European Commission

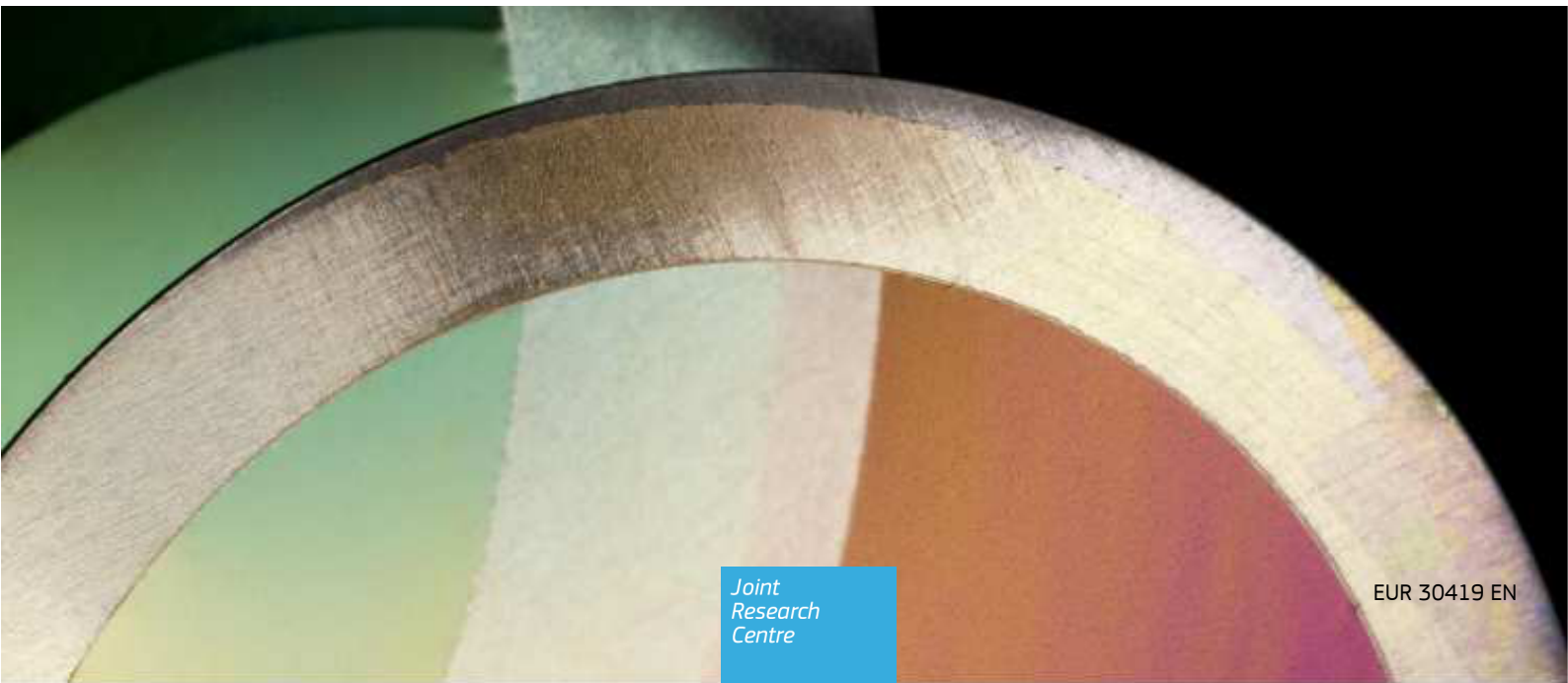
JRC TECHNICAL REPORT

Clustering and Unsupervised Classification in Forensics

From Theory to Practice

Junklewitz,H.
Ferrara, P.
Beslay,L.

2021



Joint Research Centre

EUR 30419 EN

Contact Information
Name: Henrik Junklewitz
Address: Joint Research Centre, Via Enrico Fermi 2749, TP xxx, 21027 Ispra (VA), Italy
Email: Henrik.Junklewitz@ec.europa.eu
Tel.: +39 0332 78 3571

EU Science Hub
https://ec.europa.eu/jrc

JRC119038

EUR 30419 EN

PDF ISBN 978-92-76-23872-0 ISSN 1831-9424 doi:10.2760/308387

Luxembourg: Publications Office of the European Union, 2021

How to cite this report: Junklewitz H., Ferrara P., Beslay L., Clustering and Unsupervised Classification in Forensics: From Theory to Practice, EUR 30419 EN, Publication Office of the European Union, Luxembourg, 2021, ISBN 978-92-79-23872-0, doi:10.2760/308387, JRC 119038

Contents

Abstract

Nowadays, crime investigators collect an ever increasing amount of potential digital evidence from suspects, continuously increasing the need for techniques of digital forensics. Often, digital evidence will be in the form of mostly unstructured and unlabeled data and seemingly uncorrelated information. Manually sorting out and understanding this type of data constitutes a considerable challenge, sometimes even a psychological burden, or at least a prohibitively time consuming activity. Therefore, forensic research should explore and leverage the capabilities of cluster algorithms and unsupervised machine learning towards creating robust and autonomous analysis tools for criminal investigators faced with this situation. This report presents a first comprehensive study from theory to practice on the specific case of video forensics.

1 Introduction

Nowadays, crime investigators collect an ever increasing amount of potential digital evidence from suspects, continuously increasing the need for techniques of digital forensics. Often, digital evidence will be in the form of mostly unstructured and unlabeled data and seemingly uncorrelated information. Manually sorting out and understanding this type of data constitutes a considerable challenge, sometimes even a psychological burden, or at least a prohibitively time consuming activity. Therefore, forensic research should explore and leverage the capabilities of cluster algorithms and unsupervised machine learning towards creating robust and autonomous analysis tools for criminal investigators faced with this situation.

As an illustrative study this report focuses on the specific context of digital video forensics. Because of the continuous spread of smart devices such as low-cost digital cameras, smartphones, tablets and many other similar devices, citizens have been increasingly becoming daily producers and consumers of digital contents like pictures, videos and audio recordings. Beside legal and innocent contents, this is assisting to the proliferation of multimedia content that can be used as means to perpetrate crimes, such as terrorism propaganda videos, personal treats (based on pictures and recordings), or even be illegal itself, such as pedo-pornographic material. Such a mass of content is increasingly distributed and shared between people either through traditional means such as the internet sites, social media, or the dark web. The latter has been receiving a growing attention, because it offers the possibility of establishing anonymous IP connections, making it even harder to go back to the source. What happens on regular basis is that whenever law enforcers sequester servers hosting these unlawful data, a connection analysis for retrieving the identity of the producer becomes a tough issue. Further, any manual analysis of a server of video content is extremely demanding in time and effort.

In this context, there is a need for developing tools that are able to link multimedia contents. This linking process can aid law enforcers to put these data in a context, helping them in their investigative purposes. For instance, let's suppose that a propaganda video is linked (i.e. the source device has been assessed to be the same) to another innocent video wherein the environment is recognized to be a well-known city; from these premises, investigators can fairly hypothesize that the author is in or has crossed the borders of that country. If the available prior information on the case is low, this scenario preciscely asks for unsupervised methods and clustering.

This report constitutes an initial study within the DFILE ("Digital Forensic Investigation Techniques") project, generally aiming at enhancing European crime investigator's technological capabilities. It first provides an initial theoretical study of the state of art of clustering and unsupervised machine learning approaches, with a focus on specific problems relevant for digital forensics (see Sec. 2section).

This is followed by an illustrative application study on exploring unsupervised approaches in the specific context of digital video forensics (see Sec. 4section). This is especially important in the context of the JRC-AVICAO ("Authors and Victims Identification of Child Sexual Abuse Online") work package, which aims to increase European Law Enforcement Agencies' capabilities in the fight against Child Abuse on-line. In addition, the JRC has identified other serious crimes for which video source camera identification might also support investigations, namely when terrorism-related videos are shared over the Internet. So far, the digital forensics tools developed in AVICAO have focussed on four of five use cases initially identified together with Europol's cybercrime unit EC3 - device classification, device verification, device-based image database retrieval and image-based database device retrieval (Satta and Beslay, 2014). All of these problems can be solved with a supervised approach. The fifth use case is the clustering of video based evidence and, thus, the main subject of this report's application section.

Other foreseeable application fields for unsupervised investigative methods in the future are the challenge of encrypted criminal material, analyzing fingerprints or the discovery of leads in large unstructured data bases. For this initial report, these will only be touched upon from a prospective point of view in the conclusions.

In order to conduct this report, the JRC has held in parallel an explorative expert workshop on clustering and unsupervised methods (Junklewitz and Beslay, 2018) in collaboration with Europol. The conclusions from this workshop have been worked into this report, especially to identify the most pressing challenges and issues (3section and 2.3 for clustering in a forensic setting.

1.1 Definition of Clustering

There is no universally accepted definition of clustering in the literature. Instead, a wide range of approaches and interpretations are associated with clustering across different fields of research (see Sec. 2.2). Often, essentially the same algorithms and methods for clustering tasks are used and developed, but with strongly different goals and terminology. Although clustering is usually intuitively interpreted along the lines of "the problem to partition data into a number of groups according to some measure of

similarity", it is hard to give a clear and unanambigous definition to this. The main reason for this is the vagueness associated with the meaning of "grouping" or what constitutes a "measure of similarity" (see Sec. 2section for more details).

The intuitive notion makes immediately clear that clustering is intimately connected to classification problems. Indeed, one valid definition stems from machine learning, where clustering is basically understood as a form of unsupervised classification, i.e. classification of unlabeled data for which no trainable model is readily available (see (Xu and Wunsch, 2009)).

Another point of view is more grounded in statistics, where the same task is seen as probability density estimation, usually of a mixture model that assumes that data points might be generated by a number of different underlying probability distributions. Also, this view is increasingly common in more recent advances in probabilistic machine learning, where probabilistic generative models play an important role.

Nonetheless, the most classical, and perhaps most widespread, interpretation of clustering is more grounded in exploratory data analysis and data mining. Here, clustering is simply understood as a technique to order available data into groups according to a number of easily attainable criteria, such as the distance of the data points in some measure. This view does not necessarily maintain that an underlying, "true" distribution is estimated. Instead, a more qualitative grouping of unordered data is achieved, usually subject to direct interaction from a human operator. Often, when the data becomes large or of high dimensionality, these methods remain the only feasible ones with limited computing ressources.

In the light of these considerations, we adopt a working definition for clustering for the purposes of this report. We try to be agnostic and cover all the mentioned conceptual approaches, thereby deliberately retaining some openess over exact clarity:

Clustering is a partly or fully unsupervised analysis of data to infer meaningful groups or classifications (classes, clusters). ([1])

Throughout the rest of this work, "class" is used to describe a member of the actual, true number of groups in the data, whereas "cluster" refers to a current grouping found by an algorithm.

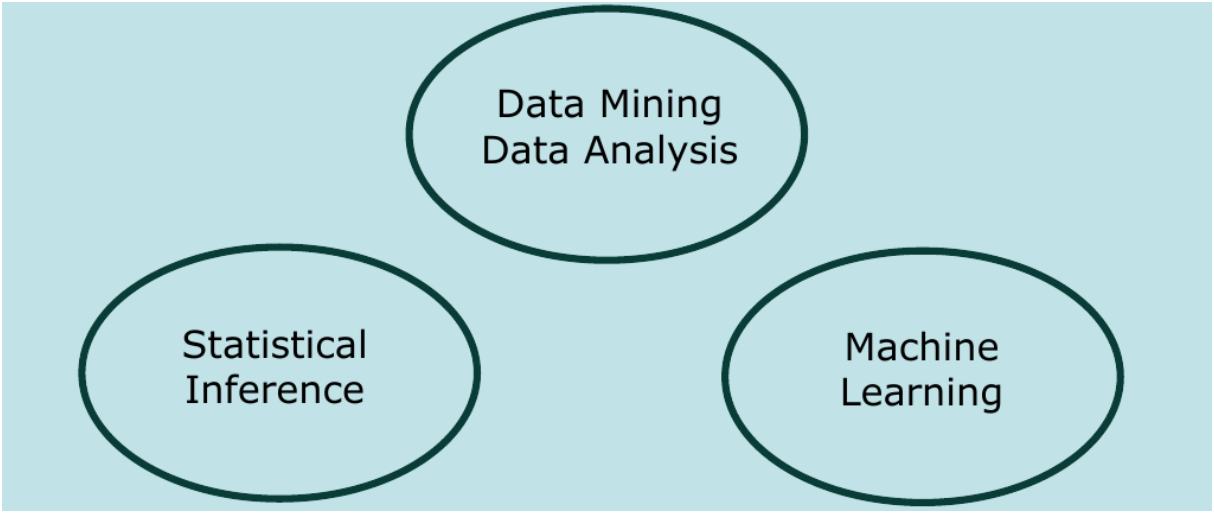


Figure 1: The three main disciplines from which techniques of clustering and unsupervised classification derive their methods.

1.2 Clustering and Classification

While considering the similarities, it seems equally important to clearly draw a distinction between clustering and classification. The important difference is that for the former no clear prior knowledge can be used and most crucially no training data is available, whereas for the latter a sufficiently large set of data is expected to be available to train a classification model. Naturally, clustering can only find less defined groupings whereas in classification, exact ordering into distinct classes is the goal. Since classification is a widespread application of supervised machine learning (ML), explicitly considering

([1]) "Meaningful" really depends on the context and is the main source of openess in this definition. It will be clearer when considering each particular algorithm. Groups really can be partitions, classifications, similar data groups, images or frequency components, sequence groups, text clusters etc....

clustering as unsupervised classification is a powerful way to approach clustering from a consistent point of view.

The most obvious benefit comes from being able to clearly distinguish important application cases. This becomes all the more important when faced with potentially large amounts of data, only limited ressources for processing, but nontheless a need for accurate results. A digital forensic scenario can easily become such a case, where any substantial lead out of considerable amounts of unordered data can make a difference for a criminal case.

It is always superior in terms of accuracy (and often computing time) to use a supervised classification method over a clustering algorithm, should labelled training data be available. This might even hold for the case where sufficient labelled data is only available through additional work, such as with data augmentation techniques or unsupervised pretraining methods. In addition, hybrid methods exist in machine learning and statistics on dealing with only partly labelled data, known as semi-supervised or missing data approaches. In cases where only a subset of the data is labelled, using a semi-supervised technique might thus be worth another consideration.

2 Background: Methods and Algorithms

2.1 Clustering Problem Categories

In practical terms the relationship between supervised and unsupervised classification means that, when faced with the task to "group data", it might be worthwhile to first investigate if the problem can be solved by (semi-) supervised machine learning instead of employing purely unsupervised clustering techniques. A clear example with high relevance for forensics would be any case involving automated classification of images into a clear number of known and easily definable categories, for instance gender estimation or face detection.

Of course, the boundaries are not sharp, and there are many cases where both types of methods can be applicable in principle. Good problem cases for an unsupervised clustering method are in general

— in the absence of trainable classification models, or labeled data to train them;

— severely lacking prior knowledge for a certain classification model to be justified;

— or simply, all cases where there is no need or time for a more complex data analysis and a more qualitative or preliminary grouping of data is sufficient.

The first two cases will be true in many forensics scenarios, for instance when unorderd data from a number of cases need to be cross-linked, categorized or classified into a most likely number of groups or connections. The last application case is of importance in cases where completely unordered data is available and any clue would be good, notwithstanding that the accuracy of the preliminary results will only be high enough to develop a qualitative assessment. Thus, even when a clustering approach might be the correct choice, there is still a wide range from exploratory data mining to accurate statistical estimation from which the clustering method can be chosen.

Utilizing this more granular view of clustering as a unsupervised classification, we explicitly acknowledge that there is a continuous range from hard supervised classification to fully unsupervised clustering into more subjective groups. As a clear guideline for our work on forensic problems, we thus roughly define four different problem categories, which serve to locate a problem at hand within this continuous spectrum:

1. Supervised classification. Classification into a number of $\mathcal{K}$ classes. A trainable model can be obtained and used to classify new data.

2. Unsupervised classification with strong prior knowledge. Situation with unlabeled data, but conditions are assumed to be mostly known, and prior knowledge available. Especially $\mathcal{K}$ and all hyperparameters governing the shape and distribution of classes are supposed to be more or less exactly known. This includes also most cases where some small subset of labeled data is available such that a semi-supervised approach might be possible. This is often not a very realistic scenario, but typical for testing conditions. It might be describing a scenario in which labeling and annotating to do 1. is too costly, but the data otherwise is of very good quality.

3. Unsupervised classification with weak prior knowledge. Some weak prior knowledge about the data might be available, such as a rough range of values for the number of classes $\mathcal{K}$, or some algorithm hyperparameters, which are known to be unimportant. Alternatively, just a few labeled examples might be available. This is a more realistic case, and the most likely application situation. In fact, this should describe most actual forensic application cases, especially if ways can be found to include some expert knowledge from investigators, or correlations from other investigation sources might give a lead.

4. Fully unsupervised classification without any prior information. This most probably turns the problem into a full data mining application for qualitative analysis, where some groupings and correlations of the data are looked for, without associating too much meaning or accuracy.

Of course, in reality the lines between those cases can be blurred, especially between 2 and 3 and when new information might become available from other sources. Nonetheless, we suggest to first identify a given problem roughly along these four categories and then decide whether a clustering application is necessary or a supervised classification approach might be feasible, which type of clustering is needed or

possible (e.g. more accurate or exploratory), and also how much should be expected in terms of accuracy and robustness of the results. A problem categorized as being fully unsupervised (4) will be much harder to solve to similar accuracy as in a case when some prior knowledge can be assumed to be known (2-3).

2.2 Overview of algorithms and methods

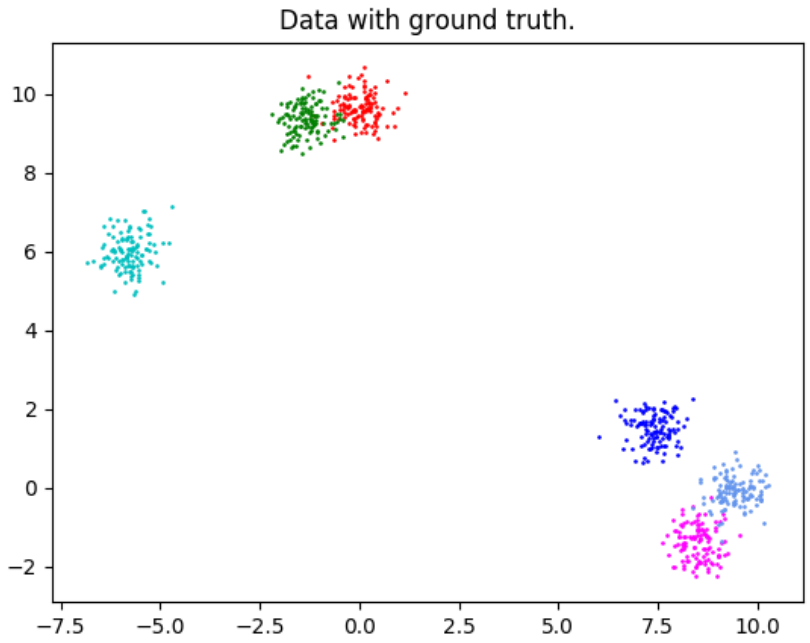


Figure 2: Simulated 2D data set of 6 classes randomly drawn from a multivariate Gaussian Model with a common standard deviation. The plot showes each data point colored with its membership to one of the six classes. This data serves as an illustrative case throughout this Section.

In this section, we give a broad overview of existing clustering algorithms and methods without any claim to be exhaustive. We mainly aim at covering the most important developments, drawing upon literature from classical computer science, statistics, machine learning and data mining. For a more thorough overview, please consult (Everitt et al., 2009, Xu and Wunsch, 2009, Bishop, 2006, Hastie et al., 2009). We place a focus to methods that we employ in later sections for forensic applications (see Sec. 4section), but aim at providing a comprehensive overview that might be useful in the forensic context for future projects.

For illustrative purposes we demonstrate key algorithms and approaches on a simulated data set of 6 classes in 2D (see Fig. 2), not designed to reflect a realistic forensic data set but to be instructive for the presented algorithms. As a general illustration of the idea behind the proposed clustering problem categorization in the previous section, we note that with a supervised SVM classifier using half the simulated data size for training, which means around 60 data points per class, we achieve a missclassification rate of around 3%. This is not very good for a SVM classifier, but no surprise considering the low number of training data. On the other hand, none of the presented unsupervised algorithms tested on this simulated data set can reliably achieve a similar classification rate, even when assuming the number of clusters and statistical behavior of the data to be exactly known.

As captured by the inherent vagueness of our definition of clustering in Sec. 1.1, the exact mathematical definition of what constitutes a "cluster" is highly dependent on the context and method used (Xu and Wunsch, 2009). We will stay with our general notion that "cluster" denotes a specific grouping found by an algorithm that may or may not accurately reflect an underlying distribution of true classes. This is important, since a found cluster might very well hint at a correlation in the data that does not need to be connected to the real underlying classes but still can hold valuable information, for instance in the case where noise from a similar source connects several independent classes. We mention more specific mathematical definitions together with certain classes of algorithms.

As with most algorithms, in clustering and unsupervised classification some objective or loss function is optimized to yield a final result. Often, the loss function can be interpreted as based on a distance

measure D between all data points (Xu and Wunsch, 2009, Bishop, 2006). This is especially true for most of the classical approaches (see Sec. 2.2.1) that, in fact, often start out from D. For our purposes, the two most important notions for D are the generalized Minkowski measure based on the L_p-norm, and the correlation measure, based on a correlation coefficient

$$D_{\mathrm{Lp}}(x_i, x_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}| \right)^p \tag{1}$$

$$D_{\mathrm{correlation}}(x_i, x_j) = \frac{\left(1 - \frac{\sum_{l=1}^{d}(x_{il}-\overline{x}_i)(x_{jl}-\overline{x}_j)}{\sqrt{\sum_{l=1}^{d}(x_{il}-\overline{x}_i)^2 (x_{jl}-\overline{x}_j)^2}} \right)}{2} \tag{2}$$

where D_{Lp} leads to a range of possible measures, for which the euclidean measure with $p = 2$ is the most widely used, and $D_{\mathrm{correlation}}$ is given as Pearson's classical coefficient, but could be as well a different correlation measure.

As with other methods of data analysis, also the results of clustering algorithms might depend on the choice of features derived from raw data. In principle, cluster algorithms exist for any kind of data and derived feature vectors, continous, discrete or even binary. Except for methods to reduce the dimensionality of the data (see Sec. 2.2.4), we will not concern ourselves with this topic in detail here, but will of course describe the specific features used in our forensic applications in Sec. 4section.

Of course, when handling combined data with vastly different scales or units, one should consider standardizing the data before applying a clustering method, for instance, see (Bishop, 2006) for an instructive example. For a detailed discussion of these topics, see e.g. (Jain and Dubes, 1988, Xu and Wunsch, 2009).

Arguably, the most prominent problem that many cluster algorithms are faced with are how to handle the number of clusters $\mathcal{K}$. This is also true in a forensic setting with larger amounts of unlabeled data, where often very few information can be assumed to be explicitly known about $\mathcal{K}$ and users will most probably not be algorithm experts, able to decide on their own what would be a proper choice for $\mathcal{K}$ when it is essentially unknown. This will be a recurring theme throughout this work and properly introduced in Sec. 2.2.3.

2.2.1 Classical approaches

The most classical methods for clustering data have their roots in early computer science and applied mathematics. The two basic approaches into which many clustering methods to date can be ordered are either based on data partioning or on hierarchical data trees (or in a more modern context graph analysis). The former go back to the work of (Lloyd, 1982) and subsequently the development of methods today collectively know as some variant of K-means. The latter are developments based on classical data hierarchy arguments developed in early computer science, e.g. (Defays, 1977).

Partitional clustering usually assumes some initial partioning on the full set of data points $x_i, i = 1, \cdots N$ into a number $\mathcal{K}$ of clusters $\{C_1, \cdots, C_K\}$ and then optimizes some objective function based on the chosen distance measure D of all data points. Thereby, the algorithm iteravily advances the exact partitioning by continuously changing the membership of data points to each cluster. These techniques usually can further be distinguished by whether they impose hard or soft boundaries, i.e. whether data points need to be exclusively member of one cluster or could also be assigned on a continuous scale, for instance a membership probability. Another angle from which to order most partitional clustering algorithms is by whether they are model based or nonparametric, i.e. whether a parametric statistical model is underlying the optimized objective function, or a nonparametric adaptive method is employed (such as in some kernel clustering methods, see Sec. 2.2.3). Especially soft partioning algorithms can often easily be interpreted in a probabilistic framework (see 2.2.2).

Hierarchical clustering methods, instead, work quite differently in that they always divide or merge up the data into either continuously smaller or larger patches, depending on whether they are designed to work in a top-down (divisive) or bottum-up (agglomerative) fashion. The hierarchy is constructed along a data tree, and a threshold criterion is usually set by the user, where to cut the tree for assigning data to a cluster. Hierarchical methods have been very successful in cases for which the data lends itself easily to a structure of a branched data tree, for instance genetic data or network graph data e.g. displaying relationships between people or objects.

2.2.1.1 K-Means based algorithms

K-means based algorithms are arguably the most widespread classical clustering algorithms, originally developed by (Lloyd, 1982). Often, still nowadays a variant of the K-means algorithm is the default

choice in many standard clustering packages. We will also apply K-means prominently in our forensic settings, albeit notifying the limitations of its approach along the way.

The K-means algorithm operates under the notion that each cluster C_n can be defined by a center point μ_n. Every cluster is grouped around those $\mathcal{K}$ cluster center points and the membership of each data point is determined by the center point closest in distance. Every data point is assigned to exactly one cluster denoted by a binary indicator $r_{nk} \in \{0, 1\}$, such that if data point x_n is assigned to cluster C_k, the indicator is $r_{nk} = 1$ and otherwise $r_{nk} = 0$. Thus, K-means provides a hard partitioning.

The objective function of the standard K-means algorithm is then defined as

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{3}$$

which represents the sum of squares of the distances of each data point to its cluster center.

When minimized for optimization, it becomes clear that this definition entails that the cluster centers are equivalent to the mean of all data points assigned to their respective cluster (Bishop, 2006), hence the name K-means.

Each iteration of the algorithm involves two steps, starting from an initial guess for the μ_k. First, J is minimized with respect to the r_{nk}l keeping the μ_k constant. In the second step, J is minimized with respect to the μ_k, now keeping the r_{nk} fixed. This procedure is repeated until some sort of convergence criterion is met. It basically means that, alternating, the cluster memberships and cluster centers are shifted around to iteravily accomodate a minimzed J. This convergence process is illustrated in Fig. 3 with the simulated data set.

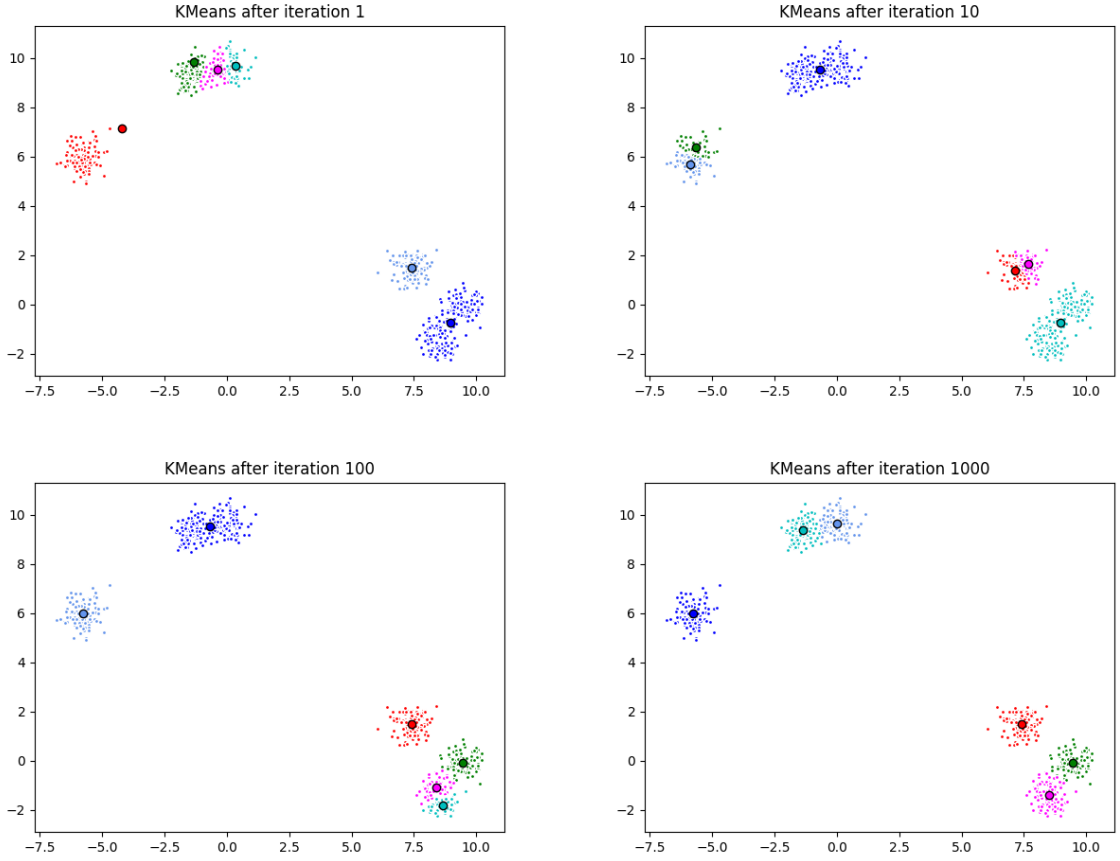


Figure 3: Illustration of the iterative process with which K-Means eventually converges on a final solution. Note that the number of clusters $\mathcal{K}$ had to be provided and although the true classes have been accurately guessed, not all data points have been correctly attributed in the two overlapping cases. Since K-Means is a hard partitioning method, it only can decide the membership for each data point once according to its closest cluster center. The intial values of the cluster centers have been chosen randomly. With more refined initializations, such as K-Means++ the convergence can be achieved much faster than the displayed 1000 iterations.

Over the years, many variants of K-means have been devised trying to tackle many of the issues with

this basic algorithm (Everitt et al., 2009). A full set of variants exist that modify the objective function J in some way, the most important one being the K-mediods algorithm that instead minimizes simply the absolute distance and not its square. Many ways of how to best initialize the algorithm have been proposed, the most successfull one being the K-means++ method that runs a number of initial tests on randomly sampled μ_k to start the algorithm.

The most pressing issue for many application is again that the K-means algorithm, and most of its variants, needs to be provided with a fixed number of clusters $\mathcal{K}$. Accordingly, for this intial report we have not explored the full range of all these methods and focused on applications using the standard K-means++ algorithm.

2.2.1.2 Hierarchical algorithms

Hierarchical algorithms take a specific approach to the clustering problem, grouping data with a nested sequence of dependent partitions ranging from a single cluster to considering each data point as a singleton cluster or vice versa, respectively known as divisive or agglomerative clustering. The resulting data structure can be displayed as a dendogram or a network of merging or branching nodes (see Fig 4 for an example on the simulated data) and crucially depends on the underlying distance measure D and, in the simplest variant, a user-set threshold where to cut the network into separate clusters. The distance measure is usually turned into a linkage function providing a network-specific way of measuring the closeness of nodes in a graph (Xu and Wunsch, 2009). The linkage again is used to construct an objective function, which has to be minimized. From this perspective, hierarchical clustering algorithms have to deal with similar numerical challenges as other type of clustering methods. Especially, the number of clusters $\mathcal{K}$usually needs to be provided and the cut-off is a highly imapctful user-dependent hyperparameter.

This approach is especially fruitful when such a hierarchical structure is naturally present in the data, for instance in evolutionary data or any sort of network analysis (Xu and Wunsch, 2009) and in those cases is also considered to offer a very natural way of visualization of the data structure and clusters in form of the dendogram.

Since our forensic data applications in this report are not naturally ordered in such a fashion, we will not prominently follow this type of clustering algorithm. We only provide some baseline results using an agglomerative clustering methods to validate our assumption that other methods are more useful for the tested cases. That being said, there are clear possible forensic application cases for hierarchical clustering methods, for instance in the analysis of networks of gathered evidences or analyzing links between forensic cases.

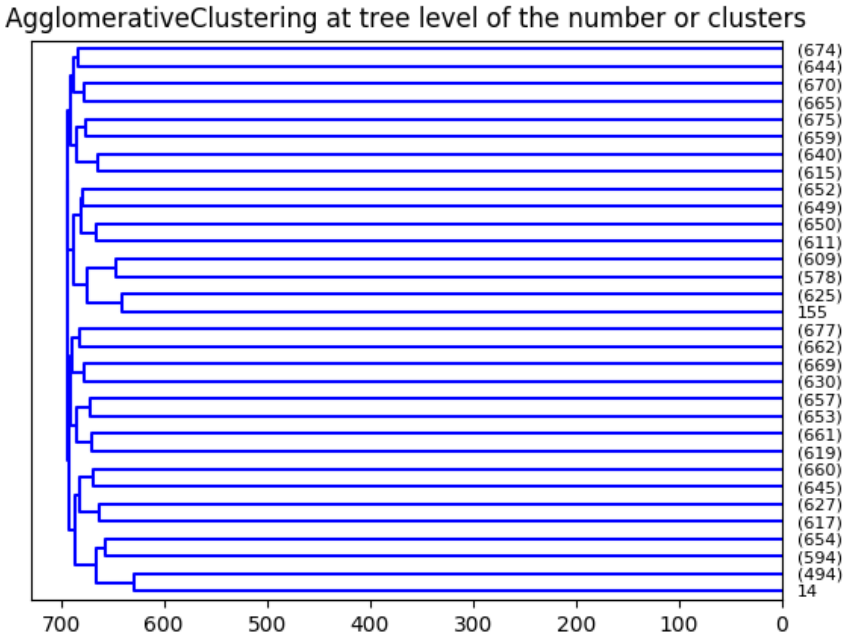


Figure 4: Typical, dendogram for an agglomerative clustering of the simulated data set of 6 true classes. The x axis shows the growing distance measure, the y axis a numbering of the data nodes. To the left and right, all further branches to fewer clusters than 6 more than 30 have been contracted.

2.2.2 Probabilistic Models for Clustering

Probabilistic models for clustering are almost all model- and partition-based algorithms. Recent years have seen an increase of the usage of probabilistic methods, largely because more complex statistical procedures are becoming more and more feasible even for larger data sets. There is also considerable overlap with some developments from probabilistic machine learning, where generative statistical models are becoming more prominent. We will not attempt to develop an exhaustive nomenclature and instead losely listing methods in this section or the next, mostly depending on whether they have been initially developed as a statistical or ML model. It is also for these type of models where the interpretation of clustering as a form of unsupervised classification becomes significant, since many of the probabilistic methods can in fact be used for supervised classification as well.

In general, probabilistic models have a number of advantages over classical methods. Most crucially, they allow to treat the problem of unknown number of clusters $\mathcal{K}$ as a statistical model comparison problem and, furthermore, offer direct ways in which these processes could be in principle conducted without intervention.

Secondly, probabilistic models by their nature provide a soft assignment to a cluster and in addition can be derived to provide a model uncertainty for a specific cluster assignment (see Fig. 5).

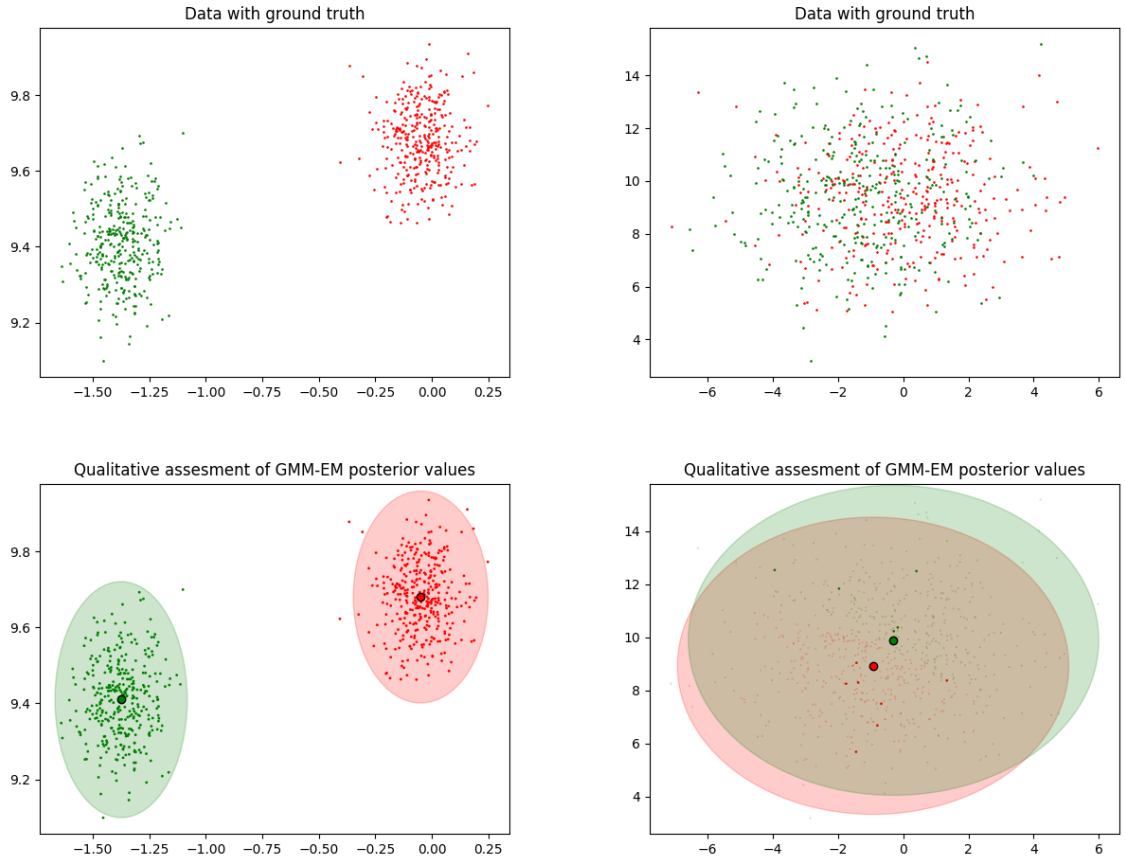


Figure 5: Illustration of how a probabilistic soft assignment algorithm provides a means to asses the model uncertainty of indivodual data points cluster membership. In the left column, a simple data set of two well separated classes has been simulated and subsequently analzed using a GMM model fit using the EM algorithm. Inthe right column, a similar data set with a higher standard deviation in each class has been produced and again the same GMM-EM algorithm has been applied. The size of the data points is proportional to their fitted log-likelihood, equivalent to a posterior probability for each data point to be assigned to its cluster. The plots illustrate well, how in the left case only very few data points come with a significant confidence in their assignment. In fact, most data points have a posterior probability of less than 0.5, which amounts to an immediate quality assessemnt of the clustering and setting a warning flag that these results should not be trusted too much. Note that the low posterior values are also consistent with the overlap of the fitted covariance regions.

All of this needs to be traded of with a generally considerably higher computational load. It needs to be decided case by case which problem can be solved and under what amount of computational resources.

2.2.2.1 Gaussian Mixture Models with the EM algorithm

The above explanation of the K-means algorithm (Sec. 2.2.1.1) can easily be interpreted as a statistical estimation procedure. The objective function of K-means is basically equivalent to the statistical assumption that a likelihood estimator minimizes the squared error norm, while restricting the data to only belonging to one single parameter of the model, in this case the cluster centers (Bishop, 2006).

In fact, this interpretation has led to a class of algorithms that by themselves have become a staple of cluster analysis: using Gaussian Mixture Models (GMM) with the Expectation-Maximization (EM) algorithm.

With a GMM, the clusters are interpreted to result from a mixture of $\mathcal{K}$ underlying Gaussian distributions with the different parameters of the Gaussians indicating the individual cluster characteristics. The means μ_k of the Gaussians become the cluster centers, their covariances Σ_k shaping the form of the cluster distribution around the center. This approach essentially turns the clustering problem into one of probablistic parameter inference (Bishop, 2006). The choice of a GMM to represent the clustered data as drawn from $\mathcal{K}$ different Gaussian distributions is not only motivated by its well-known algorithmic feasibility. It is further often a reasonable assumption that individual feature vectors of the data will be clustered around the true class vector due to a range of small stochastical effects and will thus be likely normally distributed according to the central limit theorem. Within that regime, a GMM offers a high flexibility to represent clusters of different sizes and shapes, which are governed by a full covariance matrix allowing a distribution in all feature dimensions.

A mixture model can be solved with a number of approaches. In the classical clustering literature, the most standard method is using the EM-algorithm. The basic approach progresses as follows. The cluster problem is represented by the challenge to estimate to parameters of the Gaussian mixture likelihood:

$$\mathcal{P}(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left[\sum_{c=1}^{\mathcal{K}} \pi_c \mathcal{N}(x_n|\mu_c,\Sigma_c) \right] \tag{4}$$

where $\mathcal{K}$ denotes the number of clusters, and π_k, μ_k, Σ_k denote respectively the $1-of-k$ cluster indicator vector, the class mean and the class covariance matrix. Very similar to the K-means procedure, EM is a well know statistical iteration algorithm, alternatively calculating an expected value for the GMM means μ_k and covariances Σ_k, followed by a maximization of the underlying objective function to update the cluster indicator vectors π_k. In fact, K-means can directly be interpreted as a simplified version of this EM procedure with a GMM.

This GMM-EM approach will mark the basis of our investigations for this report to represent probabilistic models, including the presentation of a preliminary approach to estimate the most likely number of clusters $\mathcal{K}$ (see Sec. 4.4). For the moment, we refer to other models only as possible extensions for a future approach.

2.2.2.2 Other Probabilistic Models

The recent literature on using probabilistic models for clustering and unsupervised analysis is growing, especially due to the rising applicability of approximate Bayesian models, e.g. (Kingma and Welling, 2014). Not all approaches will be useful for a forensics application, especially for those where the computational demand on larger data sets might outweigh their advantages. In the following we give a quick overview of methods that we deem promising.

All statistical algorithms can be combined in one way or another with some sort of meaningful statistical model comparison to find the most likely solution given different competing cluster models for a specific case. The most promising application is trying to estimate the model with the most likely number of clusters $\mathcal{K}$ in this way, for a premilinary example combines with a GMM-EM approach see Sec. 4.4. There is a host of different approaches for this (Gelman et al., 2004), in particular when changing the basic algorithm with which the model is evaluated.

Especially mixture models can be solved with a range of different algorithms and also need not be restricted to Gaussian distributions. Especially for different type of data, for instance text-based, a discrete mixture model is much more approriate, e.g. as in latent dirichlet allocation, a well known method for classifying text into topics (Blei et al., 2003). A Bayesian model can be used instead of the purely likelihood-based approach in the classical GMMM-EM algorithm. Since Bayesian models by design assume parameters to be distributed with a prior instead of being fixed, there are various possibilities in how one can find a full Bayesian multilevel statistical model (Gelman et al., 2004) for the clustering that provides an inference of any parameter with its uncertainty, for instance cluster centers μ or cluster membership labels π . In particular, this also can include the number $\mathcal{K}$, which would be handled explicitly as a solvable parameter (Bishop, 2006) and thereby obtaining a direct probabilistic statement

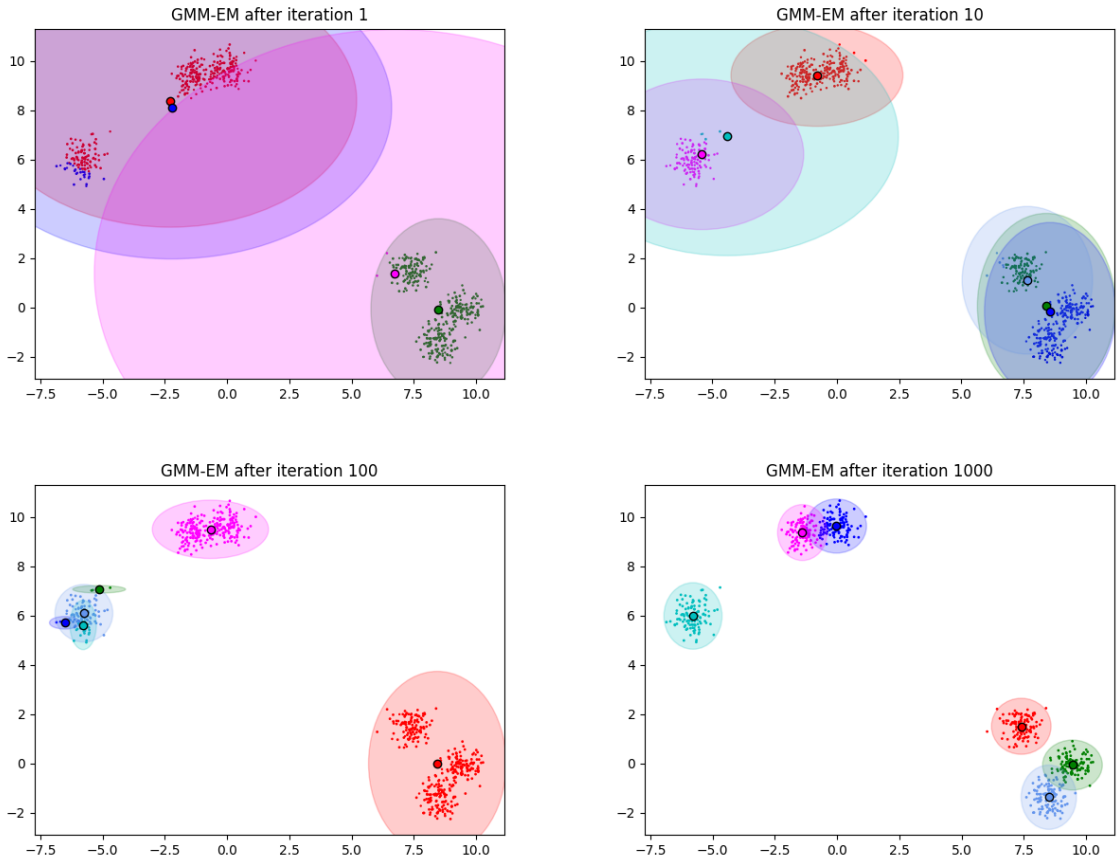


Figure 6: Illustration of the iterative process with which the EM algorithm eventually converges to a final solution for a Gaussian mixture model. The number of clusters $\mathcal{K}$ had to be provided. Note how the covariance regions of the Gaussians consistently converge to represent the typical spread of the clusters. For the two overlapping cases this also means that the algorithm accurately assumes that there might be a region between the found clusters where attribution of data points to single clusters is ambigous. The intial values of the cluster centers have been chosen randomly. With more refined initializations, such as using a run of K-Means, the convergence can be achieved much faster than the displayed 1000 iterations.

about $\mathcal{K}$ instead of a post-analysis model comparison. If we take the GMM model as an example, such a method would use a posterior $\mathcal{P}(\pi, \mu, \Sigma, K|x)$ to infere pi, μ, Σ, K by enriching the Gaussian Mixture Likelihood with suitable prior distributions

$$\mathcal{P}(\pi, \mu, \Sigma|x) = \sum_{n=1}^{N} \ln \left[\sum_{c=1}^{\mathcal{K}} \pi_c \mathcal{N}(x_n|\mu_c, \Sigma_c) \mathcal{P}(K)\mathcal{P}(\pi)\mathcal{P}(\mu)\mathcal{P}(\sigma) \right] \tag{5}$$

Nonentheless, such models will either need to be solved in a approximate inference framework (Gelman et al., 2004, Kingma and Welling, 2014) or using posterior sampling techniques (Gelman et al., 2004), which could possibly make them prohibitively costly.

There are also possibilities for statistical non-parametric and non-partitioning models to be applied to clustering. For example, Gaussian Process models can be used to design a a probabilistic variant of a kernel K-means algorithm (see Sec. 2.2.3). A different successfull route from recent years has been to apply results from probabilistic message passing to clustering, for instance the method of affinity propagation (Frey and Dueck, 2007) which has seen some success in computational biology. We test affinity propagation on our forensic video data and see promising results (see Sec. 4section).

2.2.3 Modern Machine learning models and stand alone developments

Recent decades since the early 2000s have seen a surge of new of machnine learning, graph theory based and stand-alone algorithms for clustering and unsupervised tasks. Many of those are designed for specific application cases, are often methods that look for different data representations in which a found data grouping might be more suitable or of reduced dimensionality, or are still relatively new and untested.

The most successful stand alone clustering algorithms from this phase is DBSCAN (Ester et al., 1996). It defines a specific user defined density measure of points and only uses that as a criterion for the clustering instead of a specific distance metric, which allows for a high tolerance for irregularly shaped clusters. We test DBSCAN but conclude that others are more suitable, at least for the purposes of video data (see Sec. 4section).

Kernel based methods have been introduced into clustering following the success of support vector machines in non-parametric supervised machine learning (Cortes and Vapnik, 1995), the most wide spread being kernel K-means (Xu and Wunsch, 2009) which experiments with different non-linear data representations.

Mostly for the same reasons, neural network based methods were introduced early into clustering, especially with some early success of self-organizing maps, an early day unsupervised neural network approach (Xu and Wunsch, 2009). Nowadays, deep learning is introduced heavily into clustering, still mostly for representation learning or dimensionality reduction using unsupervised methods such as autoencoders and Boltzman machines. A first straight forward application of a deep autoencoder for dimensionality reduction has been explored for the forensic video data in Sec. 4section.

Generative deep learning models have been introduced as well, for instance variational autoencoders, which can provide a powerful combination of deep learning representation learning and probabilistic GMMs (Dilokthanakul et al., 2016). Most of these newer methods are still relatively experimental, but research applications can be seen increasingly.

2.2.4 Dimensionality Reduction

Having to tackle large data sets is a long-standing problem in clustering, especially for data mining applications. Clustering high dimensional data presents two main challenges: Firstly, it might lead to unsatisfactory results because of the effects of the well known "curse of dimensionality", where the space volume gets concentrated in a thin shell and distance variations tend to vanish. Thus, different class center classifications become indistinguishable in high dimensional spaces, see e.g. (Beyer et al., 1999) and (Bishop, 2006). Secondly, the computational complexity grows with the size of the data and quickly becomes prohibitive for many clustering approaches.

In some cases, projection algorithms such as PCA or random sampling can be successful and a powerful tool to still allow to apply a standard clustering method. We provide a successfull example of using a sparse sampling methodology for the microphone data in Sec.4section.

Another important development is graph theory based clustering in which results from applied mathematical graph theory are applied to the problem. A very successful algorithm of this class is spectral clustering. It uses the eigenspectrum of the distance matrix between all data points as a representation of the data in the eigenbasis from which a suitable clustering can be deducted (Ng et al., 2001). Spectral clustering is applied with some success to process the image-based SPN features of our video data in Sec.4section

Finally, deep-learning based representation learning can also be viewed as a dimensionality reduction methods, and is increasingly been seen in the context of large data clustering (Aljalbout et al., 2018). An example is applying a deep convolutional autoencoder to image cluster problems before applying a standard method .

2.2.5 Cluster validity, model checking and hyperparameter optimization

Classically, checking the validity of a clustering model is a subjective and highly user interactive procedure that draws upon a host of different scores, indices, statistics, repeated visualizations and aggregate methodologies (Jain and Dubes, 1988, Xu and Wunsch, 2009). The literature is abound with a large host of proposals to assess the quality of a particular partitioning found by a cluster algorithm (Xu and Wunsch, 2009) for a comprehensive list). More reliable scores can be constructed for external cluster indices, which compare a given partitioning of data to an externally available, in our case the ground truth labels.

One model feature of particular importance, the most likely number of clusters $\mathcal{K}$, is classically also determined by relying on aggregate lists of such indices and scores to produce an average likely estimate for $\mathcal{K}$ (Xu and Wunsch, 2009) that often is hard to evaluate.

Probabilistic models offer a different route, but are of course more complex to compute. Frequentist hypothesis testing has been developed for cluster model checking (Tibshirani et al., 2000), Bayesian uncertainty estimates can in principle suggest a reliability of results (see Fig. 5)and generative models can use the estimated posterior to inspect typical candidates from the fitted parameter space.

In low dimensional cases, visualization tools are often the quickest method of choice to assess the basic validity of a clustering model. For higher dimensional data this becomes naturally impossible in a straight

fashion. High dimensional visualization tools such as T-SNE (van der Maaten and Hinton, 2008) exist, but are a double-edged sword because of their restricted interpretability, especially for an application case in which non-experts users are expected to be operating the method such as in an investigative forensics setting (see Fig. 7 for a simulated example).

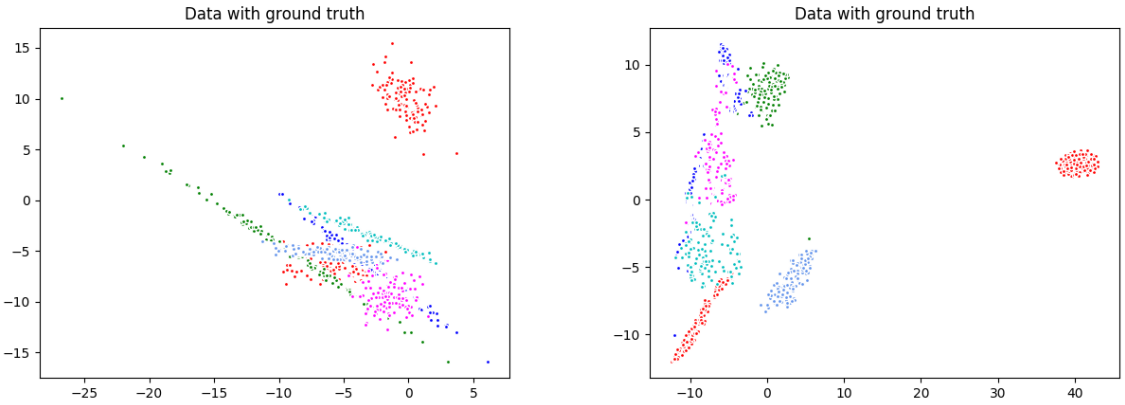


Figure 7: Illustration of the complexity of using a high dimensionality visualization tool. In both cases a more realistic data set of 6 classes has been simulated using non-isotropic Gaussian distributions. On the left, the data has been simulated in two dimensions, on the right in three and then visualized using T-SNE, which is the most successful high dimensionality visualization technique currently in use. The random seed and the cluster center distances for both simulation are the same. It can be seen that the T-SNE visualization on the right side is hard to interprete, especially note that distances between clusters or cluster shapes do not matter in the visualization. What can be seen is that the basic separation of a group of tangled clusters and one well separated cluster is still visible. This type of result is only achievable with a significant amount of adjusting hyperparameters, which seems out of the scope for an application aimed at non-expert users.

Finally, iterative approaches in which hyperparameter optimization is applied to produce a range of increasingly better performing cluster models are another route to produce some confidence in a cluster model. Hyperparameter optimization is a technique that is heavily employed in deep learning to automatically adjust the sets of complex hyperparameters in modern machine learning models. We have tested random optimization techniques in Sec. 4section, but so far without significant results.

2.2.5.1 Evaluating test data

For our later purposes in evaluating our test data in Sec.4section, we are focusing on two external cluster indices, which compare a given partitioning of data to an externally available, in our case the ground truth labels. The two indices we have chosen are among the two most widely used external indices, the Adjusted Mutual Information Score (AMI) and the Adjusted Rand Index (ARI).

The AMI basically measures the overlap of two partitionings U and V of data with a number of classes of u and v respectively, adjusted for a chance assignement. Mutual Information calculates the Kullback-Leibler divergence for the observed joint frequency distribution $P(i,j)$ of objects belonging to a class U_i or V_j in both partitionings and the two marginal frequency distributions $P(i)$ and $P(j)$ of objects belonging to a class just in one partitioning

$$MI(U,V) = \sum_{i}^{u}\sum_{j}^{v} P(i,j)\ln\frac{P(i,j)}{P(i)P(j)}. \tag{6}$$

For the AMI, this equation is further adjusted to correct for an agreement due to pure chance, making a number of assumptions on the expected value of the MI under pure randomness (Xu and Wunsch, 2009). The AMI takes a value of 1 when both partitions are identical, and 0 when the value equals the expectation for pure randomness.

Similarly, the ARI is a chance adjusted measure for similarity of two partitionings. The Rand Index measures the frequency of agreements between pairs of objects over the number of total pairs:

$$R = \frac{a+b}{n(n-1)/2} \tag{7}$$

where a is the number of pairs that contain objects from the same class for both partitionings and b the number of pairs that contain objects from different classes in both partitionings. Again, this equation

is adjusted for a chance agreement (Jain and Dubes, 1988) to arrive at the ARI and also yields a value between 0 and 1.

2.3 Problems and Challenges

Given the previous background overview of the state-of-art of clustering and unsupervised classification, we have collected a brief list of the most pressing problems and challenges from the point of view of digital forensics in an investigative setting.

This setting is likely dominated by our problem categories 3 or 4 (see Sec. 2.1), i.e. we cannot assume much of prior information already to be known in an ongoing investigative case. Although this is a somewhat conservative position, it is more likely that a crucial parameter such as the number of true classes $\mathcal{K}$ is unknown or only estimable from an investigator's domain expertise. Further, a robust algorithm should not require much technical interaction, which nonetheless often would be needed in many such cases, to assess the quality of the clustering or adjust hyperparameters. Finally, quantifying the uncertainty of the results in some way is important to make sense of the type of inherently more explorative information that can be gained from an unsupervised analysis.

From this, a more detailed list of problems and challenges from our view contains the following points:

Technical and algorithmic challenges

— How to deal with an a priori unknown or only roughly known number of clusters?

— How to evaluate the space of possible hyperparameters? Is it possible to evaluate the space of hyperparameters for each application? Are there best parameter settings?

— How to handle singularities/outliers/noise/missing data (identifying them and considered as nuisance parameters)?

— How to incorporate possibly only some prior knowledge (e.g. some labelled data, a probable range of classes/components, general: semi-supervised settings)? Related: How to incorporate completely qualitative user domain-knowledge?

— How to deal with real big data? Go for specific methods? Always trade accuracy for data size?

Analysis, Visualization and Validity

— How to best access validity/uncertainty of a given result? Knowing the ground truth and in real applications. Depending on Method?

— How to visualize or convey complex high dimensional results to, and how to interact generally with possibly non-expert users?

Methods, software and hardware considerations

— How to choose the right algorithm/approach/method/pipeline?

— Which software to use?

— Are hardware considerations important?

This analysis and is complementary workshop were exactly aimed at identifying the challenges that we most probably need to tackle first (5section.

3 Research Workshop at JRC

In parallel and support to this initial study the DG-JRC E.3 unit has held a scientific workshop on "Clustering and Unsupervised Classification in Forensics" in July 2018. The core idea of this workshop was to bring together a small, interdisciplinary group of European experts in clustering and unsupervised classification both from theoretical methods research and from various fields of scientific application. The aim was to learn about techniques, approaches and applied settings, and to discuss how those could be applied to solve unsupervised forensic data analysis problems. It was an explicit goal to bring together European experts from a wide range of backgrounds to explore new insights and to foster cross-field discussion, exchange and future collaboration between different application fields (among others bioinformatics, astrophysics and audio signal processing).

The workshop was presented with the same list of challenges as raised in Sec. 2.3. Here we only cite a condensed version of the workshop's main conclusions. The most important insights and expert proposals, which have been guiding the rest of our initial developments presented in the following chapters, were:

— For a new problem start with a literature review, but eventually spell out clearly all assumptions and needs and adopt standard methods accordingly. Do not get stuck in a specific technical bubble (for our literature review and choice of methods see Sec. 2section and Sec. 4section).

— Since not all problems can be solved by one certain approach, prioritize the challenges in Sec. 2.3 and focus on a specific problem (in our case, this is the number of classes $\mathcal{K}$, see Sec. 4.4).

— Make use of probabilistic methods if uncertainty and quality assessment is of importance. Also, probabilistic methods are the current state of the art solution for hyperparameter optimization in machine learning and statistics (both, probabilistic clustering and hyperparameter optimization are explored in the following chapters).

— For high dimensional data, different data representations and algorithmic approaches can make a huge difference. Consider dimensionality reduction methods; for images, the state-of-the-art is clearly neural network based representations (different dimensionality reduction methods and representational models are explored in Sec. 4section).

— Interaction with users and domain experts is challenging. Good method design should keep away complex algorithmic problems and expose more intuitive elements. This is also a good way to incorporate domain knowledge (clearly identified as one of our main challenges; the choice of handling $\mathcal{K}$ autonomously is a first step).

4 Forensic Application Case

This section provides a first, exploratory application of clustering to a digital forensic case. As outlined in the introduction, the use case that we are addressing is concerned with the problem of source device identification of multimedia contents, in particular images and videos. For the research at the JRC, this is extremely relevant in the context of fighting against child abuse online.

An interesting and challenging scenario is when a collection of media content has to be grouped according to the same source device, without any prior knowledge about the type and/or the number of devices that have produced these files. This scenario is actually extremely common during investigations, where sequestered web servers, personal computers or smartphones factually constitute enormous archives of multimedia contents coming from the most disparate sources. In this setting, grouping such heterogeneous data can help to discover unknown links between investigative cases, leading to new possible evidences.

Methods from pattern recognition, machine learning are extremely useful for this purpose. An overall work-flow usually includes mainly two steps: a feature extraction process, wherein intrinsic discriminating characteristics are modeled and extracted from a given data, and then a classification process, that is accomplished according some suitable criteria (i.e. distance, likelihood, etc.) defined for addressing a specific problem. Nevertheless, it has to be taken into account that because of the nature of the scenario, prior information (the number of authors, number of devices used, post processing used and so on) are limited or largely unknown. Consequently, a solution has to be sought with the methods from unsupervised machine learning and clustering reviewed in Sec. 2section, rather than with supervised classification.

With the aim of emulating the described investigative case, as an initial step, a video data set has been assembled according to a controlled protocol. Then, in continuation with our previous activities, we exploit two features capable of characterizing the traces that a source device leaves inside any multimedia content: the sensor pattern noise (SPN) introduced by the camera sensor within any visual content, and the microphone response extracted from the video's audio track.

This application study has two main goals. First, to explore the results of applying standard clustering methods to the aforementioned features. Second, to highlight and identify the most challenging issues from our theoretical list (see Sec. 2.3) in a real scenario. We basically provide a study under the clustering problem type 2 from our problem classification in Sec. 1.2, assuming all essential knowledge known.

4.1 The data sets

For our forensic application case, we work with three different sets of data.

Since the SPN features are known to be highly sensitive to video stabilization algorithms (see 4.3), we devise a simple test data set of still images from 5 cameras containing the ground truth (see 4.3 for a description). In this way we first test the feasibility and performance of clustering algorithms for the basic SPN feature vectors, without having to deal with the effects of the video stabilization.

Table 1: Devices corpus for video clustering.

Device model	Sampling Rate	Audio Codec	Video Resolution	Video Compression	# devices
Apple Iphone 4	44100 Hz	MPEG - AAC	1280x720	H264 - MPEG4	2
Apple Iphone 6	44100 Hz	MPEG - AAC	1280x720	H264 - MPEG4	1
HTC One X	48000 Hz	MPEG - AAC	1920x1080	H264 - MPEG4	3
Sony Xperia S	48000 Hz	MPEG - AAC	1920x1080	H264 - MPEG4	3
Samsung Galaxy Nexus I92	48000 Hz	MPEG - AAC	1280x738	H264 - MPEG4	2
Samsung Galaxy Nexus S	32000 Hz	MPEG - AAC	720x480	H264 - MPEG4	1
Nokia Lumia 735	48000 Hz	MPEG - AAC	1920x1080	H264 - MPEG4	3
Samsung ACE GT-S5830	48000 Hz	MPEG - AAC	640x480	MPEG4	20
Samsung Galaxy S6	48000 Hz	MPEG - AAC	3840x2160	H264 - MPEG4	1
GoPro HERO 4	48000 Hz	AAC	3840x2160	H263 - MPEG4	1
HTC One m9	48000 Hz	MPEG - AAC	3840x2160	H264 - MPEG4	1
BlackBerry Torch 9800	32000 Hz	MPEG - AAC	640x480	MPEG4	1
BlackBerry 9900 Qwerty	48000 Hz	MPEG - AAC	1280x720	H264 - MPEG4	2
Nokia Lumia 435	48000 Hz	MPEG - AAC	880x448	H264 - MPEG4	1

A second, larger benchmark data set of video also containing audio traces has been produced with which we test both, SPN and microphone features. The data set is composed of the raw data (i.e. video recordings) and the related ground-truth information (i.e. a device identifiers). Forty-two smartphones, comprising different brands and models, have been collected. It is worth to note that for some brand/models, more than one device of the same type is present, in order to evaluate if the method is able to discriminate between two different devices of the same brand/model. In Table 1 the complete list of devices is shown 1. This data set is further divided into two different settings: one of controlled recordings to reduce most unaccounted effects on the data and a set of live recordings. The controlled data set fulfills the same role for the microphone data as the still image data set for the SPN feature. We thus only test the SPN features for the live recordings.

4.1.1 Controlled recordings

The first data set is acquired according to the following protocol:

— A suitable video sequence is reproduced by means of a LCD screen and loudspeakers for audio, and recaptured by means of the all the smartphones;

— The smartphones are placed always in the same positions with respect both the room walls and the audio/visual sources;

— A video sequence, whose duration is at least 3 minutes, is recaptured and then trimmed in subsequences of 6 seconds, for each device;

— The source video sequence is composed of a set of video recordings from VIDTimit Audio-Video data set (Sanderson and Paliwal, 2004)(Sanderson and Lovell, 2009). Although the data set was

conceived for speaker and speech recognition from audio/visual features, it was suitable also as data set for our purposes. This is composed of small sentences (3 seconds each) in English, from people of different ages, with different accent and balanced in gender. We randomly select a subset of sentences, taking care of having no repetitions and a balance in gender speakers, to be concatenated in the source video. The aim of this first set of data is:

— To verify that the method effectively estimates the microphone response instead of the environment;

— To reduce as much as possible undesired noises in the recordings, that could have made the results analysis more difficult;

— To make an analysis on a wider typology of speeches, in term of age, accent, gender, which is difficult to reach in practice with live recordings.

4.1.2 Live recordings

The second data set is acquired with the following protocol:

— Two video recordings of at least two minutes with at least one person speaking are recorded indoor (large offices) and outdoor, for each device. Two male and one female voices are randomly present in the recordings, speaking English;

— Two video recordings of at least 1 minutes are recorded with no speech are acquired indoor and outdoor, for each device, so that the audio traces contain only environmental sounds;

— The recordings are trimmed in sequences of duration 6 seconds. The aim of this second set of data is to simulates real recordings, wherein speech or simply environmental noise might occur.

4.2 Audio-based clustering of video recordings

In the following subsections we describe the methodology we used to conduct a test run on clustering digital videos from the audio track, in function of the source device. First, we describe the audio features that are able to discriminate between different devices; then, we show the performance in terms of capability and limitations of the unsupervised classification methods introduced in the previous sections, under different assumptions or working conditions.

4.2.1 Audio features

In this subsection we go through the method used to characterize the microphone response. The algorithm relies on the work in (Cuccovillo et al., 2013b), where it has been used for audio tampering detection. Such an approach is based on blind channel magnitude estimation (Gaubitch et al., 2013)(Gaubitch et al., 2011), wherein the term "channel" refers to the microphone frequency response in (Cuccovillo et al., 2013a)(Cuccovillo and Aichroth, 2016) and in our study, rather than the acoustic environment, as originally conceived.

In general, the recorded audio signal can be modeled in the discrete time domain as follows:

$$x(n) = s(n) * h(n) * v(n) + w(n) \tag{8}$$

where the recorded signal $x(n)$ is expressed as convolution (operator $*$) between the emitted signal $s(n)$ and the impulsive responses of the microphone and the environment response, respectively $h(n)$ and $v(n)$. A term of additive noise $w(n)$ is also present for accounting for all type of electronic disturbances. Equation 8 can be expressed in the frequency domain by means of Short Term Fourier Transform (STFT) as:

$$X(k,l) = S(k,l)H(k,l)V(k,l) + W(K) \tag{9}$$

where $k = 0, \ldots, N_{FFT}$ and $l = 0, \ldots, L$ are frequency and time frame indexes, and $X(k,l)$,$S(k,l)$,$H(k,l)$, $V(k,l)$ and $W(k,l)$ are complex numbers.

By assuming a noiseless model, neglecting the environment response and assuming that the channel is a stationary quantity, we can rephrase our model as follows:

$$X(k,l) \approx S(k,l)H(k) \tag{10}$$

Passing to the logarithm of STFT magnitudes, we obtain:

$$log(|X(k,l)|) = log(|S(k,l)|) + log(|H(k)|) \tag{11}$$

Let's suppose now to know the log-spectrum $log(|S(k,l)|)$ of the input signal, the microphone response could be estimated as:

$$\underline{\hat{H}}(k) = \frac{1}{L}\sum_{l=0}^{L}(\underline{X}(k,l) - \underline{S}(k,l)) \tag{12}$$

where $\underline{A} = log(|A|)$, $\hat{A}$ is the estimate of A and L is the number of time frames.

However, in a forensic scenario the original signal $S(k,l)$ is unknown, but we can think to estimate $S(k,l)$ from the recorded signal $X(k,l)$. In a nutshell, the core of the method relies on finding a good estimation of the original signal, because this will affect the accuracy of the channel estimated.

To obtain the estimate $\hat{S}(k,l)$, speaker recognition literature can help to cope define suitable prior information about $S(k,l)$. From now, the focus is on speech as input signal. Concerning that, a vast literature has been produced so far, starting from (Hermansky and Morgan, 1994) wherein RASTA-filtered Mel-Frequency Cepstral Coefficients (RASTA-MFCC) have been successfully used to model human voice for speaker and speech identification. Beyond that, it is worth to note that such a feature has shown to be robust (i.e. independent) to the distortion introduced by the microphone. In (Gaubitch et al., 2013), it is shown that combining RASTA-MFCC and Gaussian Mixture Models (GMM) allows to obtain a good estimation of the original (called "clean" hereafter) speech.

4.2.1.1 GMM Training for Clean Speech

In order to reliably estimate the microphone frequency response, we define a M-components Gaussian Mixtures Model, whose components are associated to average clean spectrum. This model is our prior knowledge on about input signal and has to be learned from data by training. This is an off-line process that has to be performed just one time, once all the parameters of the system are fixed. More in detail: a training set of clean speeches $s(n)$ is split into overlapping windowed frames and the STFT is applied to obtain $S(k,l)$. Then, a vector $\mathbf{c_s}(l) = [c_s(0,l)c_s(1,l)\dots c_s(N-1,l)]$ of N RASTA-MFCCs and the average log-spectra $\underline{S}(k,l)$ are calculated for each time frame. Furthermore, the mean of the log-spectrum is subtracted as

$$\underline{\tilde{S}}(k,l) = S(k,l) - \frac{1}{K}\sum_{k=0}^{K-1}\underline{S}(k,l) \tag{13}$$

where $\mathcal{K}$ is the number of frequency points in the STFT domain.

Once we have obtained RASTA-MFCC coefficients, they are used to train a GMM model, which is defined by the mean vector μ_m, the covariance matrix Σ_m and the weights π_m of each mixture. Then, the mixture probabilities $\gamma_{l,m}$ are calculated as in (Gaubitch et al., 2013):

$$\gamma_{l,m} = \frac{\pi_m \mathcal{N}(\mathbf{c_s}(l)|\mu_m, \Sigma_m)}{\sum_{j=1}^{M}\pi_j \mathcal{N}(\mathbf{c_s}(l)|\mu_j, \Sigma_j)} \tag{14}$$

where $\mathcal{N}(\mathbf{c_s}(l)|\mu_m, \Sigma_m)$ denotes the probability density function of a multivariate Gaussian distribution.

Finally, we combine $\gamma_{l,m}$ and $\underline{\tilde{S}}(k,l)$ to obtain a weighted short-term log-spectra over all the available training set frames and thus to have the set M average clean speech log-spectra, as:

$$\overline{\underline{S}}_m(k) = \frac{\sum_{l=0}^{L-1}\gamma_{l,m}\underline{\tilde{S}}(k,l)}{\sum_{l=0}^{L-1}\gamma_{l,m}} \tag{15}$$

The average spectra of each component $\overline{\underline{S}}_m(k)$ and the parameters μ_m, Σ_m and π_m of the M-components Gaussian Mixture Model will be used to estimate the microphone response in the following part of the algorithm.

4.2.1.2 Blind microphone response estimation

The clean speech model is then used to estimate the microphone response. Again, The STFT analysis is applied to the observed audio signal $x(n)$, obtaining an N-dimensional feature vector of RASTA-MFCC coefficients $\mathbf{c_x}(l) = [c_x(0,l)c_x(1,l)\dots c_x(N-1,l)]$ and the corresponding average log-spectrum $\underline{X}(k,l)$ for each frame l. Also here, the mean of log-spectrum is subtracted.

Now, we are going to estimate the clean speech log-spectrum $\underline{\hat{X}}(k,l)$ by using the observed feature vectors $c_x(l)$ and the GMM parameters (μ_m, Σ_m, π_m) obtained during the training phase. The probabilities $\gamma'_{l,m}$ given by $c_x(l)$ from the GMM model are calculated as in Eq. (14), for each Gaussian component.

Table 2: Working settings.

Sampling rate	32 kHz
FFT points	1024
Window	Hanning
Window time	25 ms
Step time	12 ms
# Gaussian Components	64
# RASTA-MFCC coefficients	13
Record duration	6 s

These probabilities are used to estimate the average of clean speech log-spectrum for each frame as a weighted sum of clean speech log-spectrum of each Gaussian component. In formula:

$$\underline{\hat{S}}(k,l) = \sum_{m=1}^{M} \gamma_{l,m}^{'} \underline{\overline{S}}(k,l) \tag{16}$$

Finally, the microphone response is estimated assuming that $\underline{S}(k,l) \approx \underline{\hat{S}}(k,l)$ and applying Eq. (12).

4.2.2 Experimental evaluation

In the following paragraphs we evaluate the performance of video clustering algorithms based on microphone traces, by varying our working assumption from simplest, but less realistic, ones to the more demanding, but closer to real-life cases.

4.2.2.1 Settings

MATLAB([2]) has been used to implement the method described in the previous subsection. Functions such as audioread and audioinfo are used to read raw data file and file metadata, respectively. Then, PLP and RASTA-MFCC in MATLAB toolbox (Ellis,) is used for spectral analysis and MFCCs extraction. Then, MATLAB Statistics and Machine Learning Toolbox function fitgmdist is used to train the Gaussian Mixture Model, while posterior function is used to get the probabilities given a set of observed RASTA-MFCC coefficient and trained Gaussian Mixture Model. In order to train the GMM model, the VIDTimit Audio-Video dataset has been used. In particular, has been used all the recordings that has not been used to generate the source video for the controlled dataset. The same model has been used also for the live recordings dataset. In table 2 we list the several parameters that have been set, both for training and testing, to make the experimental evaluation. The choice of using a sampling rate of 32 kHz is due to the fact that this is the minimum frequency at which an audio is sampled in the overwhelming majority of smartphones. The choice of 64 components for the GMM has been suggested by literature, whereas the choice of the first 13 RASTA-MFCC is suggested as a trade-off between computational complexity and robustness against compression (Sigurosson et al., 2006), because compression is always present in case of audio extracted from video recording. The other parameters are chosen by comparing best practices from the state of art. The duration of each recording is 6 seconds.

4.2.2.2 Test run protocol

In order to test the performance of unsupervised methods on the audio feature, we devised a test run of 100 repetitions of five different clustering algorithms: K-Means, agglomerative hierarchical clustering, a GMM model fit with the EM algorithm, affinity propagation and DBSCAN. These algorithms are chosen to representatively span the different types of methods presented in the previous chapters, ranging from hard-partitioning, hierarchical, and probabilistic to stand-alone. We assume a setting of clustering problem category 2 (see Sec. 2.1), i.e. all important information about the data are supposed to be known including the number of classes $\mathcal{K}$ and its statistical behavior. For each run, we run through $K-1$ subiterations, each time testing the algorithm for all settings ranging from 2 to $\mathcal{K}$.

We have reduced the size of the raw audio feature to 512, only keeping the FFT spectrum and not its derivatives since initial tests have shown that the higher order elements do not seem to improve feature. Nonetheless, the size can put limitations on the performance, especially when run on a standard desktop PC and when considering application to huge data bases are with more complex models. A single run of K-Means does less than a minute but an application of a more complex model such as the GMM longer

([2]) © 1994-2017 The MathWorks, Inc.

up to 10-15 minutes. This is a good example of the fact the probabilistic methods can have a higher demand on computational resources.

To explore the possibilities, we have also applied dimensionality reduction methods. First, a sparse random projection method and, second, a deep autoencoder representation learning technique.

We first present a test run of only clustering between smartphone models only of different brands for both controlled and live recordings (see Sec. 4.2.2.3). Second, we present a run on all available smartphone models, even when from the same brand where we assume the difference in the microphone response to be less significant (see Sec. 4.2.2.4)

4.2.2.3 Inter-model audio clustering

The results are presented in Fig. 8 for the controlled recordings and in Fig. 9 for the live recordings. As shown, all algorithms except DBSCAN basically converge to more or less reasonble results. A noticeable degradation for the live data can be observed, but this is an expected result considering the higher levels of measurement noise for the live recordings.

It should be noted that DBSCAN has a number of highly sensitive hyperparameters and most likely not the full parameter space has been explored. We further note that the Gaussian likelihood based GMM models fare quite well, which is encouraging because those can be directly turned into a method to determine the most likely number of clusters (see Fig. 4.4) and probably is due to the fact that a Gaussian distribution is a good approximation for the noise with which the feature are distributed around their true values.

The sparse random projection method works very well and shows almost no degradation in performance, whereas the autoencoder model fares significantly worse. However, it should be noted that not the full depth of the deep learning based model was explored and it is very likely that this method can be significantly improved.

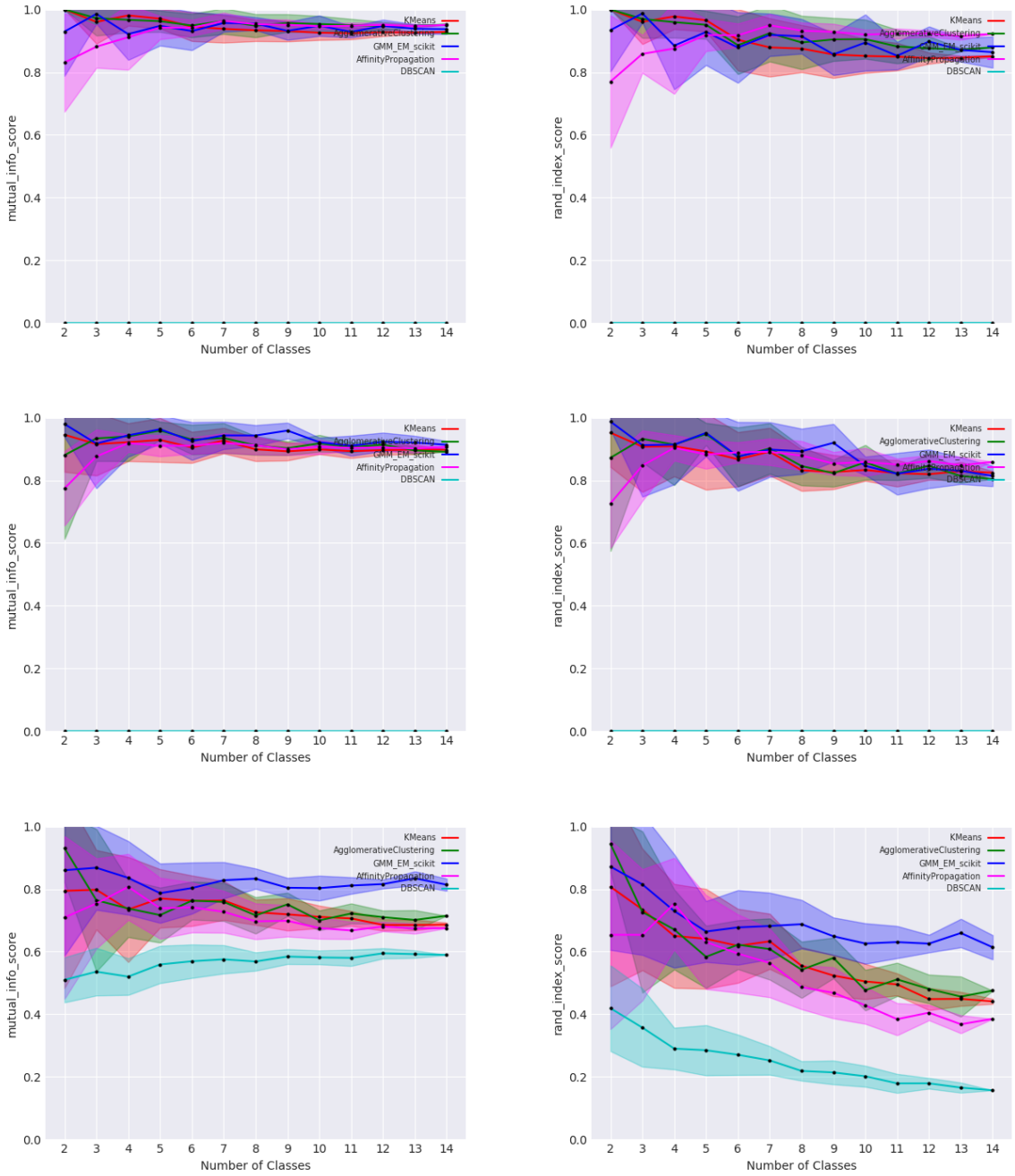


Figure 8: Results of applying five different clustering algorithms to the audio microphone feature controlled recordings. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score. First row: algorithms applied to the full feature vector of size 512. Middle row: algorithms applied to a sparse projection of the full feature vector into a size of 4. Last row: algorithms applied to a deep autoencoder projection of the full feature vector into a size of 4.

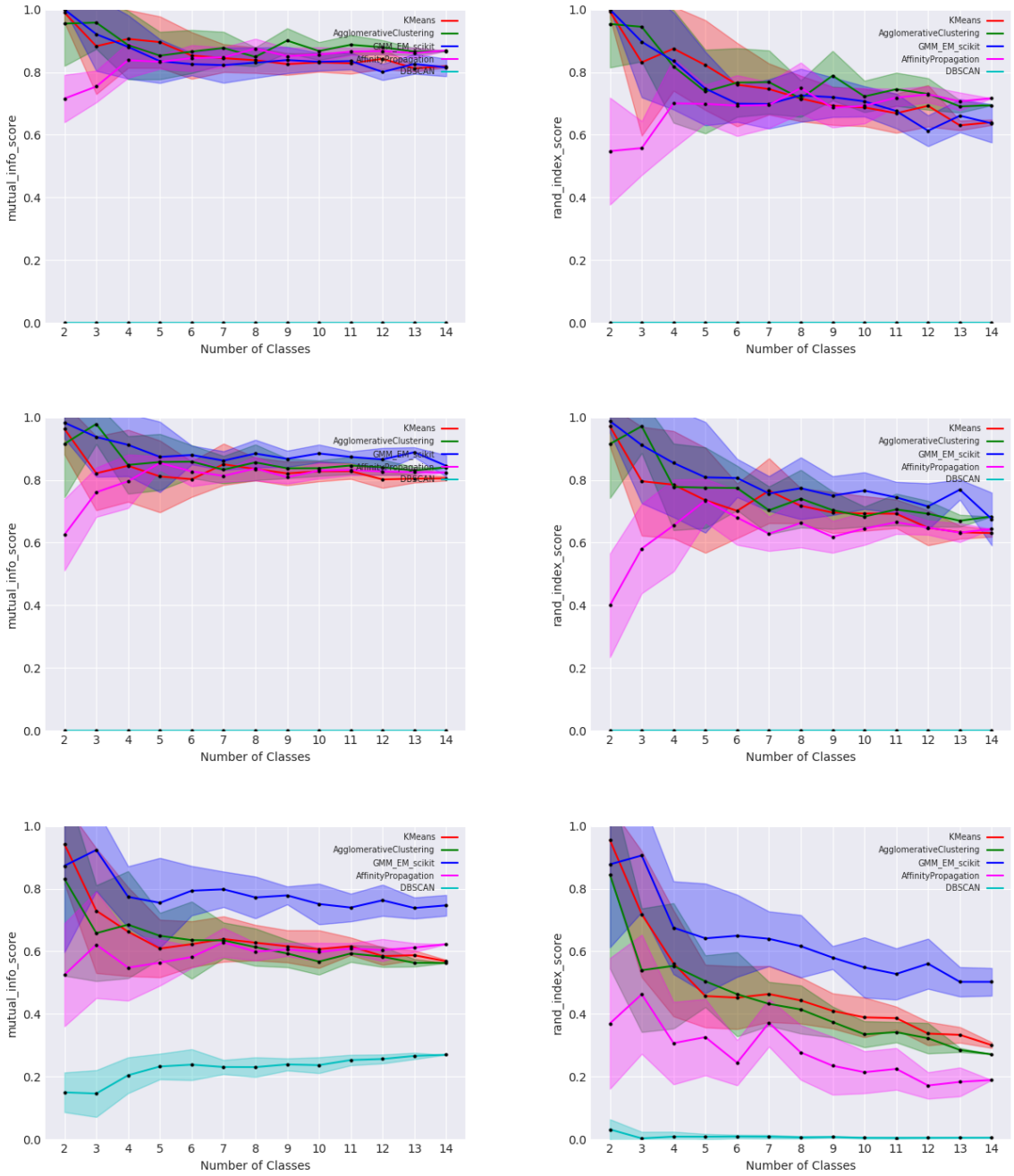


Figure 9: Results of applying five different clustering algorithms to the audio microphone feature live recordings. The data only comes from smartphone models from different brands. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score. First row: algorithms applied to the full feature vector of size 512. Middle row: algorithms applied to a sparse projection of the full feature vector into a size of 4. Last row: algorithms applied to a deep autoencoder projection of the full feature vector into a size of 4.

4.2.2.4 All-model audio clustering

The results are presented in Fig. 8 for the controlled recordings and in Fig. 9 for the live recordings. The findings are largely consistent with the test runs on the inter-model data. In comparison, an overall degradation of quality can be observed, as measured by the mutual information and rand index scores. While this is entirely expected considering the much more challenging setting, for the live recordings the quality is already quite low. It stands to reason that in such a setting, any results from the clustering can only be expected to be approximative.

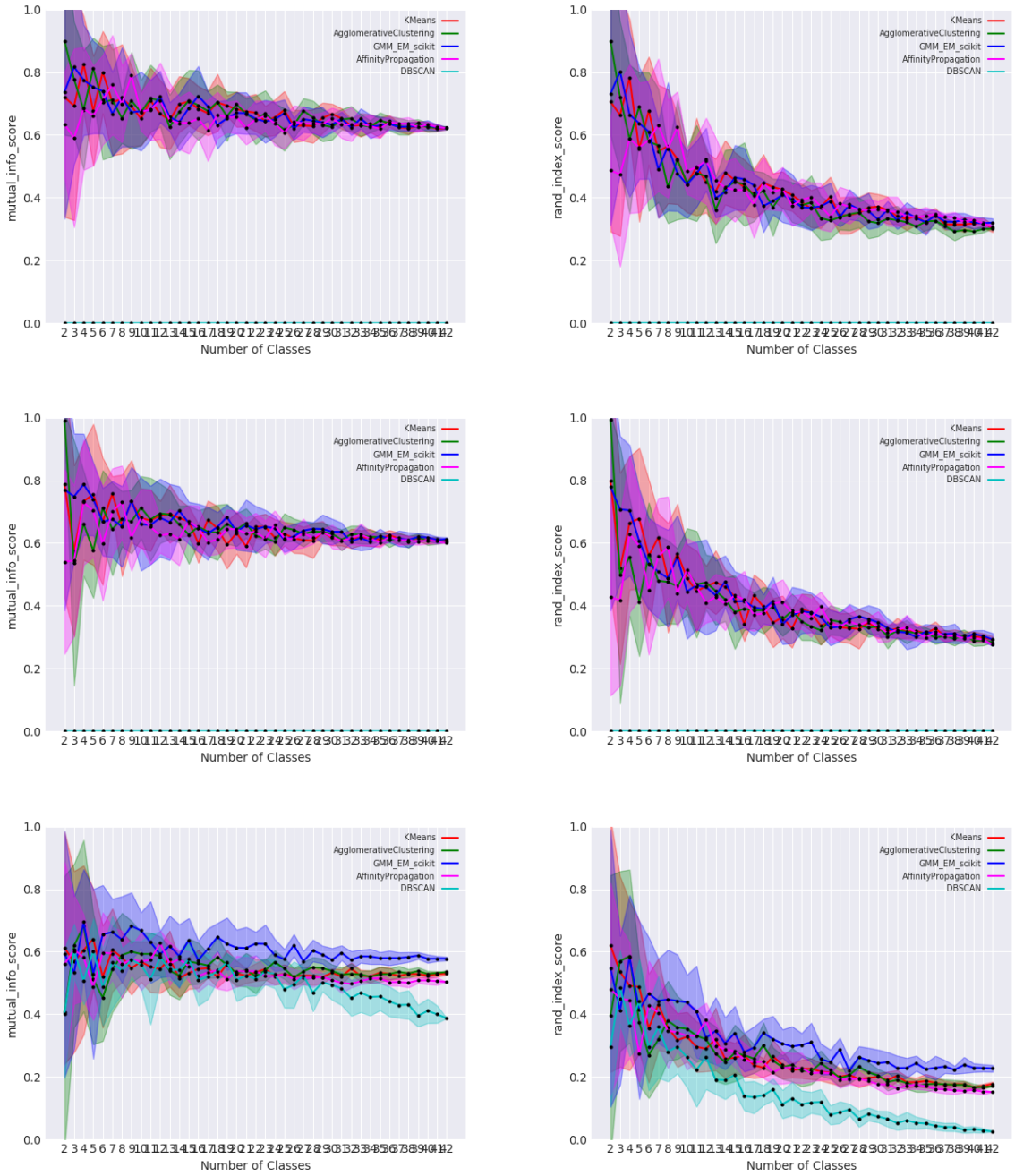


Figure 10: Results of applying five different clustering algorithms to the audio microphone feature controlled recordings. The data comes from both smartphone models from different and the same brands. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. $\mathrm{Left\ column}$: Mutual Information Score. $\mathrm{Right\ column}$: Rand Index Score. $\mathrm{First\ row}$: algorithms applied to the full feature vector of size 512. $\mathrm{Middle\ row}$: algorithms applied to a sparse projection of the full feature vector into a size of 4. $\mathrm{Last\ row}$: algorithms applied to a deep autoencoder projection of the full feature vector into a size of 4.

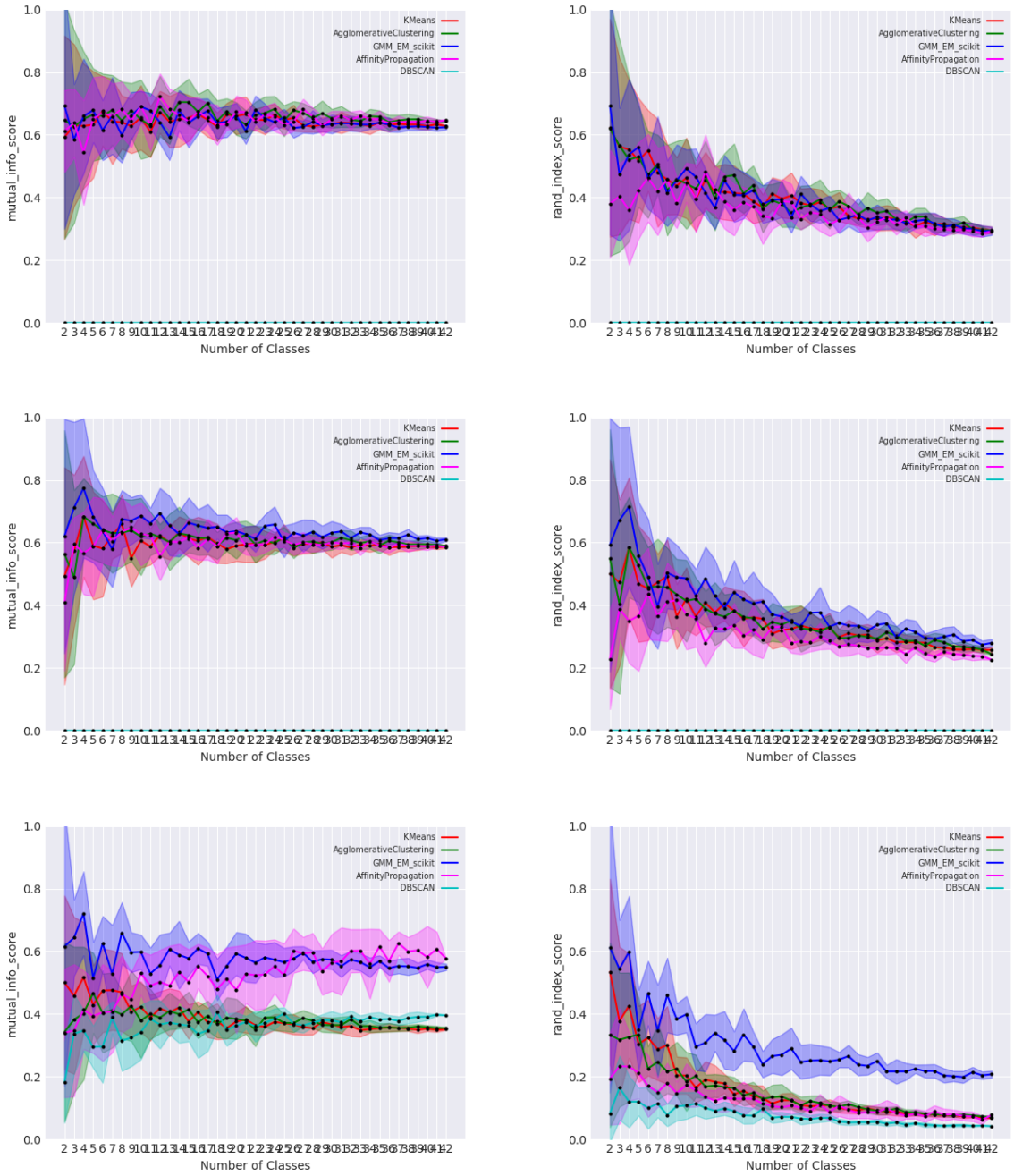


Figure 11: Results of applying five different clustering algorithms to the audio microphone feature live recordings. The data comes from both smartphone models from different and the same brands. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score. First row: algorithms applied to the full feature vector of size 512. Middle row: algorithms applied to a sparse projection of the full feature vector into a size of 4. Last row: algorithms applied to a deep autoencoder projection of the full feature vector into a size of 4.

4.3 Image-based clustering of video recordings

Sensor Pattern Noise (SPN) in multimedia forensics is analogous to the unique striations and markings, left behind on the bullet as it passes through the gun's barrel, in real forensic ballistic. SPN is attributed to slight imperfections in the manufacturing of individual sensors of digital imaging devices such as cameras, camcorders and scanners, which produce a unique fingerprint (also called Photo Response Non Uniformity - PRNU). SPN is a bi-dimensional multiplicative type of noise, commonly modeled as a zero-mean white Gaussian noise. Formally, the output of an imaging sensor can be modeled as follows

$$I = I^{(0)} + I^{(0)}K + \Theta \tag{17}$$

where I is the output image, $I^{(0)}$ is the noiseless image which would be acquired in ideal condition, K is the SPN and Θ is a generic noise term that embeds all possible noise contribution, such as dark current, shot noise, read-out noise and quantization noise.

4.3.1 SPN Features Extraction

In this subsection, we present the method we used for SPN extraction from still images and video intra-frames([3]).

4.3.1.1 Noise extraction in DWT domain.

For each color channel RGB of a still image or video I-frame, the noise W was extracted using the Daubechies wavelet transform $\mathbf{DB_4^8}$, where the index 8 means the Daubechies wavelet order and 4 is the number of scales of wavelet decomposition. The transformed image $\mathbf{DB_4^8}$ coefficients are calculated using Eq. 18 as

$$b_{i,j} = \mathbf{DB}_4^8(I_{i,j}) \tag{18}$$

where here and after the indexes i, j refer to the pixel coordinates. Then, a Wiener-type [...] filter F_W based on the minimum of local variances σ_{min} for a series of S by S pixels neighborhoods (namely 3×3, 5×5, 7×7 and 9×9), was applied over the $\mathbf{DB}_4^8$ coefficients for each level (1-4), as described by the following Eq. 19 and Eq. 20.

$$\sigma^2(i,j) = max\{0, \frac{b_{i,j}U_S}{S^2}\} \tag{19}$$

$$\sigma_{i,j}^{min} = min\{\sigma_3^2, \sigma_5^2, \sigma_7^2, \sigma_9^2\} \tag{20}$$

where $*$ is the convolution operator and U_s is a unit matrix of size $S \times S$. The noise wavelet coefficients $b^{'}$ were estimated applying the Wiener filer in the DWT domain as formalized in Eq. 21 and Eq. 22.

$$F_{W_{i,j}} = \frac{(\sigma_{i,j}^{min})^2}{(\sigma_{i,j}^{min})^2 + \sigma_0^2} \tag{21}$$

$$b_{i,j}^{'} = b_{i,j}F_{W_{i,j}} \tag{22}$$

At the end, the image noise is extracted applying the inverse Wavelet Transform $\mathbf{DB}_4^8$ in Eq 23.

$$W_{i,j} = \{\mathbf{DB}_4^8\}^{-1}(b_{i,j}^{'}). \tag{23}$$

4.3.1.2 Attenuation of saturated pixels.

As SPN noise is multiplicative [...], the correlation is higher in textured areas and, if the image pixels are saturated, then no SPN is detected. Therefore, the image has been pre-processed with a saturation mask f_s where all pixels in a neighborhood having intensity greater than a threshold T_s are set to zero according to Eq. 24, and f_I is an attenuation function given by Eq. 25, where T_c and σ_C are the attenuation threshold and variance, respectively.

$$f_S(I;T_S) = \begin{cases} 0 & , \quad I > T_S \\ 1 & , \quad I \leq T_S \end{cases} \tag{24}$$

$$f_I(I;T_C;\sigma_C) = \begin{cases} e^{-\frac{(I-T_C)^2}{\sigma_C}} & , \quad I > T_C \\ I/T_C & , \quad I \leq T_C \end{cases} \tag{25}$$

The image after pre-processing, Eq. 26 called I_F is:

$$I^{'} = f_S(I;T_S)F_I(I;T_C;\sigma_C) \tag{26}$$

([3]) intra-frames or I-frames in a compressed video contain the main visual information, whereas B-frames and P-frames contain only information relevant to motion changes

	Still Images	Live Video Recordings
Number of devices	5	41
Identical devices	Yes	Yes
Media Type	JPEG	MPEG/MOV/3GP
Duration	*N.D.*	$1-2$ min
Crop size	1024×1024	480×480
Location	Central crop	Central crop
SPN content	Natural	I-frames

Table 3: Experimental settings for SPN extracted from still images and videos.

4.3.1.3 Estimate SPN using Maximum Likelihood Estimator.

Then K can be estimated from a set of L images or I-frames by using the Maximum Likelihood Estimator (MLE) by Eq. 27, where c_1, c_2 and c_3 are the color channels RGB.

$$K^{c_n} = \frac{\sum_{l=1}^{L} W_l I_l'}{\sum_{l=1}^{L} (I_l')^2} \tag{27}$$

4.3.1.4 SPN normalization.

The noise term extracted in the previous step usually contains periodic signals, also known as linear patterns, due to the in-camera image processing such as Color Filter Array interpolation, row-wise and column wise operations etc. One way to mitigate this effect is to alter K such as to have a Zero Mean (ZM) for each row and column respectively [...] as described in Eq. 28:

$$K_{ZM} = f_{ZM}(K_{i,j}^c) = K_{i,j}^c - \frac{\sum_{i=1}^{M} K_{i,j}^c}{M} - \frac{\sum_{j=1}^{N} K_{i,j}^c}{N} + \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} K_{i,j}^c}{MN} \tag{28}$$

4.3.1.5 Convert SPN to grayscale.

SPN extracted from each of the corresponding image channels (RGB) is converted to a grayscale by forming a weighted sum of c_1, c_2 and c_3 components as:

$$K_{gray} = 0.2989 K_{ZM}^{c_1} + 0.5870 K_{ZM}^{c_2} + 0.1140 K_{ZM}^{c_3} \tag{29}$$

4.3.1.6 Wiener filtering for JPEG compression artifacts removal.

In order to further filter out JPEG compression (also known as 'blockiness') artifacts which contaminate the extracted SPN, a low pass attenuation filter is applied, performing Wiener filtering in the Fourier domain [...] over to K_{gray} derived from the previous step, as:

$$K_{FFT} = \mathbf{FFT^{-1}}(F_W(\mathbf{FFT}(K_{gray}))) \tag{30}$$

4.3.2 Experimental settings

In this subsection, we point out the experimental settings we used to extract SPN from media contents such as images and videos. The main settings are summarized in Table 3. Here, it is worth to note that in case of video footage, the SPN is created by averaging the SPN extracted from I-frames as described in Subsection 4.3.1. By considering that most of the video encoders embed an I-frame per second, the number of I-frames varies between 60 and 120, depending on the video duration.

4.3.3 Results on still images

In order to test the performance of unsupervised methods on the SPN feature, we again devised a test run of 100 repetitions of five different clustering algorithms: K-Means, agglomerative hierarchical clustering, a GMM model fit with the EM algorithm, affinity propagation and DBSCAN. These algorithms are chosen to representatively span the different types of methods presented in the previous chapters, ranging from hard-partitioning, hierarchical, and probabilistic to stand-alone. We assume a setting of clustering problem category 2, i.e. all important information about the data are supposed to be known including the number of classes K and its statistical behavior. For each run, we run through $K-1$ subiterations, each time testing the algorithm for all settings ranging from 2 to K.

The size of the raw SPN feature of 1024^2 (simply concatenating all image pixels) puts severe limitations on the performance, especially when run on a standard desktop PC. A single run of K-Means might take about 5-10 minutes, an application of a more complex model such as the GMM longer, up to 20 minutes. In fact, the GMM model is also memory limited by the size of the data, since it is impossible to hold a complex covariance matrix of size 1024^4 in standard memory. For the full feature vector, we have thus restricted the GMM covariance to be sticticly diagonal. This is a good example of the fact the probabilistic methods can have a higher demand on computational resources.

In addition we again have applied a dimensionality reduction method to alleviate this problem. For the SPN feature, only the sparse projection method is used since the autoencoder representation technique cannot be straight-forwardly fit on a standard machine, again for memory restrictions. On a larger machine or using a streaming technique, a deep learning based representation should nonthelesse be calculable, in principle.

The results are presentetd in Fig. 12. As shown, all algorithms basically fail to converge on a useful solution for the SPN feature vectors. The most likely reason, so far identified, lies in the choice of the underlying distance measure. It is noticeable that in a supervised setting of problem type 1, previous research has shown that the SPN feature vector works exceptionally well. The difference is, that this work employed a corellation based distance measure. In fact, since the SPN feature itself is basically a pixel-uncorrelated white noise signal, a direct value-based distance measure is likely to be very noisy itself. We have tested this assumption by also using a correlation distance measure and then employing a spectral method to project the correlation distance matrix into its eigenbasis and then applied K-Means in that space. The promising results are shown in Fig. 13.

Unfortunately, as for now this severely restricts the options for applying a clustering method to the SPN feature vectors. Other than for the microphone features, we only successfully can use the correlation-based spectral method in the moment. This especially restricts the options for further development of the SPN clustering into a more autonomous method, for example excludes a simple application of a statistical model comparison method as presented in Sec. 4.4.

4.3.4 Results on video frames

For the full video frame data, we have employed exactly the same approach is for the still images. The application of standard methods yield similarly negative results as in the previous section. We thus directly present the results of the correlation distance based spectral method in Fig. 14. The quality of the results is much degraded to the test case with still images, again en entirely expected result. Still, the spectral clustering method yields acceptable results, considering the challenging setting of the video frames. The applied compression and interpolation routines are expected to degrade the quality of the SPN features.

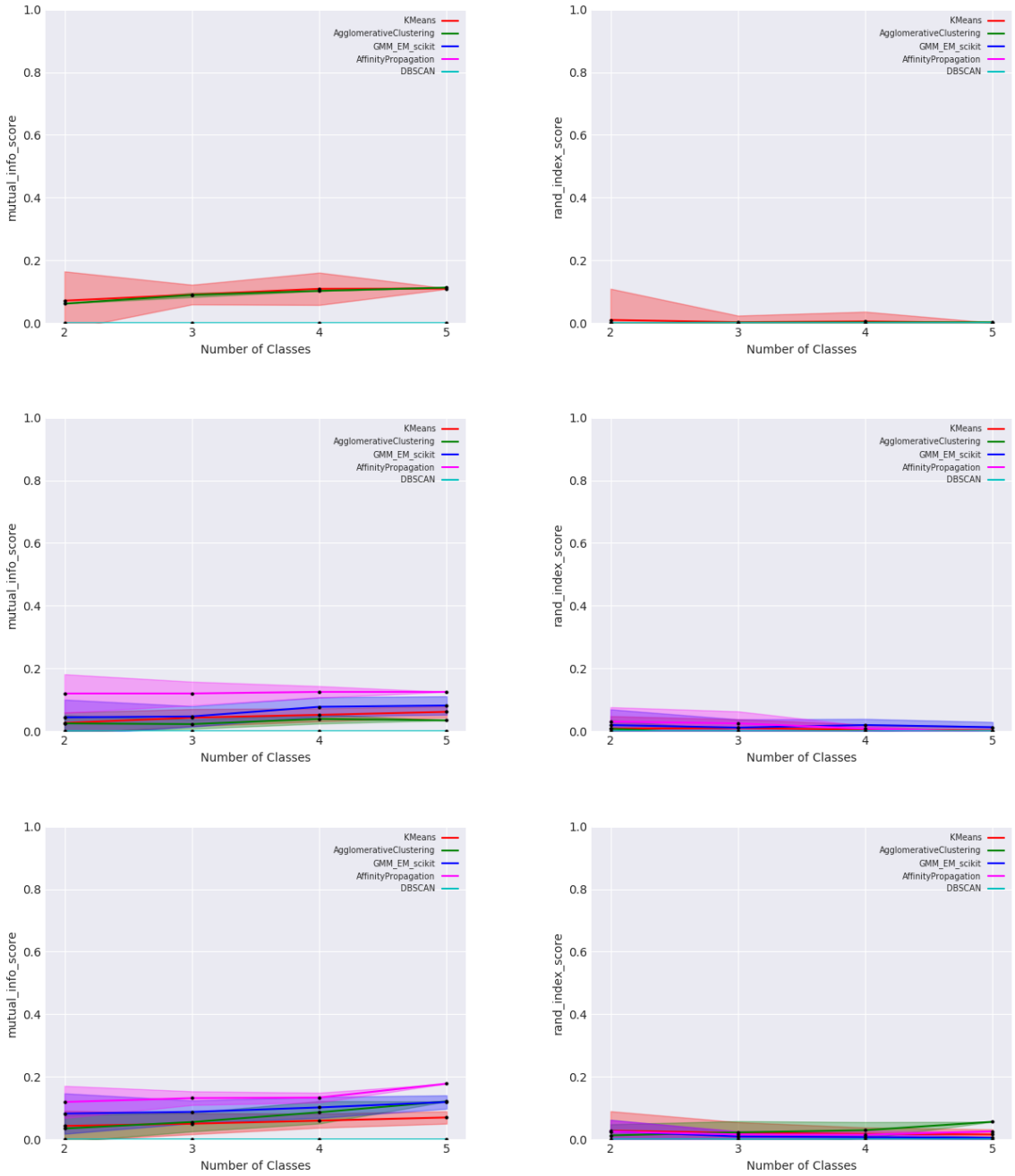


Figure 12: Results of applying five different clustering algorithms to the SPN still image feature data. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score. First row: algorithms applied to the full feature vector of size 1024^2. Middle row: algorithms applied to a sparse projection of the full feature vector into a size of 16. Last row: algorithms applied to a sparse projection of the full feature vector into a size of 8.

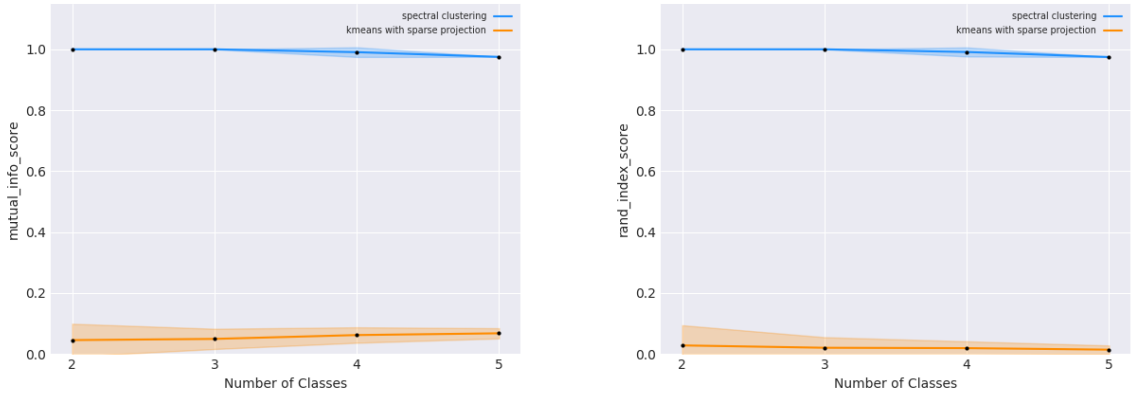


Figure 13: Results of using a spectral method to project a correlation distance matrix into its eigenbasis and then applying K-Means in that space. Results are compared to simply using K-Means on the untransformed data, de facto assuming an euclidean distance metric. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score.

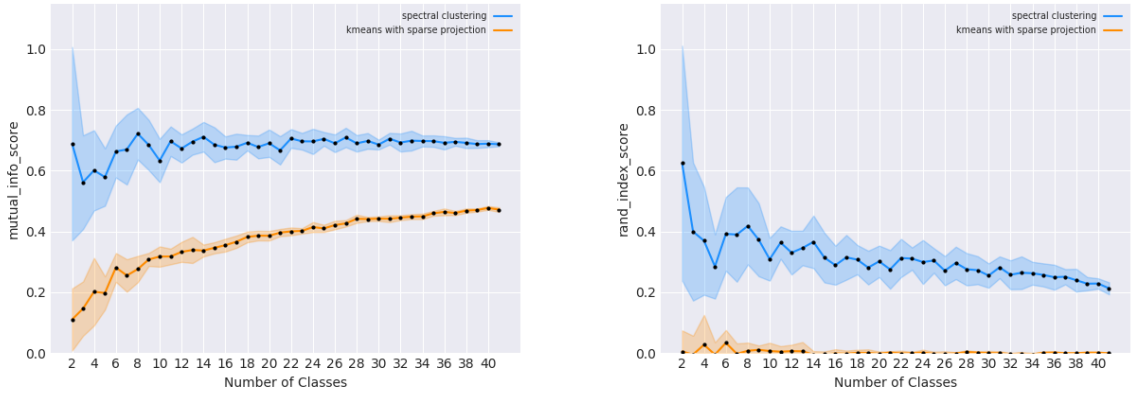


Figure 14: Results of using a spectral method to project a correlation distance matrix into its eigenbasis and then applying K-Means in that space. Results are compared to simply using K-Means on the untransformed data, de facto assuming an euclidean distance metric. In total 100 repeated runs have been conducted and the shaded regions mark the variance in performance of the different methods as measured by cluster validity indices. Left column: Mutual Information Score. Right column: Rand Index Score.

Table 4: Performance on controlled recordings data set. For each data dimension, the mean values and the standard deviation of estimated number of clusters, the corresponding Adjusted Rand Index and Adjusted Mutual Information are. For each data size, 100 runs are performed.

	Data dimensions					
	2	4	8	16	32	64
	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
$\hat{k}$	8.56 ± 0.98	**16.03 ± 1.50**	27.68 ± 3.91	27.22 ± 4.06	26.39 ± 4.18	27.72 ± 4.36
ARI	0.42 ± 0.04	**0.83 ± 0.01**	0.74 ± 0.04	0.77 ± 0.04	0.77 ± 0.04	0.78 ± 0.04
AMI	0.72 ± 0.01	**0.90 ± 0.01**	0.88 ± 0.01	0.89 ± 0.01	0.89 ± 0.01	0.90 ± 0.01

4.4 Explorative Case: Model based Clustering with unknown number of classes

As an illustrative case, here we take a first step into moving our application into problem type 3. We aim at using a natural extension of a probabilistic clustering method to use a statistical model comparison to estimate the likely number of true classes $\mathcal{K}$. We present a method that relies on the GMM-EM approach that we could successfully apply to the microphone data in Sec. 4.2.

The approach progresses as follows. The cluster problem is again represented by the Gaussian mixture likelihood:

$$\mathcal{P}(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left[\sum_{c=1}^{\mathcal{C}} \pi_c \mathcal{N}(x_n|\mu_c,\Sigma_c) \right] \tag{31}$$

where $\mathcal{K}$ denotes the number of classes, and π_c, μ_c, Σ_c denote respectively the $1-of-c$ class vector, the class mean and the class covariance matrix. Now, for any suitably pre-chosen range of possible class numbers $\tilde{\mathcal{K}}$, the problem is solved $\tilde{\mathcal{K}}$ times using the EM algorithm. Finally, a choice is made between all $\tilde{\mathcal{C}}$ using the fitted log-predictive likelihood densities adjusted for overfitting by a statistical information criterium, in our case the Bayesian Information Criterium (BIC). The BIC adds a penalty term to the likelihood, derived from an assumption about an asymptotical Gaussian posterior distribution to the likelihood. This acts to regularize the well known tendency of a maximum likelihood approach to overfit its parameters. The procedure effectively amounts to minimizing the BIC as a function of the model likelihood depending on all class models $\tilde{\mathcal{C}}$ to get an estimate $\hat{k\mathcal{C}}$ for the most likely number of classes:

$$\hat{\mathcal{C}} = \text{argmin}_{c\in\tilde{\mathcal{C}}} \left[BIC_c \right] \tag{32}$$

The BIC is known to be more conservative and thus favoring smaller models than other information criteria. In our tests and simulations, we observed the best performance for our data using the BIC. For a detailed derivation and differences between the BIC and other criteria, see (Gelman et al., 2004).

Of course, applied in a fully unsupervised fashion without any further prior information on $\mathcal{K}$, it is not expected that this approach results in perfect matches. Our goal is to see how far a statistical model comparison can be run autonomously from any specific user input in order to reach a reasonable and robust estimate $\hat{\mathcal{K}}$. The exploration of other and more involved statistical approaches or the inclusion of qualitative prior knowledge, for instance from an ongoing forensic investigation, could be part of future work.

4.4.1 Model Comparison Results

Experiments were carried on two different videos sets. For both of them, the number of the source device is 14, meanwhile the number of samples is 434 for controlled recordings and 696 for live recordings data set. For each of them, we performed the clustering process by varying the compressed size of the feature vectors as we described in 4.2 and then take 100 runs per size. We clustered compressed feature vectors whose sizes vary in $\{2, 4, 8, 16, 32, 64\}$. In Figure 15, we show how BIC score varies for different number of classes tested at a generic algorithm run, for different sizes of feature vectors. We can observe that for low dimensional features the BIC curves keeps a convex shape, whose minimum is close to the right number of classes, whereas for high dimensional features this desired behavior decays.

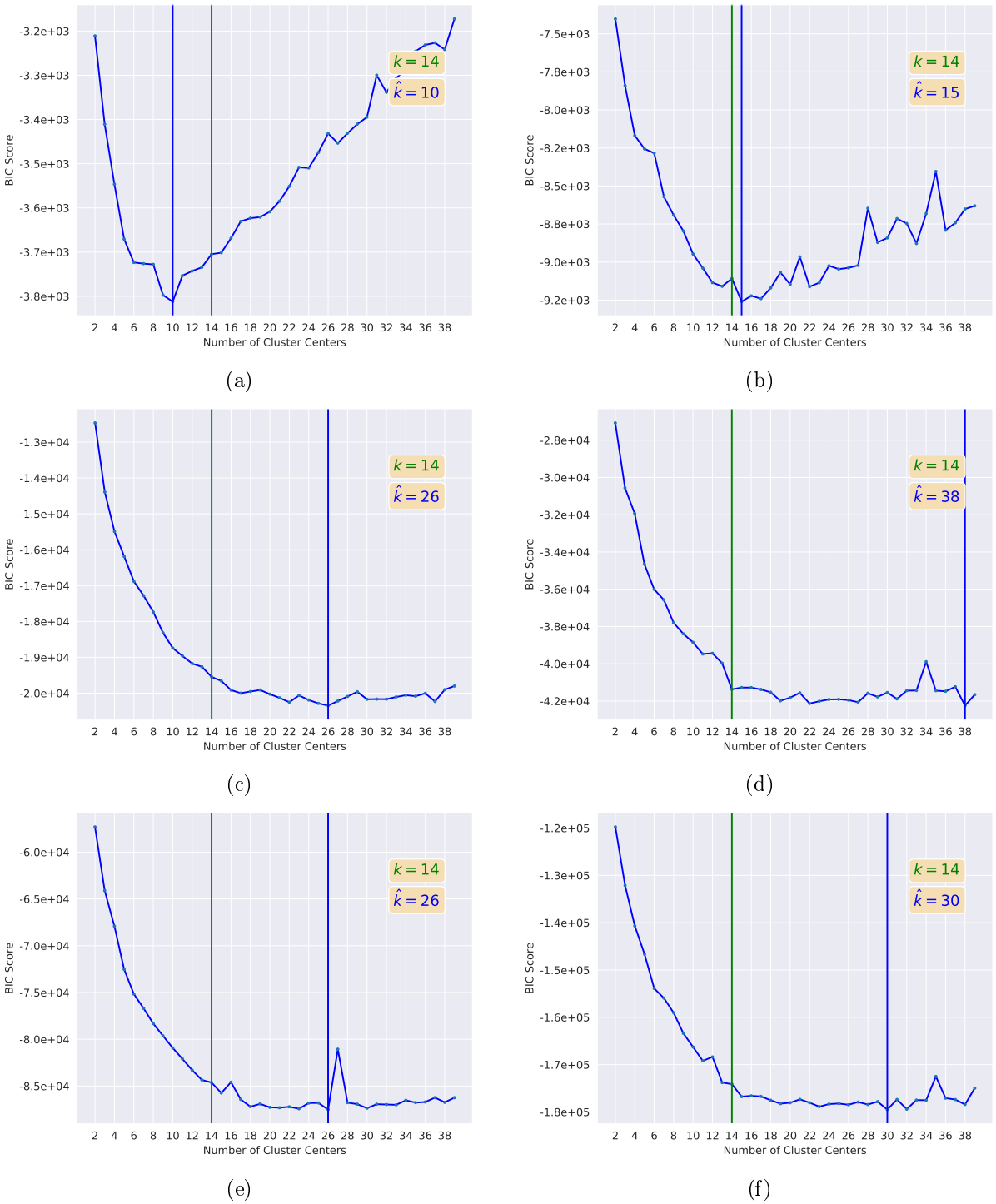


Figure 15: Examples of BIC behavior in function of the number of cluster tested at a generic clustering run. The curves are related to 2-D (a), 4-D (b), 8-D (c), 16-D (d), 32-D (e) and 64-D (f).

Table 5: Performance on live recordings data set. For each data dimension, the mean values and the standard deviation of estimated number of clusters, the corresponding Adjusted Rand Index and Adjusted Mutual Information are presented. For each data size, 100 runs are performed.

	Data dimensions					
	2	4	8	16	32	64
	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
$\hat{k}$	11.53 ± 1.53	**21.62 ± 2.25**	35.71 ± 2.33	36.23 ± 2.35	36.14 ± 2.14	36.25 ± 2.16
ARI	0.52 ± 0.04	**0.70 ± 0.03**	0.57 ± 0.03	0.56 ± 0.03	0.55 ± 0.02	0.56 ± 0.02
AMI	0.73 ± 0.02	**0.86 ± 0.01**	0.84 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.84 ± 0.01

A comprehensive description of the clustering performance is shown in Table 4, for the controlled recordings data set, and in Table 5, for the live recordings data set. By comparing them, we observe that in both cases the maximum of the accuracy in clustering is obtained with 4-D reduced features. Moreover, for the controlled data set, the performance are better in average, both in terms of estimated number of clusters and ARI and AMI. It can be seen that, even in a small explorative test case, it is possible to estimate the number of classes $\mathcal{K}$ approximately using the BIC model comparison method. The order of magnitude can surely be estimated with some confidence. For the controlled data, with the right representational model, the estimate even comes relatively close to the ground truth. Note that this method does not require any more input or prior information than a broad range of cluster centers to be tested; nor does it require any further interaction by the user.

Finally, we perform also a qualitative evaluation of the clustering, by showing in Figure 16 and Figure 17 the projected true distributions of the data (in their original space) and the ones obtained after clustering. The scatter plots are obtained by means of t-sne high dimensional data visualization tool. Please note that the t-sne algorithm does not display the true distribution of the data, but only a non-linear projection that should only preserve groupings of data, while distances are meaningless. The examples of clustering outputs are obtained by using 4-D feature in both cases.

While for the controlled recordings set the number of $\hat{K}$ is reasonably close to the right number of clusters ($\mathcal{K} = 14$), in Figure 17 the data seem to be over-partitioned. However, in both situations it is worth to note that there is a limited "contamination" between the detected clusters. This means that, even in case in which the estimated number of clusters is not close to the right one, the clusters are split in homogeneous sub-clusters. Nonetheless, using the t-sne algorithm can only provide a rough qualitative assessment and we leave the discussion of the usefullness of high dimensional visualization tools to further work.

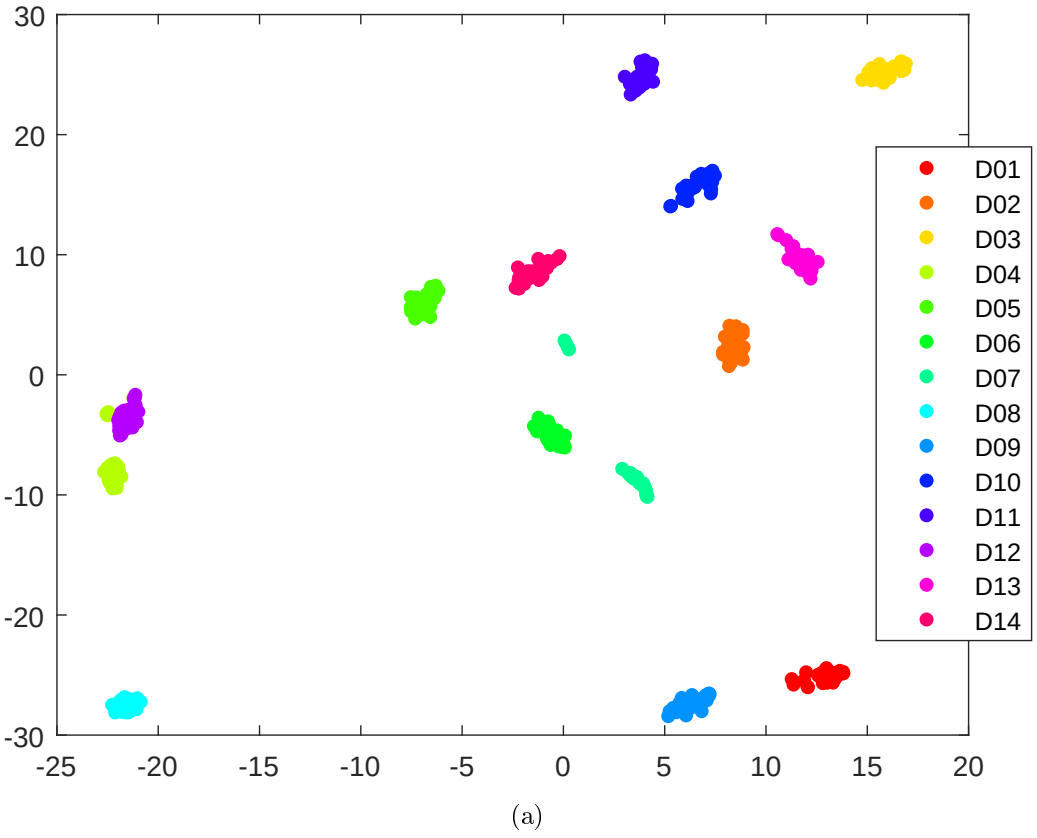
4.5 Conclusions of the applications case

We can draw some first lessons from the presented application case on SPN and microphone response data that will guide our overall report conclusion.

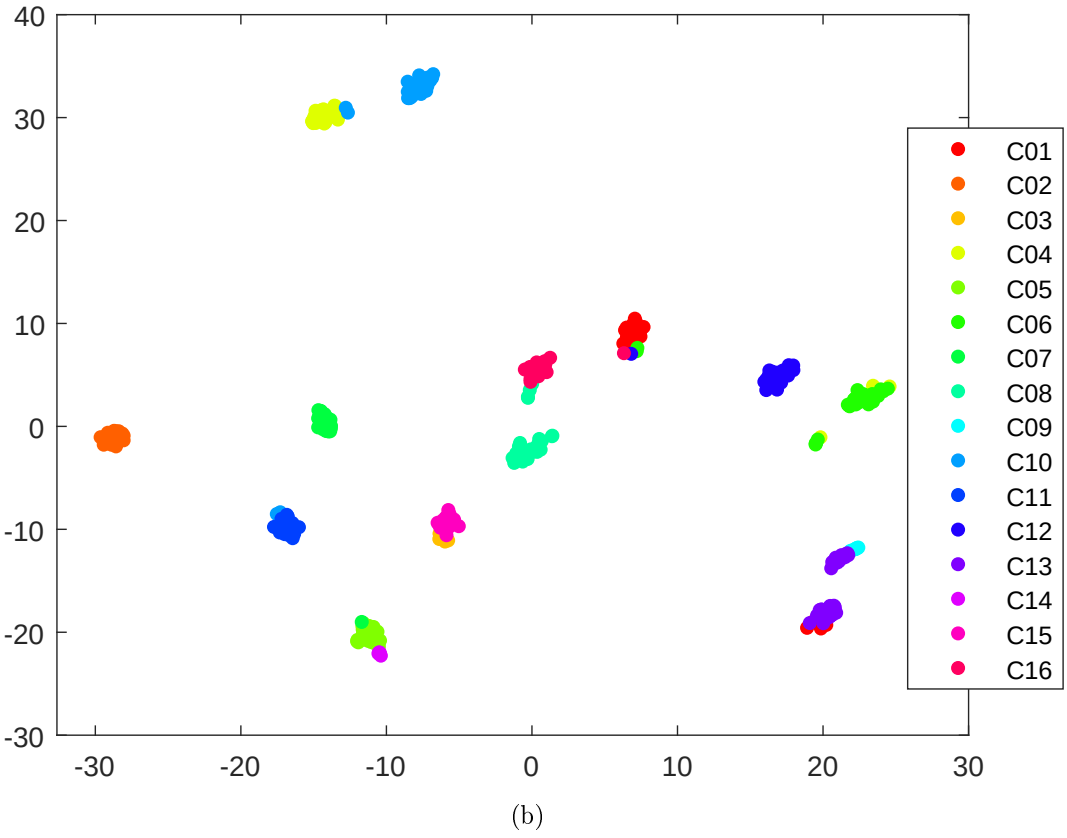
From our initially identified list of cluster challenges (see 2.3), we can report that some are indeed of particular importance for digital video forensics. Even more when considering the specific investigative circumstances we deemed likely in which not much prior information should be considered available and only a minimum of user interaction should be expected.

The central lessons are:

— No standard method: Even for two relatively similar application cases as in analyzing various video features such as images and audio tracks, we cannot assume to rely on a single standard method to perform best. Even for our tests under the unrealistic assumptions of clustering problem type 2 (see Sec. 1.2), we can conclude that for the microphone features Gaussian likelihood-based methods such as K-means or GMMs work well, but not at all for the SPN-derived features. Probably because of the specific nature of the SPN as random and uncorrelated noise pixels, a shape-based distance measure works better, which we showed by applying a correlation distance based spectral clustering method with more success.

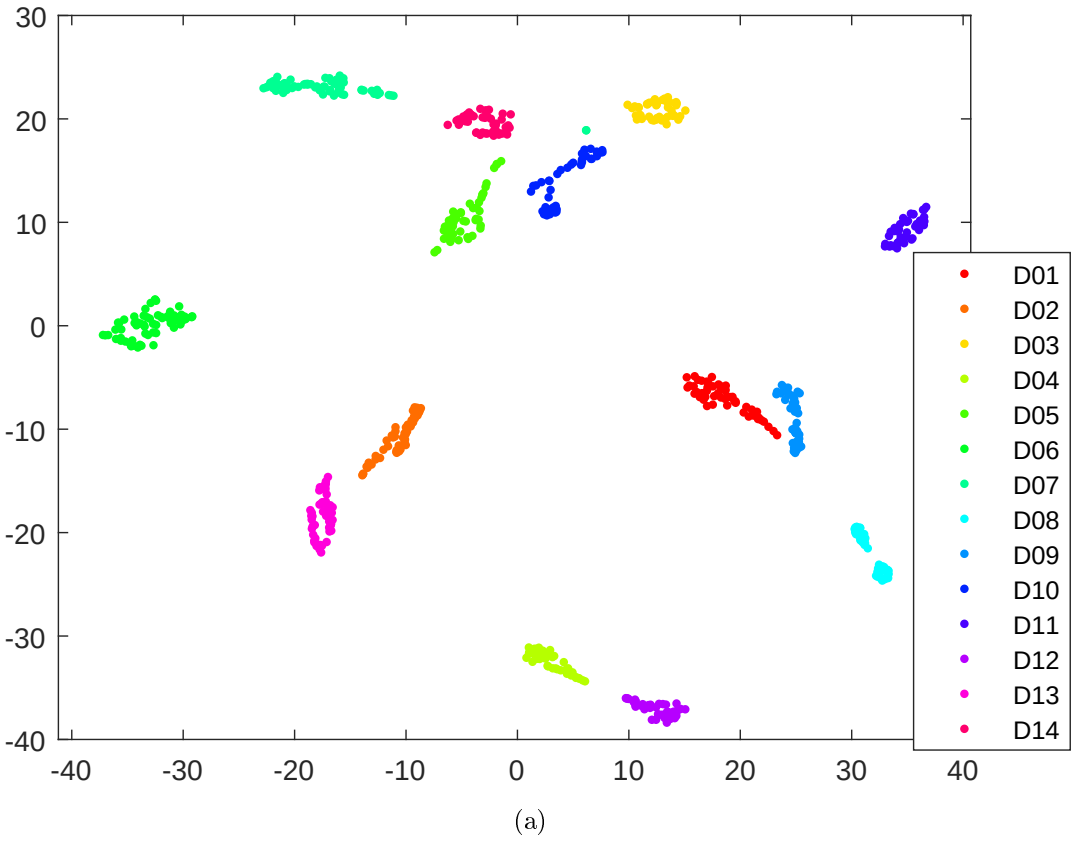


(a)

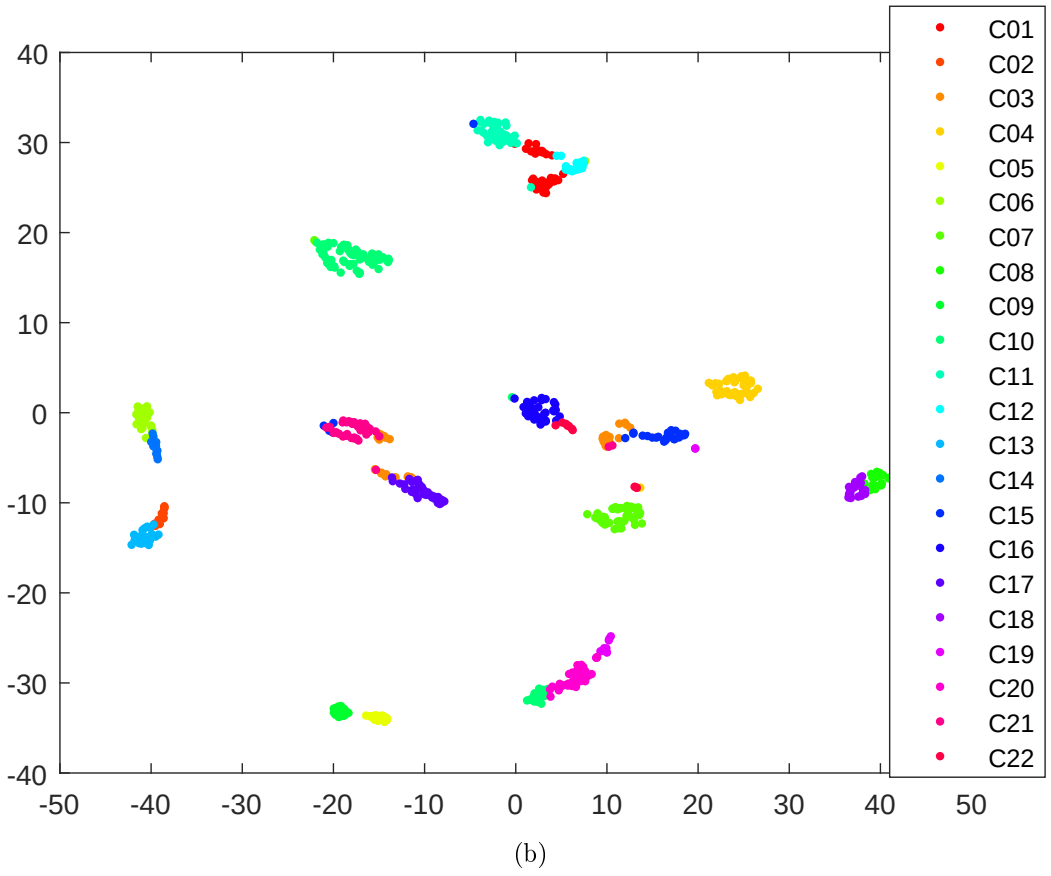


(b)

Figure 16: Scatter plots of controlled recordings data points using t-sne. In (a) the projected ground truth distribution is shown, in (b) the predicted classes using a 4-d feature vectors.



(a)



(b)

Figure 17: Scatter plots of live recordings data points using t-sne. In (a) the projected ground truth distribution is shown, in (b) the predicted classes using a 4-d feature vectors.

— Unsufficient algorithmic autonomy of standard methods: the amount and significance of hyperparameters and information needed to successfully run a standard clustering method, even under the simplified assumptions of our clustering problem category 2 is very high. This presents severe limitations to foresee using a clustering in a setting where a more autonomous method is needed while less prior information is available, such as in problem category 3 or 4. The two most immediate problems are the handling of the number of clusters $\mathcal{K}$ and the need for hyperparameter optimization methods. We have shown that the number of clusters $\mathcal{K}$ can possibly be handled using a statistical model comparison, at least for the microphone case.

— Data Dimensionality: The size and dimension of the tested data can become prohibitively large to simply apply standard clustering methods. This is especially true for the image based SPN data. It will likely be a problem for many digital forensic applications. There is, thus, a need to identify suitable methods of dimensionality reduction and understand their effect on the performance of the clustering.

— Non-expert user interaction: methods from clustering and unsupervised machine learning are very complex and often need manual intervention of expert users to yield optimal results. Visual inspection of results would be a solution, but is rendered almost impossible for high dimensional data such as the SPN and microphone feature vectors. Whether we can rely on projection-based data visualization techniques such as T-SNE is doubtful because of the non-deterministic, non-trivial interpretation of their results. In general, it seems necessary to find a way of how to include domain knowledge a priori.

5 Outlook and Next Activities

From the overall conclusions of our application tests in Sec. 4.5, we can derive options to move forward. As identified through the state-of-art background analysis (see Sec. 2.3) and the clustering workshop (see Sec. 3section), we have seen that the potential list of challenges and issues is large and cannot be handled exhaustively at once. It also became clear from the workshop that to move into real applications, it is desirable to design a clear real forensic test scenario together with investigators, for instance at EUROPOL. This is, firstly, important to investigate whether the achievable quality of clustering results, as showcased in Sec. 4section, is sufficient for a real case, and, secondly, to explore ways of interaction with investigators. The choice of this scenario should ideally be guided by a combination of the needs of investigators and the existing algorithmic capabilities to solve the known challenges.

We propose two different principal routes to move the clustering applications forward from this. It should be clear that we do not regard this as a clear distinction as it is very likely that both will be needed to some degree in a real forensic setting. Both have been partly mentioned by EUROPOL representatives at our workshop.

— The challenge of "Big Data". the focus of this route is on tackling data dimensionality issues and considers scalability and fastness of algorithms as a major challenge, especially for the analysis of large sets of images. Within this paradigm, we likely will have to assume a problem of class 3 or 4 and would have to rely more on approximate methods, qualitative results and solutions known from data mining and database knowledge discovery. Methods that work on precomputed distances or metrics, reuse lookup tables and whose complexity is not exceeding $\mathcal{O}(n)$ should be preferred. It should be noted that this type of application could be solved better within an existing database solution of big data framework. Typical applications would be the mass unsupervised analysis of a large image database or cross-referencing archived forensic data to detect undiscovered leads.

— The challenge of algorithmic design and prior information. This route is placing the focus more on algorithmic challenges and use cases were an accurate unsupervised machinery is needed and quantitatively reliable results are desired, albeit neither much prior information is available, such as problem category 3, nor an expert data analyst is expected to available. A smaller scale data application would be ideal where a showcase could be developed, gradually moving into the direction of an "autonomous AI investigator". It could even be thoroughly investigated where unsupervised methods might be usable to move the problem into catgery 1 to apply supervised methods, for instance using forms of automated annotation. Derived from the experience of this study, two specific issues should probably be tackled first:

1. Handling the unknown number of classes $\mathcal{K}$.
2. Investigating how to include prior information especially from non-expert users and how to convey the results best to crime investigators. This also includes thinking about validity analysis and uncertainty of results.

It should be noted that the second route is more in line with the initial application study conducted in this report. The smaller scale data used in the AVICAO project suggests to some degree to continue along this route.

Nonetheless, we also emphasize that the clustering problem is somewhat more horizontal and that the basic state-of-the art of the field in principle allows us to widen the scope to different type of data then the video data from AVICAO. Applications in other forensic and investigative domains are possible, including in analyzing semantic data for password guessing or the clustering of biometric fingerprints.

References

Aharon, M., Elad, M. and Bruckstein, A., '*rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation', IEEE Transactions on Signal Processing, Vol. 54, No 11, Nov 2006, pp. 4311–4322. ISSN 1053-587X. .

Aljalbout, E., Golkov, V., Siddiqui, Y. and Cremers, D., 'Clustering with deep learning: Taxonomy and new methods', 01 2018.

Amerini, I., Caldelli, R., Del Mastio, A., Di Fuccia, A., Molinari, C. and Rizzo, A. P., 'Dealing with video source identification in social networks', Signal Processing: Image Communication, Vol. 57, 2017, pp. 1–7. ISSN 0923-5965. . URL http://www.sciencedirect.com/science/article/pii/S0923596517300759.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U., 'When is "nearest neighbor" meaningful?', In 'Proceedings of the 7th International Conference on Database Theory', ICDT '99. Springer-Verlag, London, UK, UK. ISBN 3-540-65452-6, pp. 217–235.

Bishop, C. M., 'Pattern recognition and machine learning (information science and statistics)', Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Blei, D. M., Ng, A. Y. and Jordan, M. I., 'Latent dirichlet allocation', J. Mach. Learn. Res., Vol. 3, Mar. 2003, pp. 993–1022. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.

Buchholz, R., Kraetzer, C. and Dittmann, J., 'Microphone classification using fourier coefficients', In 'Information Hiding', , edited by S. Katzenbeisser and A.-R. SadeghiSpringer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-04431-1, pp. 235–246.

Caldelli, R., Amerini, I., Picchioni, F. and Innocenti, M., 'Fast image clustering of unknown source images', In '2010 IEEE International Workshop on Information Forensics and Security', ISSN 2157-4766, pp. 1–5. .

Chen, M., Fridrich, J., Goljan, M. and Lukáš, J., 'Source digital camcorder identification using sensor photo response non-uniformity', In 'Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX', 65051G. .

Chen, S., Pande, A., Zeng, K. and Mohapatra, P., 'Live video forensics: Source identification in lossy wireless networks', IEEE Transactions on Information Forensics and Security, Vol. 10, No 1, Jan 2015, pp. 28–39. ISSN 1556-6013. .

Chuang, W. H., Su, H. and Wu, M., 'Exploring compression effects for improved source camera identification using strongly compressed video', In '2011 18th IEEE International Conference on Image Processing', ISSN 1522-4880, pp. 1953–1956. .

Cortes, C. and Vapnik, V., 'Support-vector networks', Machine Learning, Vol. 20, No 3, Sep 1995, pp. 273–297. ISSN 1573-0565. . URL https://doi.org/10.1007/BF00994018.

Cuccovillo, L. and Aichroth, P., 'Open-set microphone classification via blind channel analysis', In '2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 2074–2078. .

Cuccovillo, L., Mann, S., Aichroth, P., Tagliasacchi, M. and Dittmar, C., 'Blind microphone analysis and stable tone phase analysis for audio tampering detection', In 'Audio Engineering Society Convention 135', URL http://www.aes.org/e-lib/browse.cfm?elib=17016.

Cuccovillo, L., Mann, S., Tagliasacchi, M. and Aichroth, P., 'Audio tampering detection via microphone classification', In '2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)', pp. 177–182. .

Das, A., Borisov, N. and Caesar, M., 'Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components', In 'Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security', CCS '14. ACM, New York, NY, USA. ISBN 978-1-4503-2957-6, pp. 441–452. . URL http://doi.acm.org/10.1145/2660267.2660325.

Defays, D., 'An efficient algorithm for a complete link method', The Computer Journal, Vol. 20, No 4, 01 1977, pp. 364–366. ISSN 0010-4620. . URL https://doi.org/10.1093/comjnl/20.4.364.

Dempster, A., Laird, N. and Rubin, D., 'Maximum likelihood from incomplete data via the em algorithm', Lecture Notes in Computer Science, Vol. 39, No 1, 1977, pp. 1–38.

Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K. and Shanahan, M., 'Deep unsupervised clustering with gaussian mixture variational autoencoders'. 2016.

Ellis, D. P. W., 'Plp and rasta (and mfcc, and inversion) in matlab'. URL http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/.

Eskidere, O., 'Source microphone identification from speech recordings based on a gaussian mixture model', Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 22, No 3, 2014, pp. 754–767.

Eskidere, O. and Karatutlu, A., 'Source microphone identification using multitaper mfcc features', In '2015 9th International Conference on Electrical and Electronics Engineering (ELECO)', pp. 227–231. .

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., 'A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise', In 'Proceedings of the Second International Conference on Knowledge Discovery and Data Mining', KDD'96. AAAI Press, pp. 226–231. URL http://dl.acm.org/citation.cfm?id=3001460.3001507.

Everitt, B. S., Landau, S. and Leese, M., 'Cluster analysis', Wiley Publishing, 4th edn., 2009. ISBN 0340761199, 9780340761199.

Frey, B. J. and Dueck, D., 'Clustering by passing messages between data points', Science, Vol. 315, No 5814, 2007, pp. 972–976. ISSN 0036-8075. . URL https://science.sciencemag.org/content/315/5814/972.

Garcia-Romero, D. and Espy-Wilson, C. Y., 'Automatic acquisition device identification from speech recordings', In '2010 IEEE International Conference on Acoustics, Speech and Signal Processing', ISSN 1520-6149, pp. 1806–1809. .

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L. and Zue, V., 'Timit acoustic-phonetic continuous speech corpus'. 1993.

Gaubitch, N. D., Brookes, M. and Naylor, P. A., 'Blind channel magnitude response estimation in speech using spectrum classification', IEEE Transactions on Audio, Speech, and Language Processing, Vol. 21, No 10, Oct 2013, pp. 2162–2171. ISSN 1558-7916. .

Gaubitch, N. D., Brookes, M., Naylor, P. A. and Sharma, D., 'Single-microphone blind channel identification in speech using spectrum classification', In '2011 19th European Signal Processing Conference', ISSN 2076-1465, pp. 1748–1751.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B., 'Bayesian data analysis', Chapman and Hall/CRC, 2nd ed. edn., 2004.

Hanilci, C., Ertas, F., Ertas, T. and Eskidere, O., 'Recognition of brand and models of cell-phones from recorded speech signals', IEEE Transactions on Information Forensics and Security, Vol. 7, No 2, April 2012, pp. 625–634. ISSN 1556-6013. .

Hanilçi, C. and Kinnunen, T., 'Source cell-phone recognition from recorded speech using non-speech segments', Digital Signal Processing, Vol. 35, 2014, pp. 75–85. ISSN 1051-2004. . URL http://www.sciencedirect.com/science/article/pii/S1051200414002565.

Hastie, T., Tibshirani, R. and Friedman, J., 'The elements of statistical learning: data mining, inference and prediction', Springer, 2 edn., 2009. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

Hermansky, H. and Morgan, N., 'Rasta processing of speech', IEEE Transactions on Speech and Audio Processing, Vol. 2, No 4, Oct 1994, pp. 578–589. ISSN 1063-6676. .

Hyun, D.-K., C.-H., C. and Lee, H.-K., 'Camcorder identification for heavily compressed low resolution videos', 2012.

Iuliani, M., Fontani, M., Shullani, D. and Piva, A., 'A hybrid approach to video source identification', CoRR, Vol. abs/1705.01854, 2017.

Jahanirad, M., Wahab, A. W. A., Anuar, N. B., Idris, M. Y. I. and Ayub, M. N., 'Blind source mobile device identification based on recorded call', Engineering Applications of Artificial Intelligence, Vol. 36, 2014, pp. 320 – 331. ISSN 0952-1976. . URL http://www.sciencedirect.com/science/article/pii/S0952197614002073.

Jain, A. K. and Dubes, R. C., 'Algorithms for clustering data', Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

Johnson, W. B. and Lindenstrauss, J., 'Extensions of lipschitz mappings into a hilbert space', Contemporary Math., Vol. 26, 1984, pp. 189–206.

Junklewitz, H. and Beslay, L., 'Workshop on Clustering and Unsupervised Classification in Forensics', Tech. rep., Joint Research Centre, 2018.

Kingma, D. and Welling, M., 'Auto-encoding variational bayes', .

Kot, A. C. and Cao, H. Image and Video Source Class Identification, Springer New York, New York, NY. ISBN 978-1-4614-0757-7, 2013. pp. 157–178. . URL https://doi.org/10.1007/978-1-4614-0757-7_5.

Kotropoulos, C., 'Telephone handset identification using sparse representations of spectral feature sketches', In '2013 International Workshop on Biometrics and Forensics (IWBF)', pp. 1–4. .

Kotropoulos, C., 'Source phone identification using sketches of features', IET Biometrics, Vol. 3, No 2, June 2014, pp. 75–83. ISSN 2047-4938. .

Kotropoulos, C. and Samaras, S., 'Mobile phone identification using recorded speech signals', In '2014 19th International Conference on Digital Signal Processing', ISSN 1546-1874, pp. 586–591. .

Kraetzer, C., Oermann, A., Dittmann, J. and Lang, A., 'Digital audio forensics: A first practical evaluation on microphone and environment classification', In 'Proceedings of the 9th Workshop on Multimedia & Security', MM&Sec '07. ACM, New York, NY, USA. ISBN 978-1-59593-857-2, pp. 63–74. . URL http://doi.acm.org/10.1145/1288869.1288879.

Kraetzer, C., Qian, K., Schott, M. and Dittmann, J., 'A context model for microphone forensics and its application in evaluations', Vol. 7880. pp. 7880 – 7880 – 15. . URL https://doi.org/10.1117/12.871929.

Kraetzer, C., Schott, M. and Dittmann, J., 'Unweighted fusion in microphone forensics using a decision tree and linear logistic regression models', In 'Proceedings of the 11th ACM Workshop on Multimedia and Security', MM&Sec '09. ACM, New York, NY, USA. ISBN 978-1-60558-492-8, pp. 49–56. . URL http://doi.acm.org/10.1145/1597817.1597827.

Li, C. T., 'Source camera identification using enhanced sensor pattern noise', IEEE Transactions on Information Forensics and Security, Vol. 5, No 2, June 2010, pp. 280–287. ISSN 1556-6013. .

Li, Y., Zhang, X., Li, X., Zhang, Y., Yang, J. and He, Q., 'Mobile phone clustering from speech recordings using deep representation and spectral clustering', IEEE Transactions on Information Forensics and Security, Vol. 13, No 4, April 2018, pp. 965–977. ISSN 1556-6013.

Lloyd, S., 'Least squares quantization in pcm', IEEE Transactions on Information Theory, Vol. 28, No 2, March 1982, pp. 129–137. ISSN 1557-9654. .

Lukas, J., Fridrich, J. and Goljan, M., 'Digital camera identification from sensor pattern noise', IEEE Transactions on Information Forensics and Security, Vol. 1, No 2, June 2006, pp. 205–214. ISSN 1556-6013.

Luo, D., Korus, P. and Huang, J., 'Band energy difference for source attribution in audio forensics', IEEE Transactions on Information Forensics and Security, Vol. 13, No 9, Sep. 2018, pp. 2179–2189. ISSN 1556-6013.

Marra, F., Poggi, G., Sansone, C. and Verdoliva, L., 'Blind prnu-based image clustering for source identification', IEEE Transactions on Information Forensics and Security, Vol. 12, No 9, Sept 2017a, pp. 2197–2211. ISSN 1556-6013. .

Marra, F., Poggi, G., Sansone, C. and Verdoliva, L., 'Blind prnu-based image clustering for source identification', IEEE Transactions on Information Forensics and Security, Vol. 12, No 9, Sept 2017b, pp. 2197–2211. ISSN 1556-6013.

Milani, S., Cuccovillo, L., Tagliasacchi, M., Tubaro, S. and Aichroth, P., 'Video camera identification using audio-visual features', In '2014 5th European Workshop on Visual Information Processing (EUVIP)', pp. 1–6. .

Milani, S., Fontani, M., Bestagini, P., Barni, M., Piva, A., Tagliasacchi, M. and Tubaro, S., 'An overview on video forensics', APSIPA Transactions on Signal and Information Processing, Vol. 1, 2012, p. e2. .

Ng, A. Y., Jordan, M. I. and Weiss, Y., 'On spectral clustering: Analysis and an algorithm', In 'Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic', NIPS'01. MIT Press, Cambridge, MA, USA, pp. 849–856. URL http://dl.acm.org/citation.cfm?id=2980539.2980649.

Panagakis, Y. and Kotropoulos, C., 'Automatic telephone handset identification by sparse representation of random spectral features', In 'Proceedings of the on Multimedia and Security', MM&Sec '12. ACM, New York, NY, USA. ISBN 978-1-4503-1417-6, pp. 91–96. . URL http://doi.acm.org/10.1145/2361407.2361422.

Panagakis, Y. and Kotropoulos, C., 'Telephone handset identification by feature selection and sparse representations', In '2012 IEEE International Workshop on Information Forensics and Security (WIFS)', ISSN 2157-4766, pp. 73–78. .

Pandey, V., Verma, V. K. and Khanna, N., 'Cell-phone identification from audio recordings using psd of speech-free regions', In 'Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on', pp. 1–6. .

Phan, Q., Boato, G. and De Natale, F. G. B., 'Accurate and scalable image clustering based on sparse representation of camera fingerprint', IEEE Transactions on Information Forensics and Security, December 2018, pp. 1–1. ISSN 1556-6013.

Piva, A., 'An overview on image forensics', ISRN Signal Processing, Vol. 2013, No 3, 2013, p. 22. .

Reynolds, D. A. and Rose, R. C., 'Robust text-independent speaker identification using gaussian mixture speaker models', IEEE Transactions on Speech and Audio Processing, Vol. 3, No 1, Jan 1995, pp. 72–83. ISSN 1063-6676. .

Sanderson, C. and Lovell, B., 'Multi-region probabilistic histograms for robust and scalable identity inference', Journal of the Royal Statistics Society, Vol. 5558, 2009, pp. 199–208.

Sanderson, C. and Paliwal, K. K., 'Identity verification using speech and face information', Digital Signal Processing, Vol. 14, No 5, 2004, pp. 449–480. ISSN 1051-2004. . URL http://www.sciencedirect.com/science/article/pii/S1051200404000363.

Satta, R. and Beslay, L., 'Camera fingerprinting based on Sensor Pattern Noise as a tool for combatting Child Abuse on-line', Tech. rep., Joint Research Centre, 2014.

Shullani, D., Fontani, M., Iuliani, M., Al Shaya, O. and Piva, A., 'Vision: a video and image dataset for source identification', EURASIP Journal on Information Security, Vol. 2017, No 15, 2017. .

Sigurosson, S., Petersen, K. B. and Lehn-Schioler, T., 'Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music', In 'Proc. of 7th International Conference on Music Information Retrieval', Victoria, Canada.

Taspinar, S., Mohanty, M. and Memon, N., 'Source camera attribution using stabilized video', In '2016 IEEE International Workshop on Information Forensics and Security (WIFS)', pp. 1–6. .

Tibshirani, R., Walther, G. and Hastie, T., 'Estimating the number of clusters in a dataset via the gap statistic', Vol. 63, 2000, pp. 411–423.

Valsesia, D., Coluccia, G., Bianchi, T. and Magli, E., 'Compressed fingerprint matching and camera identification via random projections', IEEE Transactions on Information Forensics and Security, Vol. 10, No 7, July 2015a, pp. 1472–1485. ISSN 1556-6013. .

Valsesia, D., Coluccia, G., Bianchi, T. and Magli, E., 'Large-scale image retrieval based on compressed camera identification', IEEE Transactions on Multimedia, Vol. 17, No 9, Sept 2015b, pp. 1439–1449. ISSN 1520-9210. .

van der Maaten, L. and Hinton, G., 'Visualizing data using t-sne'. 2008.

van Houten, W. and Geradts, Z., 'Source video camera identification for multiply compressed videos originating from youtube', Digital Investigation, Vol. 6, No 1, 2009, pp. 48–60. ISSN 1742-2876. . URL http://www.sciencedirect.com/science/article/pii/S1742287609000310.

Xu, R. and Wunsch, D., 'Clustering', Wiley-IEEE Press, 2009. ISBN 9780470276808.

Zou, L., He, Q. and Feng, X., 'Cell phone verification from speech recordings using sparse representation', In '2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', ISSN 1520-6149, pp. 1787–1791. .

Zou, L., He, Q. and Wu, J., 'Source cell phone verification from speech recordings using sparse representation', Digital Signal Processing, Vol. 62, 2017, pp. 125–136. ISSN 1051-2004. . URL http://www.sciencedirect.com/science/article/pii/S1051200416301865.

Zou, L., Yang, J. and Huang, T., 'Automatic cell phone recognition from speech recordings', In '2014 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)', pp. 621–625. .

List of abbreviations and definitions

List of figures

List of tables

Annexes

Annex 1. Title of annex

GETTING IN TOUCH WITH THE EU

In person

All over the European Union there are hundreds of Europe Direct information centres. You can find the address of the centre nearest you at: https://europa.eu/european-union/contact_en

On the phone or by email

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

- by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

- at the following standard number: +32 22999696, or

- by electronic mail via: https://europa.eu/european-union/contact_en

FINDING INFORMATION ABOUT THE EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website at: https://europa.eu/european-union/index_en

EU publications

You can download or order free and priced EU publications from EU Bookshop at: https://publications.europa.eu/en/publications. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see https://europa.eu/european-union/contact_en).

KJ-NA-30419-EN-N

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub

ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub

Publications Office of the European Union

doi:10.2760/308387

ISBN 978-92-76-23872-0