and then selected the result with the best clustering silhouette (this performance measure is described in the next section). The procedure turned out to be sufficient to build a satisfactory subspace clustering model, as shown in Section 3.

2.5 Standard clustering methods

The performance of SubCMedians is compared with that of the standard clustering methods k-means and Gaussian Mixture Models (GMMs). In both cases we set the number of clusters to 26, namely the number of clusters detected by SubCMedians. In k-means we used the Euclidean distance $\| \cdot \|^2$ and re-initialized the algorithm 300 times, then we took the model with lowest residual sum of squares. In GMM we re-initialized the algorithm 300 times and the model with maximal log-likelihood was used. Initial component means were generated by the k-means algorithm, initial mixing proportions were set to uniform and initial covariance matrices were defined diagonal and initialized according to the obtained k-means result, namely by computing the variance for each variable in a particular cluster. Parameter learning was performed by computing a maximum likelihood solution through the Expectation-Maximization (EM) algorithm. The full covariance matrix was used and a maximum of 100 iterations were executed. Inference was performed, given a trained GMM, by assigning each data point to the component (i.e., cluster) with maximum posterior probability. The performance of the three algorithms is shown in Table 3.

2.6 Performance evaluation and variable ranking

To assess the performances of our method we employed two measures, *purity* and *silhouette* [15, 24]. The former is an external criterion, which compares the result of the clustering with a ground truth. In our case this is used to assess the capability of the proposed framework to detect known situations. The latter is an internal index, which measures the goodness of the clustering without using the ground truth but assessing only the compactness of the clusters. More in detail, *purity* is computed by formula $\mathcal{P}(C) = \frac{1}{N} \sum_{k \in K} \max_{d \in D} |k \cap d|$, where C is a clustering, N is the total number of points, K is the set of clusters and D is the set of classes. Purity values close to $1/|D|$ represent clusterings very fragmented in different labels, while purity values close to 1 identify clusterings having almost one label for each cluster. The *silhouette* of the i-th data point is computed as $\mathcal{S}(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$, 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. Values range from -1 to 1 where high values indicate points belonging to perfectly compact and separated clusters and low values indicate clustering with mixed clusters.

To improve the visual interpretability of the state models, we sorted their variables by decreasing symmetrical uncertainty (SU) [16]. SU is a measure of relevance of a variable f_r with respect to a clustering solution I and can be computed as $SU(f_r, I) = 2(\frac{IG(f_r \mid I)}{H(f_r)+H(I)})$ where $H(I)$ is the entropy of the clustering labels and $IG(f_r \mid I)$ is the information gain that is computed as $IG(f_r \mid I) = H(f_r) - H(f_r \mid I)$, and $H(f_r)$ is the entropy of variable f_r and $H(f_r \mid I)$ is the conditional entropy of f_r given I. A value 1 of SU indicates that the variable f_r is completely related to clustering I while a value 0 means that the variable f_r is absolutely irrelevant.Finally to check cluster coherence we also used t-SNE [34] a dimensionality reduction method based on Stochastic Neighbor Embedding that produces clear 2D visualizations (as scatter plots) of the multidimensional datasets.

2.7 The eXplainable Modeling (XM) tool

The software *eXplainable Modeling* (XM) is a free Python tool which supports the processes of data analysis and model generation. The current version XM1.5.1 provides the following kinds of visualization of data: time series, heatmaps, 2D/3D scatter plots, boxplots, histograms, geolocation and t-SNE. Moreover, it enables to generate clusterings by k-means, GMM and SubCMedians. The main

advantage of XM is the enhanced interpretability of data and related models that it enables, due to its integrated and interactive visualization modes. All the charts displayed in Section 3 were generated by this tool.

3 RESULTS

In this section we analyze the performance and the clusters (representing drone state models) of the clustering generated by SubCMedians, and prove the ability of the proposed methodology to recognize known states and discover new ones.

3.1 Best clustering and related performance

The performance of the best clustering computed by SubCMedians and the related best clusterings generated by k-means and GMM using the same number of clusters are displayed in Table 3. This table shows that the silhouette of the SubCMedians clustering is slightly higher than that of k-means, and much better than that of GMM. Moreover, the key property of the SubCMedians clustering, which motivates its usage, is model sparsity and related enhancements of model interpretability. This is observable in the last two columns of Table 3 that display both the total size of the clustering model (i.e., 247 for SubCMedians and 702 for k-means and GMM) and the average number of variables for each cluster (9.5 in SubCMedians and 27 in k-means and GMM).

Table 3: Properties and performance of the best clusterings generated by SubCMedians, k-means and GMM.

Method	Silhouette	K	Size($\mathcal{M}$)	avg(#v)
SubCMedians	0.155	26	247	9.5
KM	0.151	26	702	27
GMM	-0.076	26	702	27

The list of clusters (i.e., drone state-models) generated by SubCMedians is reported in Figure 2.a. For each model, the table shows: the number of selected variables $\mathcal{D}_i$, the number of observations belonging to the cluster O_i, the cluster silhouette S_i and the cluster purity in relation to the 7 situations described in Section 2.3 (i.e., in-water P_{IWi}, out-water P_{OWi}, downstream P_{DSi}, upstream P_{USi}, no-stream P_{NSi}, manual drive P_{MDi}, autonomous drive P_{ADi}), where i represents the cluster index. The clusters are sorted by silhouette, which is available also when no ground truth is known.

We observe that cluster subspaces have between 1 and 20 variables, and each cluster groups together between 20 and 3739 samples. The maximum silhouette is 0.754 for cluster M_{24} and the minimum is -0.245 for cluster M_{21}. In order to show that these clusters have a direct connection to meaningful drone states, we analyze some of the clusters having high purity. In particular we show the relationships between the real geolocation of the known states in the 6 experiments (i.e., the labeling), the geolocation of the related clusters computed by SubCMedians, and the parameters of the clusters.

3.2 Best clusters for state OW

We start with the analysis of cluster M_{22} which has a high purity for state out-water (i.e., $P_{OW} = 0.834$, see Figure 2). The second and third column of Figure 3 show, respectively, where the drone was actually out-water during the 6 experiments (i.e., red point labeling in the maps of the second column) and where the samples belonging to the cluster M_{22} are located (i.e., red points in the paths in the third column). In experiment ESP2 (i.e., first row of Figure 3), for instance, the drone was turned on a lot of time before putting it into the water, this can be seen by the long red line in the map. That state was correctly detected by cluster M_{22} whose samples (i.e., red path in the third column of Figure 3) almost perfectly correspond to the red line in the map in the second and third columns of Figure 3. Similar behaviors can be observed also in the other experiments (e.g., ESP5 and ESP4). We notice that variable electrical conductivity (i.e., the raw signal coming from the sensor) was not part of the set of variables retained, as described in Section 2.2, therefore the recognition of this state is not naive. Moreover, we highlight that automatically discovering this state is of interest for data filtering.

In order to understand the statistical properties that characterize the out-water state, we analyze the parameters of cluster M_{22}, which are listed in Figure 2.b (see the table for model M_{22}). The variables selected by SubCMedians are displayed in the first column, the related coefficients and SU are reported in the second and third column, respectively. Variables are sorted by SU to simplify the analysis and put more informative variables on top. The main properties are a high voltage ($v = 16.844\ V$), null signals to engines ($\bar{m}_0 = \bar{m}_1 = 0.000$), quite low standard deviation of speed ($\hat{s} = 0.167\ m/s$) and null standard deviation of electrical conductivity ($\hat{ec} = 0.000\ S/m$). From these parameters, summarized also in Table 4, we can understand that the drone is at the beginning of the mission, since the battery is completely full, the operator is not providing any signal to the engines, he/she is moving quite steadily (possibly to bring the boat to the water) and the electrical conductivity is completely fixed. The last property is the only one related to the environment and it confirms that the boat is out-water. Also recognizing the different stages of the mission is of great interest for drone autonomy.

Table 4: Models of known states: summary. The main differences among parameters (in bold) are analyzed in the text.

V	OW (M_{22})	US (M_{23})	DS (M_1)	AD (M_{10})	N (M_{24})
v	**16.844**	15.139	15.103	-	-
$\bar{m}_0$	**0.000**	**0.783**	0.000	**0.075**	-
$\bar{m}_1$	**0.000**	**0.604**	0.000	**0.069**	-
$\hat{s}$	**0.167**	0.192	-	**0.050**	-
$\hat{ec}$	**0.000**	-	1.399	2.339	**187.935**
$\hat{v}$	0.006	**0.088**	0.013	**0.003**	0.014
$\hat{m}_0$	0.002	**0.134**	0.044	**0.008**	-
$\hat{m}_1$	0.000	**0.211**	-	**0.022**	-
$\bar{s}$	-	**1.290**	-	**0.455**	-
$\hat{h}$	89.228	**15.887**	17.782	**4.520**	-
$\hat{a}$	-	0.201	0.161	**0.063**	-

3.3 Best clusters for states UN and DN

A similar analysis can be performed on clusters related to upstream and downstream navigation, namely M_{23} and M_1 respectively, which have high purity P_{US} and P_{DS} (see Figure 2.a). The

Cluster	$\mathcal{D}_i$	O_i	Sil $\mathcal{S}_i$	In/Out water $\mathcal{P}_{IWi}$	$\mathcal{P}_{OWi}$	Up/down/no stream $\mathcal{P}_{USi}$	$\mathcal{P}_{DSi}$	$\mathcal{P}_{NSi}$	Auto/Man drive $\mathcal{P}_{MDi}$	$\mathcal{P}_{ADi}$
M_{24}	3	120	0.754	0.733	0.267	0.000	0.000	1.000	0.555	0.079
M_{26}	17	1575	0.590	0.160	**0.840**	0.000	0.000	1.000	1.000	0.000
M_{11}	1	62	0.506	1.000	0.000	0.016	0.049	0.926	0.613	0.387
M_6	17	2833	0.357	0.288	**0.712**	0.000	0.000	1.000	0.888	0.112
M_{18}	1	49	0.324	1.000	0.000	0.000	0.029	0.971	0.612	0.387
M_8	10	319	0.260	0.906	0.094	0.000	0.048	0.952	0.773	0.226
M_{12}	6	216	0.195	0.986	0.014	0.015	0.000	0.984	0.413	**0.587**
M_5	1	20	0.157	1.000	0.000	0.000	0.000	1.000	0.250	**0.750**
M_1	14	794	0.152	0.780	0.220	0.000	**1.000**	0.000	0.940	0.060
M_{23}	18	1571	0.141	1.000	0.000	**0.575**	0.039	0.385	0.840	0.160
M_{17}	13	916	0.116	0.980	0.020	0.000	0.045	0.954	0.698	0.302
M_{15}	20	3739	0.112	0.999	0.001	0.000	0.005	0.994	0.113	**0.887**
M_{13}	7	253	0.110	0.893	0.107	0.043	0.093	0.863	0.898	0.102
M_3	3	167	0.099	0.988	0.011	0.029	0.029	0.942	0.848	0.151
M_{19}	12	517	0.083	0.978	0.021	0.000	0.034	0.966	0.772	0.227
M_{10}	16	2970	0.081	0.998	0.002	0.000	0.008	0.991	0.026	**0.974**
M_{22}	15	1033	0.055	0.165	**0.834**	0.027	0.000	0.973	1.000	0.000
M_7	6	424	-0.008	1.000	0.000	0.033	0.017	0.950	0.669	0.330
M_9	10	416	-0.032	1.000	0.000	0.083	**0.847**	0.069	0.945	0.055
M_2	8	93	-0.042	0.946	0.053	0.013	0.051	0.935	0.977	0.022
M_4	3	56	-0.057	1.000	0.000	0.040	0.080	0.880	0.821	0.178
M_{16}	12	491	-0.071	1.000	0.000	0.136	0.176	0.687	0.988	0.012
M_{25}	7	146	-0.078	1.000	0.000	0.000	0.013	0.987	0.555	0.445
M_{20}	12	624	-0.100	1.000	0.000	0.078	0.117	0.805	0.966	0.034
M_{14}	1	39	-0.188	1.000	0.000	0.000	0.000	1.000	0.462	**0.538**
M_{21}	14	738	-0.245	0.626	0.374	0.007	0.048	0.945	0.905	0.095

(a) Main properties and performance of state-models detected by SubCMedians.

M_1

Var	Coef	SU
v	15.103	0.062
$\bar{v}$	15.103	0.062
m_0	-0.016	0.036
$\hat{a}$	0.161	0.026
m_1	-0.004	0.020
$\bar{m_1}$	-0.000	0.015
$\bar{m_0}$	-0.000	0.014
$\hat{m_0}$	0.044	0.011
$\hat{h}$	17.782	0.007
$\hat{v}$	0.013	0.007
$\bar{s}$	-0.000	0.007
$\hat{ec}$	1.399	0.003
$\widetilde{m_0}$	-0.000	0.002
$\widetilde{h}$	0.350	0.001

Interp.: downstream

M_{10}

Var	Coef	SU
$\bar{m_0}$	0.075	0.125
m_1	0.051	0.104
$\bar{s}$	0.455	0.098
$\bar{m_1}$	0.069	0.096
$\hat{a}$	0.063	0.095
s	0.516	0.094
m_0	0.070	0.092
$\hat{h}$	4.520	0.065
$\hat{m_1}$	0.022	0.056
$\hat{s}$	0.050	0.085
$\hat{m_0}$	0.008	0.055
$\hat{v}$	0.003	0.048
$\widetilde{v}$	-0.000	0.038
$\hat{ec}$	2.339	0.013
$\hat{T}$	-0.000	0.012
$\widetilde{ec}$	-0.000	0.006

Interp.: autonomous drive

M_{22}

Var	Coef	SU
v	16.844	0.063
$\bar{v}$	16.844	0.063
$\bar{m_1}$	-0.000	0.045
m_1	-0.000	0.043
$\bar{m_0}$	-0.000	0.043
m_0	-0.000	0.033
$\bar{s}$	0.167	0.030
$\widetilde{m_1}$	0.000	0.028
$\hat{h}$	89.228	0.020
$\hat{ec}$	-0.000	0.020
$\hat{v}$	0.006	0.019
$\hat{m_1}$	-0.000	0.018
$\hat{m_0}$	0.002	0.017
$\widetilde{v}$	-0.000	0.017
$\widetilde{ec}$	-0.000	0.012

Interp.: out of water

M_{23}

Var	Coef	SU
$\bar{m_0}$	0.783	0.178
$\bar{m_1}$	0.604	0.175
m_0	0.866	0.163
m_1	0.695	0.147
$\hat{v}$	0.088	0.124
$\hat{m_1}$	0.211	0.096
$\bar{s}$	1.290	0.069
$\bar{v}$	15.139	0.064
$\hat{m_0}$	0.134	0.064
s	1.240	0.062
v	15.139	0.061
$\hat{a}$	0.201	0.058
$\hat{s}$	0.192	0.047
$\widetilde{a}$	0.002	0.034
$\hat{h}$	15.887	0.016
$\hat{do}$	0.030	0.005
$\hat{T}$	-0.000	0.003
$\widetilde{do}$	0.003	0.002

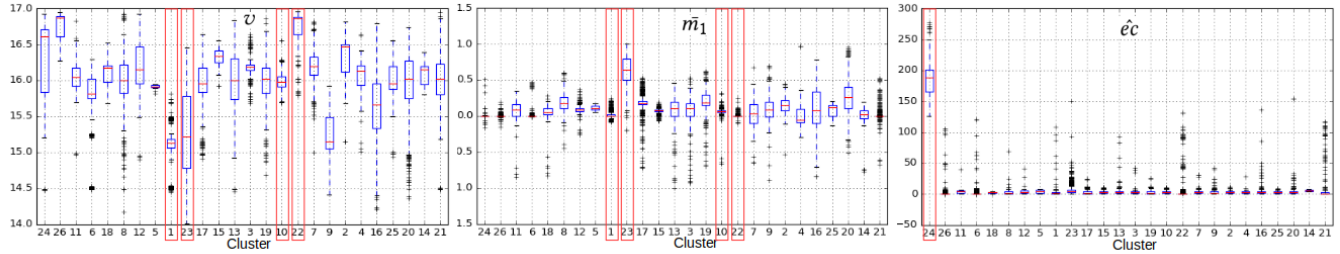
Interpr.: upstream

M_{24}

Var	Coef	SU
$\hat{ec}$	187.935	0.302
m_1	0.078	0.004
$\hat{v}$	0.014	0.002

Interp.: drone put in the water

(b) Detail on specific state-models: parameters and symmetrical uncertainty (SU). M_1, M_{10}, M_{22} and M_{23} are models of known states, M_{24} is a newly discovered state.



(c) Box-plots of a few variables involved in the clusters displayed in point b

Figure 2: State-models generated by SubCMedians: performance and parameters.

fourth column of Figure 3 shows the labeling for state upstream/-downstream/no-stream, while the fifth and sixth columns display the geolocation of the samples belonging to cluster M_{23} and M_1, respectively. The maps of experiments ESP2 and ESP5 (see first and second rows of Figure 3) show a very good match between the true state (i.e., green for upstream and blue for downstream) and the related cluster. We remind that samples in which the label was not available were not used to compute the purity. It is interesting to notice that the lower purity of cluster M_{23} for state upstream (i.e., $P_{US} = 0.575$) can be explained by the (false positive) red points in the paths of experiments ESP4, GARDA3 and mainly ITA6. These points correspond to situations in which the drone drove in lakes at very high engine power, which have very similar statistical properties to upstream navigations and for this reason they were mixed up into the same cluster.

The main properties of cluster M_{23} (see Figure 2.b) are: high signal to engine (i.e., $\bar{m_0} = 0.783$ and $\bar{m_1} = 0.604$, which have also high SU that indicates that the operator provided full power to the boat, high standard deviation of voltage (i.e., $\hat{v} = 0.088\ V$) which is typical when the battery level decreases more sharply, high standard deviations of signal to engines ($\hat{m_1} = 0.211$ and $\hat{m_0} = 0.134$) which is typical of manual drive at high speed, high mean speed (i.e., $\bar{s} = 1.290\ m/s$), low standard deviation of heading ($\hat{h} = 15.887°$) which corresponds to straight movement of the boat. All these properties seem to be reasonable for identifying the upstream navigation state, in which the boat needs much power to contrast the water flow. The differences between models of upstream navigation (i.e., M_{23}) and downstream navigation (i.e., M_1) can be visualized in Figure 2.b and 2.c, and summarized in Table 4.

3.4 Best clusters for state AD

As a last case study, we analyze cluster M_{10} which has the highest purity for autonomous drive (i.e., $P_{AD} = 0.974$). The geolocation of the samples belonging to the cluster is shown in the eighth column of Figure 3 and the ground truth (i.e., green for manual drive, blue for autonomous drive, red for non labeled samples) is displayed in the seventh column. The cluster mainly covers a large part of experiment ITA1, which was completely performed by autonomous drive. The other part of this path is covered by cluster M_{15} which has also a large purity P_{AD}.

Cluster M_{10} is characterized by low and very specific values of signal to engines (i.e., $\bar{m_0} = 0.075$, $\bar{m_1} = 0.069$, $\hat{m_0} = 0.008$, $\hat{m_1} = 0.022$), average/low speed (i.e., $\bar{s} = 0.455\ m/s$), very small standard deviation of acceleration, speed, heading and voltage (i.e, $\hat{a} = 0.063\ m/s^2$, $\hat{s} = 0.050\ m/s$, $\hat{h} = 4.520°$, $\hat{v} = 0.003\ V$). All these

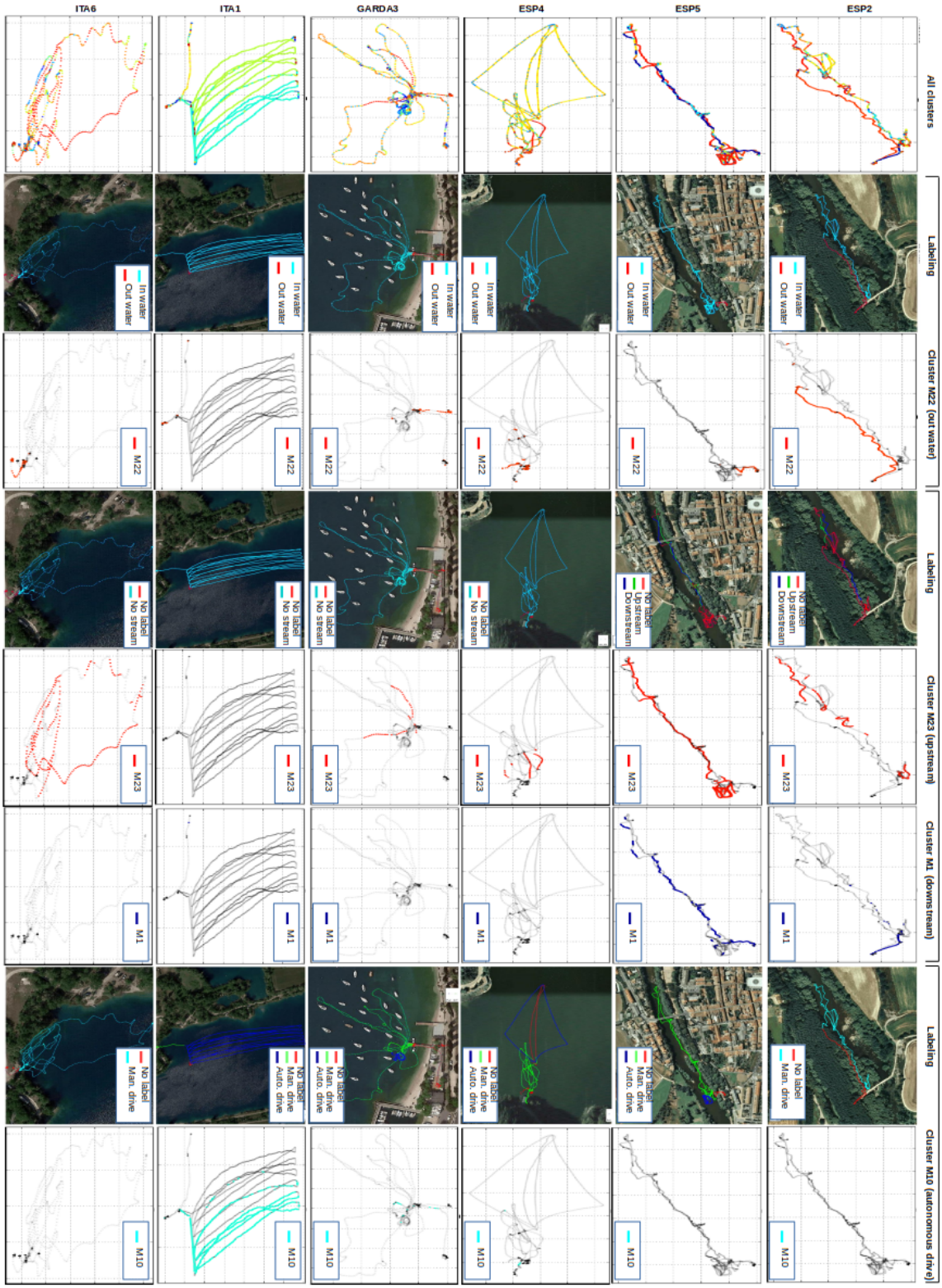


Figure 3: Geolocation of state-models computed by SubCMedians, with specific focus on some models having high purity.

properties, summarized also in Table 4, identify a very stable style of navigation which is typical of autonomous drive.

3.5 Novelty detection

To prove the novelty detection capabilities of our method, we finally analyzed all the other clusters and tried to use the qualitative (i.e., based on chart visualization) and quantitative (i.e., based on internal performance measures of clusters and information measures of variables) tools of XM to get meaningful interpretations of related state-models. The cluster with higher silhouette (namely 0.754), for instance, is M_{24} which is mainly characterized by a very high standard deviation of electrical conductivity ($\hat{ec} = 187.935\ S/m$, see model parameters in Figure 2.b and the third box plot on the right hand side of Figure 2.c). This cluster includes only 120 samples which correspond to the specific instants in which the boat was put or recovered into/from the water. Other states of interest were discovered that cannot be described here for space limitations.

4 CONCLUSION AND FUTURE WORK

We proposed a first complete framework for generating and analyzing state models of water monitoring drones from real sensor data. The methodology was tested on a dataset containing data from six missions performed in both rivers and lakes. Resulting models and data clustering were analyzed showing that known states were recognized and the sparsity of the related models improved their interpretability. This first achievement motivated our analysis of the other models in which we interestingly discovered meaningful unknown states in a novel detection perspective.

This line of research can be extended in several applicative and theoretical directions. First, we want to scale the approach to datasets containing dozens or a hundred of missions possibly with online methods able to update our models when new data are available instead of recomputing them from scratch. Second, new clustering performance measures and stability approaches should be tested to statistically prove (or rank) the significance of each model and of related variables. Third, the capability of detecting more complex situations, such as the presence of dangerous waves or wind, should be tested, which needs an expansive labeling stage. Fourth, we aim to investigate the connection between the feature extraction process and the state detection performance since different variables could enable to discover different states. Finally, a goal-oriented way to rank states according to their importance should be identified, where states gain more importance if their knowledge improves the drone capability to reach its mission objectives.

REFERENCES

[1] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. 2012. *CBARS: Cluster Based Classification for Activity Recognition Systems.* Springer Berlin Heidelberg, 82–91.

[2] A. Asperti, D. Cortesi, and F. Sovrano. To appear. Crawling in Rogue's dungeons with (partitioned) A3C. In *The 4th Int. Conf. Machine Learning, Optimization and Data science (LOD 2018), Volterra, Italy.* Springer.

[3] R. Barták and M. Vomlelová. 2017. Using Machine Learning to Identify Activities of a Flying Drone from Sensor Readings. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017,* V. Rus and Z. Markov (Eds.). AAAI Press, 436–441.

[4] V. Berenz, F. Tanaka, and K. Suzuki. 2012. Autonomous battery management for mobile robots based on risk and gain assessment. *Artificial Intelligence Review* 37, 3 (2012), 217–237.

[5] C. M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[6] L. Bottarelli, M. Bicego, J. Blum, and A. Farinelli. 2016. Skeleton-Based Orienteering for Level Set Estimation. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence.* 1256–1264.

[7] A. Castellini, G. Beltrame, M. Bicego, D. Bloisi, J. Blum, M. Denitto, and A. Farinelli. 2018. Activity Recognition for Autonomous Water Drones based on Unsupervised Learning Methods. In *Proc. 4th Italian Workshop on Artificial Intelligence and Robotics (AI*IA 2017),* Vol. 2054. 16–21.

[8] A. Castellini, G. Beltrame, M. Bicego, J. Blum, M. Denitto, and A. Farinelli. 2018. Unsupervised activity recognition for autonomous water drones. In *Proceedings of the Symposium on Applied Computing, SAC 2018.* ACM, 840–842.

[9] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. 2012. Sensor-Based Activity Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 790–808.

[10] M. Denitto, A. Farinelli, and M. Bicego. 2017. Biclustering of time series data using factor graphs. In *ACM SAC 2017.* ACM, 1–3.

[11] M. R. Endsley. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors* 37, 1 (1995), 32–64.

[12] A. Farinelli, 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 (2012), 477–501.

[13] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. 2008. An HDP-HMM for Systems with State Persistence. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08).* ACM, 312–319.

[14] D. Hallac, S. Vare, S. Boyd, and J. Leskovec. 2017. Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data. In *Proc. 23rd ACM SIGKDD (KDD '17).* ACM, 215–223.

[15] M. Hassani and T. Seidl. 2017. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam Journal of Computer Science* 4, 3 (2017), 171–183.

[16] Y. Hong, S. Kwong, Y. Chang, and Q. Ren. 2008. Consensus unsupervised feature ranking from multiple views. *Pattern Recognition Letters* 29, 5 (2008), 595 – 602.

[17] K. Jain and V. V. Vazirani. 2001. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM (JACM)* 48, 2 (2001), 274–296.

[18] L. P. Kaelbling and T. Lozano-Perez. 2013. Integrated Task and Motion Planning in Belief Space. *International Journal of Robotics Research* 32, 9-10 (2013).

[19] E. Kim, S. Helal, and D. Cook. 2010. Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Computing* 9, 1 (2010), 48–53.

[20] H.-P. Kriegel, P. Kröger, and A. Zimek. 2009. Clustering High-dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering. *ACM Trans. Knowl. Discov. Data* 3, 1, Article 1 (2009), 1:1–1:58 pages.

[21] Y. Kwon, K. Kang, and C. Bae. 2014. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications* 41, 14 (2014), 6067 – 6074.

[22] M. Lavielle and G. Teyssière. 2006. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal* 46, 3 (2006), 287–306.

[23] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. 2013. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 171–184. https://doi.org/10.1109/TPAMI.2012.88

[24] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. 2010. Understanding of Internal Clustering Validation Measures. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM '10).* IEEE Computer Society, 911–916.

[25] S. C. Madeira and A. L. Oliveira. 2004. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 1, 1 (Jan. 2004), 24–45.

[26] G. Montanez, S. Amizadeh, and N. Laptev. 2015. Inertial Hidden Markov Models: Modeling Change in Multivariate Time Series.

[27] E. Müller, S. Günnemann, I. Assent, and T. Seidl. 2009. Evaluating Clustering in Subspace Projections of High Dimensional Data. *Proc. VLDB Endow.* 2, 1 (2009), 1270–1281.

[28] L. Parsons, E. Haque, and H. Liu. 2004. Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explor. Newsl.* 6, 1 (June 2004), 90–105.

[29] S. Peignier, C. Rigotti, A. Rossi, and G. Beslon. 2018. Weight-based search to find clusters around medians in subspaces. In *Proceedings of the Symposium on Applied Computing, SAC 2018.* ACM, 471–480.

[30] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. 2014. Review: A Review of Novelty Detection. *Signal Process.* 99 (June 2014), 215–249.

[31] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong. 2013. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery* 26, 2 (2013), 332–397.

[32] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. 2013. An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression. *IEEE Trans. Automation Science and Engineering* 10, 3 (2013), 829–835.

[33] D. L. Vail, M. M. Veloso, and J. D. Lafferty. 2007. Conditional Random Fields for Activity Recognition. In *Proceedings of the 6th International Joint Conference on*

Autonomous Agents and Multiagent Systems (AAMAS '07). ACM, New York, NY, USA, Article 235, 8 pages.

[34] L. van der Maaten and G. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.