A Flexible EM-Like Clustering Algorithm for Noisy Data

Violeta Roizman, Matthieu Jonckheere, and Frédéric Pascal, Senior Member, IEEE

Abstract—Though very popular, it is well known that the Expectation-Maximization (EM) algorithm for the Gaussian mixture model performs poorly for non-Gaussian distributions or in the presence of outliers or noise. In this paper, we propose a Flexible EM-like Clustering Algorithm (FEMCA): a new clustering algorithm following an EM procedure is designed. It is based on both estimations of cluster centers and covariances. In addition, using a semi-parametric paradigm, the method estimates an unknown scale parameter per data point. This allows the algorithm to accommodate heavier tail distributions, noise, and outliers without significantly losing efficiency in various classical scenarios. We first present a very general underlying model for independent, but not necessarily identically distributed, samples of elliptical distributions. We then derive and analyze the proposed algorithm in this context, showing in particular important distribution-free properties of the underlying data distributions. The algorithm convergence and accuracy properties are analyzed by considering the first synthetic data. Finally, we show that FEMCA outperforms other classical unsupervised methods of the literature, such as k-means, EM for Gaussian mixture models, and its recent modifications or spectral clustering when applied to real data sets as MNIST, NORB, and 20newsgroups.

Index Terms—Clustering, high-dimensional data, mixture models, robust estimation, semi-parametric model.

I. INTRODUCTION

The clustering task consists of arranging a set of elements into groups with homogeneous properties/features that capture some important structure of the whole set. As with other unsupervised learning tasks, clustering has become of great interest due to the considerable increase in the amount of unlabeled data in recent years. As the characteristics of real-life data—in geometrical and statistical terms—are very diverse, an intensive research effort has been dedicated to defining various clustering algorithms that adapt to some particular features and structural properties, e.g., [1], [2], [3], [4], [5]. We refer to [6], and the clustering review by [7] for discussions on the different methods and how to choose one, depending on the settings. Among the different types of clustering algorithms, the Expectation-Maximization (EM) procedure to estimate the parameters of an underlying Gaussian Mixture Model (GMM) (see, for instance, the review work by [8]) is a very popular method as its model-based nature typically allows other algorithms to be outperformed when the data is low dimensional, and the clusters have elliptical shapes. This model represents the data distribution as a random variable given by a mixture of Gaussian distributions. The corresponding clustering criterion is simple: all points drawn from a given normal distribution are considered to belong to the same cluster. The Expectation-Maximization algorithm (EM) [9] is a general statistical method used to estimate the parameters of a probabilistic model based on the likelihood maximisation. It is an iterative algorithm with two main steps: the expectation part and the maximization part. In particular, for the GMM case, closed-form expressions exist to obtain parameter estimations at the maximization step. However, its performance decreases significantly in various scenarios of interest for machine learning applications:

- When the data distribution has heavier (or lighter) tails than the Gaussian one and/or in the presence of outliers or noise (see, for instance, [10]). This phenomenon can be explained by the non-robustness of the estimators computed by the algorithm: means and sample covariance matrices [11].
- The presence of different scales in the data might complicate the global ordering of the observations around their closest centers (e.g., through Mahalanobis distances). The usual normalization procedure for the estimation of covariance matrices might be too rigid to get satisfactory clustering results in the presence of significant variability intra and inter-clusters [12].
- When the dimension increases (even in the Gaussian case), the covariance matrix estimation is crucially affected by the high dimensionality, as shown by [13]. Some solutions in that direction include regularization and parsimonious models that restrict the shape of the covariance matrix to decrease the number of parameters to be estimated [14].

To improve the performance of the GMM-EM clustering algorithm in the context of noisy and diverse data, different strategies were contemplated. One consists of modifying the model to take into account the noise, and the other one is to keep the original model and replace the estimators with others that are able to deal with outliers [15]. Several variations of the Gaussian mixture model have been developed in that research
line. In particular, some variations target the problem of mixtures of more general distributions, which allows us to model a broader range of data and possibly allows for the presence of noise and outliers. Regarding using non-Gaussian distributions, [16] proposed an important model defined as a mixture of multivariate $t$-distributions. In this work, the authors suggested an algorithm ($t$-EM or EMMIX in the literature) to estimate the parameters of the mixture with known and unknown degrees of freedom by maximizing the likelihood and addressing the clustering task. More recently, [17], [18], [19] considered hyperbolic and skew $t$-distributions.

Other robust clustering approaches worth mentioning are models that add an extra term to the usual Gaussian likelihood and algorithms with modifications inspired by usual robust techniques such as robust point estimators, robust scales, weights for observations, and trimming techniques. For instance, [20] considered the presence of a uniform noise as background while [21] proposed RIMLE, a pseudo-likelihood-based algorithm that filters the low-density areas. [22] replaced the usual mean and sample covariance with the spatial median and the rank covariance matrix (RCM). [23] introduced a robust scale to define a k-means-like algorithm that can deal with outliers. Furthermore, [24] proposes a robust mixture of distributions estimation based on robust functionals. Moreover, in the work of [25], [26], and [27], different weights for the observations were proposed where small weights correspond, as usual in the robust literature, to observations that are far from the cluster centers. Finally, trimming algorithms such as TCLUST [12] leave out a proportion of data points that are far from all the means in order to estimate the parameters in the M-step better.

Moreover, approaches have been proposed outside the EM framework and steaming from boosting methods in supervised learning algorithms. In [28], some weights are iteratively attached to each point to improve the clustering performance. Though following a completely different approach here, we also aim at defining an algorithm with greater flexibility brought by an extra parameter per data point. This flexibility allows both to outperform more traditional algorithms and to be adaptive to a large class of underlying distributions. Different from boosting-based methods, however, it indirectly follows the path of robust statistical approaches.

The proposed method is inspired by the robust applications of the Elliptical Symmetric (ES) distributions [29], [30]. Of course, elliptical distributions have been widely used in many applications where Gaussian distributions were not able to approximate the underlying data distribution because of the presence of heavy tails or outliers [31], [32]. This general family includes, among others, the class of compound-Gaussian distributions that contain Gaussian, $t$- and $k$- distributions [33], [34], [35] as well as the class of Multivariate Generalized Gaussian Distributions [36].

In this paper, we present

- A generic model with independent (but not identically distributed in general) data involving one scale parameter per data point.
- A clustering algorithm with the following characteristics: 1/ it follows the two steps expectation and maximization of EM algorithms; 2/ at the E-step, the expected conditional log-likelihood leads to estimators independent of the shapes of the distributions; 3/ at the M-step, it derives cluster center and covariance matrix estimations, which turn to follow classical robust estimators.

We show that, under mild assumptions, the membership probability estimates do not depend on the shape of data distributions, making the algorithm generic, simple, and robust. It can be noticed that the indirect estimations of the scale/nuisance parameters could also be used for classification and outlier detection purposes by discriminating data and helping data assignment [37], but this is not our scope here. A key feature of the proposed algorithm is to be self-contained in the sense that no extra-parameters need to be tuned as is the case for approaches mentioned above (e.g., penalty parameters, rejection thresholds, and other distribution parameters such as shapes or the degrees of freedom).

In the sequel, we include practical and theoretical studies providing evidence of the algorithm’s performance. In particular, we theoretically justify the efficiency of our algorithm using various arguments:

1) When the underlying model belongs to the class of elliptical distributions, with different means and dispersion matrix per cluster but with cluster-independent density generators (even different ones within clusters), then the estimation of membership probabilities does not depend on each specific density function. This is a consequence of the fact that these probability estimations do not depend on the scale factors of the covariance matrix but only on the scatter/Dispersion matrices. Hence, the algorithm makes no mismatch error when the density generator is unknown whenever this assumption is fulfilled. This is shown in Proposition 4.

2) Even when the density function is different for every cluster, there are regimes where the mismatch error can be controlled. We give an example using $t$-distributions with various degrees of freedom. See Proposition 5.

From a practical perspective, the induced clustering performance is largely improved compared to k-means, the EM algorithm for GMM and HDBSCAN [38], [39], [40] when applied to real data sets such as MNIST variations [41], NORB [42] and 20newsgroups [43]. In agreement with the proposed results, previous works on the classification of the MNIST dataset suggest the non-Gaussianity of the clusters [44]. Compared to spectral clustering and $t$-EM, TCLUST, and RIMLE, our algorithm performs similarly in classic cases and much better in others. Furthermore, the proposed algorithm is able to provide accurate estimations of location and dispersion parameters even in the presence of heavy-tailed distributions or additive noise, as proved in simulations where our algorithm beats the other compared models.

The rest of the paper is organized as follows. In Section II, after introducing the models of interest in detail, we present the clustering algorithm and discuss some of its important aspects, notably by proving convergence results on the estimation of the parameters. Section III is devoted to the experimental results,
which allow us to show the improved performance of the proposed method for different synthetic and real data sets compared to other commonly used methods. Finally, conclusions and perspectives are stated in Section IV.

II. MODEL AND THEORETICAL JUSTIFICATIONS

This Section presents the underlying theoretical model and the proposed clustering algorithm. Let \{\mathbf{x}_i\}_{i=1}^n\ be a set of \(n\) independent data points belonging to \(K\) different classes \(C_1, \ldots, C_K\) of cardinality \(n_k\) such that if \(\mathbf{x}_i\) belongs to class \(C_k\), then its distribution is given by a density \(f_{i, \theta_k}\) depending on the one hand on a nuisance parameter \(\tau_{ik}\) and on the other hand on class parameters \(\theta_k\). Remark that because of this nuisance parameter, the points are not i.i.d but only independent. As usual in clustering, we complete the model by latent variables that are not observed \((Z_{ik})_{1\leq i\leq n}\), representing the cluster label of each observation \(\mathbf{x}_i\). They follow a multinomial distribution with parameters \(\pi_k\). We denote by \(\theta_k = (\theta_k, \pi_k)\) the extended clustering parameters.

Remark 1. Let us underline the level of generality of the model. Within a particular cluster, data points share identical mean and scatter matrices. However, data points are supposed to be independent but not i.i.d. Indeed, the \(K\) clusters are only characterized by parameters \(\theta_k\), while the shape of the distributions can change from one observation to another. In the next paragraph, we explain the relevance of such a general structure, where we fix a set of distributions for the data.

We consider a very large class of distributions in order to generalize the classical Gaussian mixture model significantly: the Elliptical Symmetric (ES) distributions. We shall suppose that the density function associated with a point in class \(k\) follows an ES distribution with mean \(\mu_k\) and covariance matrix \(\Sigma_k\) and can be written as

\[
\begin{align*}
  f_{i, \theta_k}(\mathbf{x}_i) &= A_{ik} |\Sigma_k|^{-1/2} \tau_{ik}^{-m/2} g_{i,k} \\
  &\times \left( (\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k) \right)^{-(m+1)/2},
\end{align*}
\]

where \(A_{ik}\) is a normalization constant, \(g_{i,k} : [0, \infty) \rightarrow [0, \infty)\) is any function (called the density generator) such that (1) defines a pdf. The matrix \(\Sigma_k\) reflects the structure of the covariance matrix of \(\mathbf{x}_i\). Note that the covariance matrix is equal to \(\Sigma_k\) up to a scale factor if the distribution has a finite second-order moment (see [30] for details). This is denoted \(ES(\mu_k, \tau_{ik} \Sigma_k, g_{i,k}())\). Note that the clustering parameter is \(\theta_k = (\pi_k, \mu_k, \Sigma_k)\) while the nuisance parameter is \(\tau_{ik}\).

At this stage, some comments are in order:

1) When \(g_{i,k} = g\), \(\forall i, k\), then all the points follow the same ES distribution, but each class has a different mean and covariance matrix. However, note that this model is much more general than a Gaussian mixture model as the class of ES distributions is much wider and includes, in particular, lighter and heavier tails than Gaussian ones. Also, the presence of the nuisance parameters makes the model non-i.i.d. in general. An important aspect of our results is that our algorithm is, in that case, insensitive to the function \(g\), and hence allows us to treat efficiently real data sets even when the function \(g\) is unknown.

2) When \(g_{i,k} = g_i, \forall i, k\), we obtain a more general model than the previous one, where, even if the distribution of the points does not depend on the classes, except for their mean and covariances, the data within a class might follow e.g., a mixture of ES distributions (with fixed mean and covariance matrix). It has practical importance since many data sets are compiled from different data sources with different characteristics. In that case, again, our algorithm is still insensitive to the functions \(g_i\), giving a lot of modelling flexibility and mismatch robustness.

3) When \(g_{i,k} = g_k\), \(\forall i, k\), then we are considering one different ES distribution per class of data. For this setting, the clustering results do depend on \(g_k\), which can be a practical obstacle to getting sound results. However, we show that our method can alleviate this dependence, leading to good performance in some regimes.

Elliptical distributions have been used in many applications where one has to deal with the presence of heavy tails or outliers [31], [32]. This general family includes Gaussian, \(t\)− and \(k\)− distributions, among others [33], [34], [35]. This model admits a Stochastic Representation Theorem. A vector \(\mathbf{x}_i \sim ES(\mu_k, \tau_{ik} \Sigma_k, g_{i,k}())\) if and only if it admits the following stochastic representation [45]

\[
\mathbf{x}_i \sim \mathcal{N}_d(\mu_k + \sqrt{Q_{ik}} \sqrt{\tau_{ik}} \mathbf{A}_k \mathbf{u}_i, \Sigma_k),
\]

where the non-negative real random variable \(Q_{ik}\), called the modular variate, is independent of the random vector \(\mathbf{u}_i\) that is uniformly distributed on the unit \(m\)-sphere and \(\mathbf{A}_k \mathbf{A}_k^T\) is a factorization of \(\Sigma_k\) while \(\tau_{ik}\) is a deterministic but unknown nuisance parameter.

Note that, in this work, one considers that \(C_{ik} = \tau_{ik} \Sigma_k\) can also depend on the \(i\)th observation, through the nuisance parameter. We assume that the distributions at hand have a second-order moment and that \(C_{ik}\) is the covariance matrix for identifiability purposes. This assumption implies the particular normalization on \(Q_{ik}\), that is

\[
E[Q_{ik}] = \text{rank}(C_{ik}) (= \text{rank}(\Sigma_k)) = m,
\]

when \(\Sigma_k\) is full rank, following for instance [46]. In the sequel, we hence call \(C_{ik}\) the covariance matrix and \(\Sigma_k\) the scatter matrix.

Finally, an ambiguity remains in the scatter matrix \(\Sigma_k\). Indeed, for any positive real number \(c\), \(C_{ik}/c, c \Sigma_k\) lead to the same covariance matrix \(C_{ik}\). In this work, we choose to fix the trace of \(\Sigma_k\) to \(m\), meaning we formally consider shape matrices \(\Sigma_k = \frac{m}{\text{tr}(\Sigma_k)} \Sigma_k\) as cluster features. Since there is no impact on the derivations (see Remark 3), we keep the notation \(\Sigma_k\) for simplicity purposes. Other normalizations could have been chosen instead, for instance, imposing a unit-determinant for \(\Sigma_k\) without affecting the clustering results.

[46] showed in the complex case that, given random sample from \(\mathbf{x}_i \sim CES(0_m, \tau_{ik} \Sigma_k, g_{i,k}())\), the estimation of \(\tau_{ik}\) using Maximum Likelihood Estimation (MLE) is decoupled.
from the estimation of $\Sigma_k$. Furthermore, the authors proved that the maximum likelihood estimator for $\Sigma_k$ is Tyler’s estimator, regardless of the functions $g_k$. This is a remarkable result, underlying the universal character of the Tyler estimator in this class of distributions. We will build on this distribution-free property of Tyler’s estimator, which is central to our results.

A. The M-Step: Parameter Estimation for the Mixture Model

Similarly to the EM for GMM, we compute the label for each observation and cluster in the E-step, while in the M-step, we estimate the parameters of interest $\theta = (\theta_k)_{k=1}^K$.

We start by expressing the expected conditional log-likelihood of the model. Let $x = (x_1, \ldots, x_n)$ be a sample, $Z = (Z_1, \ldots, Z_n)$ be the corresponding latent variables, and $\theta$ be a set of parameters.

$$E[Z|x, \theta] [\ell(Z, x; \theta)] = E[Z|x, \theta] \left[ \log \left( \prod_{i=1}^n \prod_{k=1}^K \pi_k \prod_{i=1}^n \pi_k \right) \right]$$

$$= E[Z|x, \theta] \left[ \log \left( \prod_{i=1}^n \prod_{k=1}^K \left( \pi_k \prod_{i=1}^n \pi_k \right) \right) \right]$$

$$= \sum_{i=1}^n \sum_{k=1}^K \sum_{l=1}^K \pi_{ik} \pi_{lk} \left( \sum_{i=1}^n \pi_{ik} \right) \log \left( \pi_k f_i \pi_k \left( x_i \right) \right)$$

Then, replacing the density function, we get

$$E[Z|x, \theta] [\ell(Z, x; \theta)] = \sum_{i=1}^n \sum_{k=1}^K \pi_{ik} \left[ \log(\pi_k) + \log(\pi_{ik}) \right]$$

$$+ \log \left( \frac{C_{ik}^{-\frac{1}{2}} g_i((x_i - \mu_k)^T C_k^{-1} (x_i - \mu_k))}{\pi_k} \right)$$

where $\pi_{ik} = P_i \theta \left( Z_i = k | x_i = x_i \right)$ with $\sum_{k=1}^K \pi_{ik} = 1$, and $C_{ik} = \tau_{ik} \Sigma_k$. The values $\pi_{ik}$, which conditionally depend on $\theta^*$, are the ones computed in the E-step. On the other hand, the parameters grouped into $\theta^*$ are derived with a maximum likelihood approach in the M-step.

We now include two propositions that summarize the derivation of the estimators for all the model parameters. As underlined previously, a key step using the ideas in [46] consists of factorizing the likelihood into two factors, which further allows the description of fundamental properties of the estimators in the E and M steps. In Proposition 1, we derive the estimator for the $\tau$ parameters. Then, in Proposition 2, we derive the rest of the model parameters.

Proposition 1: Suppose $x_1, \ldots, x_n$ an independent sample with $x_i \sim ES(\mu_k, \tau_{ik} \Sigma_k, g_k(\cdot))$ for some $k \in \{1, \ldots, K\}$. Suppose $\int t^{m/2} g_k(t) dt \leq \infty, \forall i, k$. Then, the derivation of the maximum likelihood estimation of the $\tau_{ik}$ parameters is decoupled from one of the rest of the estimators. For fixed parameters $\Sigma_k$ and $\mu_k$, the $\tau_{ik}$’s estimators are computed as

$$\hat{\tau}_{ik} = \frac{\left( x_i - \mu_k \right)^T \Sigma_{ik}^{-1} (x_i - \mu_k)}{a_{ik}} , \forall i \leq n, \forall k \leq K$$

$$p(\tau) = \left[ \sum_{i=1}^n \sum_{k=1}^K \pi_{ik} f_i \pi_k \left( x_i \right) \right]^{\frac{2}{\tau}}$$

$$= C/\tau$$

As in many other examples on unbounded domains, this is an improper prior as it does not correspond to a proper distribution on $\mathbb{R}^+$. We now describe the ML estimators for $\theta$.

Proposition 2: Given an independent random sample $x_1, \ldots, x_n$, the latent variables $Z_i$, and expected conditional log-likelihood of the model stated before, the maximization w.r.t. $\theta_k$ for $k = 1, \ldots, K$, leads to the following equations that the estimators have to fulfil. The closed equations are

$$\hat{\pi}_{ik} = \frac{1}{n} \sum_{i=1}^n p_{ik}$$

for the proportion of each distribution,

$$\hat{\mu}_k = \frac{1}{n} \sum_{i=1}^n p_{ik} x_i$$

$$\hat{\Sigma}_k = \frac{1}{n} \sum_{i=1}^n w_{ik} (x_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x_i - \hat{\mu}_k)$$

where $w_{ik} = p_{ik} / \sum_{i=1}^n p_{ik}$, for the scatter matrices.

**Proof:** See Appendix B.

It follows from the derivation of Proposition 2 that there is a system of two fixed-point equations that hold for the estimators of $\mu_k$ and $\Sigma_k$. This system is iteratively solved to obtain these coupled estimators as explained in Section II-D.

We can now prove a fundamental property of the algorithm, which is the monotonicity of the likelihood of the model. We later illustrate this property with simulations in Section II-D.

We now prove a fundamental property of the algorithm, which is the monotonicity of the likelihood of the model. We later illustrate this property with simulations in Section II-D.

**Proof:** See Appendix A.

**Remark 2:** Considering the scale parameter $\tau$ as an unknown (nuisance) parameter in our model is equivalent from a Bayesian point of view to considering a Jeffreys prior [47], which corresponds to the unnormalized uniform distribution on the real line. This result can be obtained using simple computations leveraging the shape of the ES densities. Indeed, by definition, Jeffreys prior is proportional to the square root of the determinant of the Fisher information matrix (scalar in that case):
and assuming the convergence of the iterations associated with the fixed-point equations system derived in Proposition 2, the steps defined by the estimator updates from Propositions 1 and 2 lead to a succession \( \{ \theta_t \}_{t=1}^N \) with an increasing likelihood.

**Proof:** See Appendix C.

It is important to notice that the derivation of estimators in our model results in usual robust estimators for the mean and covariance matrices. Specifically, both can be assimilated to \( M \)-estimators with a specific \( u \) function [11]. Actually, both the expressions for the mean and the scatter matrix estimators are very close to the corresponding Tyler’s \( M \)-estimator (see for more details [49], [50]). Main differences arise from the mixture model, leading to different weights involved by the different distributions. However, in the case of clusters with equal probability, i.e., \( p_{ik} = 1/K \) for \( k = 1, \ldots, K \) and \( i = 1, \ldots, n \), one retrieves exactly the Tyler’s \( M \)-estimator for the scatter matrix while the mean estimator differs only from the square-root at the denominator (see the explanation later on). Although our estimators are derived as usual MLE (but for parametrized (thanks to the \( \tau_{ik} \)’s elliptical distributions, they are intrinsically robust. Indeed, as detailed in [51], Tyler’s and Maronna’s \( M \)-estimators can either be obtained through MLE approaches for particular models (e.g., Student-\( t \) \( M \)-estimators) or directly from other cost functions (e.g., Huber \( M \)-estimators) and all those estimators are by definition robust.

Thus, this approach can be seen as a generalization of Tyler’s \( M \)-estimators to the mixture case. Indeed, one has for the \( \mu_k \)

\[
\frac{1}{n} \sum_{i=1}^{n} u_1 \left( (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right) (x_i - \mu_k) = 0,
\]

with \( u_1(t) = p_{ik}/t \), while \( \hat{\Sigma}_k \) can be written as

\[
\hat{\Sigma}_k = \frac{1}{n} \sum_{i=1}^{n} u_2 \left( (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right) \\
\times (x_i - \mu_k)^T (x_i - \mu_k)
\]

with \( u_2(t) = m \frac{w_{ik}}{t} \) and \( w_{ik} = \frac{n p_{ik}}{\sum_{l=1}^{K} p_{lk}} \).

This similarity to classical Tyler’s estimators explains the robust character of our proposal. Indeed, the difference lies in the weights terms \( p_{ik} \) and \( w_{ik} \) appearing in the \( u_j(.) \) functions traditionally introduced in the robust statistics literature. These naturally implies that those \( u_1(.) \) and \( u_2(.) \) functions continue to respect Tyler’s conditions (although [49] used \( u_1(t^{1/2}) \) instead of \( u_1(t) \), see [51] for more details).

The convergence of the fixed-point equations defining the \( M \)-estimators has been shown in [11] but under a restrictive assumption on the \( u \) function, which is not fulfilled in our case. On the other hand, Kent proved in [52] that for fixed mean, there is the convergence of the fixed-point equation for the covariance estimator under a normalization constraint. Finally, he also showed that for some \( u \) functions, the joint mean covariance estimations boil down to a constrained covariance estimation. Unfortunately, this trick does not work in our case. Hence, the joint convergence of the fixed-point equations for Tyler’s estimators is still an open problem in statistics, even in the case of one distribution (no mixture).

We later perform analysis and simulations that confirm the algorithm’s robustness in practice. In particular, the setups in Section III-A include distributions with heavy tails, different distributions, and noise.

**B. The E-Step: Computing the Conditional Probabilities**

In contrast to the estimators derived in Proposition 2, (5) shows that the estimation of the \( \tau_{ik} \) parameters are linked to the functions \( g_{i,k} \) that characterizes the corresponding Elliptical Symmetric distribution. We now give a central result for our algorithm. The following proposition shows that the \( p_{ik} \)’s estimators do not depend on density generators when \( g_{i,k} = g_i \).

**Proposition 4:** Given an independent random sample \( x_i \sim ES(\mu_k, \tau_{ik} \Sigma_k, g_i(.) \) for some \( k = 1, \ldots, K \), the resulting estimated conditional probabilities \( \hat{\pi}_{ik} = \hat{\Pi}_{ik}(Z_i = k \mid x_i = x_i) \) have the following expression for all \( i = 1, \ldots, n \) and \( k = 1, \ldots, K \)

\[
\hat{\pi}_{ik} = \frac{\hat{\pi}_k \left( (x_i - \hat{\mu}_k)^T \Sigma_k^{-1} (x_i - \hat{\mu}_k) \right)^{-m/2} |\Sigma_k|^{-1/2}}{\sum_{j=1}^{K} \hat{\pi}_j \left( (x_i - \hat{\mu}_j)^T \Sigma_j^{-1} (x_i - \hat{\mu}_j) \right)^{-m/2} |\Sigma_j|^{-1/2}},
\]

where \( \hat{\pi}_k, \hat{\mu}_k \) and \( \hat{\Sigma}_k \) are given in Proposition 2.

**Proof:** See Appendix D.

**Remark 3:**

- The consequences of Proposition 4 (combined with Proposition 2) are of utmost importance since it allows us to derive the conditional probabilities required in the E-step independently of the distributions \( g_i(.) \) and the \( \tau_{ik} \)’s parameters. In other words, for any independent ES-distributed observation \( x_i \) with mean \( \mu_k \) and covariance matrix \( \tau_{ik} \Sigma_k \), a unique EM algorithm is derived that does not depend on the shapes of the various involved distributions. This is essential because the absence of precise knowledge on the specific data distribution is the most common situation in real-life applications while estimating it might significantly degrade the performance.

- Second, it evidences that the normalization of the \( \Sigma_k \) estimator does not affect the probability computation in the E-step. In other words, the normalization of the scatter matrices is not relevant to the clustering results. On the other hand, the normalization of \( \Sigma_k \) only affects the scale of the \( \tau_{ik} \) parameters.

- The particular case where the data points arise from a mixture of one ES distribution, \( g_i = g, \forall 1 \leq t \leq n \), is contained in Proposition 4. We remark on the particular example included in this case when all the distributions are Gaussian. If \( x_i \sim N(\mu_k, \tau_{ik} \Sigma_k) \) then the corresponding density generator is \( g(t) = e^{-t^2/2} \). The corresponding maximizer is \( \arg \max \left\{ \sum_{i=1}^{m} g(t) \right\} = m \), consequently the estimator is, as derived in (5), as follows:

\[
\hat{\tau}_{ik} = \frac{(x_i - \hat{\mu}_k)^T \Sigma_k^{-1} (x_i - \hat{\mu}_k)}{m}.
\]
The case where \( g_t = g_k \) cannot be directly handled as a particular case of Proposition 4. Indeed, assuming each class is drawn by a common ES distribution \( g_k \) implies in general that extra-parameters, such as, for instance, the degree of freedom \( \nu_k \) for \( t \)-distributions, the shape parameters for the \( K \)-distributions and the generalized Gaussian distributions, depend on \( k \). Those parameters have to be estimated in the M-step. We give an example in the next section in the particular case of a mixture of \( t \)-distributions.

C. Different Density Generator per Class

When the density generator depends on the class, our computations show that the \( p_{ik} \) do depend on the \( g_k \), as opposed to the previous case. When the density generators are known (which is quite unrealistic in practice), this assumption naturally increases the clustering performance since extra \textit{a priori} information is added to the model. On the contrary, it implies a performance loss when the real data distribution is not the assumed one.

To illustrate the type of dependence reached in that case, we derive the E-step for the particular case of a mixture of multivariate \( t \)-distributions with different degrees of freedom \( \nu_k \). That is the case where there are \( K \) different \( g_k \) functions, one for each cluster. The probability density function of each distribution is given by

\[
 f_{i,\theta_k}(x_i) = \frac{\Gamma\left(\frac{\nu_k + m}{2}\right)}{\pi^{\frac{d}{2}} \Gamma\left(\frac{\nu_k}{2}\right)} \left(\nu_k \pi_k \right)^{-m/2} \left[1 + \frac{(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}{\nu_k \nu_k}ight]^{-\left(\nu_k + m\right)/2}.
\]

The following proposition states a quantitative approximation of the estimated conditional probabilities in terms of the “Gaussian value” (i.e., the value obtained for class independent \( g_t \)) when the \( \nu_k \) parameters and \( m \) grow at the same rate.

**Proposition 5:** Given an independent sample of a mixture of \( K \) \( t \)-distributions, with \( x_i \sim t_{\nu_k}, \nu_k \) being the degrees of freedom. If for each \( k \), \( \nu_k \approx c_k \), then

\[
 \hat{p}_{ik} = \frac{\hat{p}_k \nu_k \pi_k}{\sqrt{\nu_k}} \sqrt{\frac{c_k}{1+c_k}} + O\left(\frac{1}{m}\right).
\]

**Proof.** See Appendix E.

This scenario includes, of course, the case where all the \( \nu_k \) are equal (See Remark 3). Additionally, it includes, when the degrees of freedom are large, with fixed dimension \( m \), the Gaussian case as detailed in the proof. Finally, the other intermediate situations where all the \( \nu \) parameters do not differ much from the dimension \( m \) are hence shown to be very close to the Gaussian computation. If neither of these conditions applies, an \textit{ad hoc} estimation of the \( \nu_k \) is possible and must be performed in the M-step.

**D. Implementation Details and Numerical Considerations**

The general structure of the proposed algorithm is the same as the one of the classical EM for GMM. The differences between both algorithms lie in the \( p_{ij} \) expression and the recursive update equations for the parameter estimations. We design slightly different variations of the M-step and study the convergence, precision, and speed. We do this considering that the estimators for \( \mu \) and \( \Sigma \) are weighted versions of the classic estimators. More precisely, based on equations (7) and (8), we propose four alternatives to accelerate the convergence speed. These versions depend on two different aspects. One aspect consists of using the just computed estimation of the mean or the estimator from the previous iteration of the loop. The other facet is proposed to emphasize the weights of the data points in the computation of the estimators based on Tyler’s estimator. Section II mentions that the location and scatter estimators are close to Tyler’s up to the square root of the Mahalanobis distance when the location is unknown. We propose modifying the weights by adding this square root to mimic Tyler’s estimator. The different versions are defined as follows:

1) Version 1: the parameter \( \mu \) used to compute the estimator \( \Sigma \) is the one obtained in the \textit{same iteration} of the fixed-point loop.
2) Version 2: the \( \mu \)-parameter is the one obtained in the \textit{previous iteration}.
3) Version 3: we propose an \textit{accelerated method} where the quadratic forms

\[
(x_i - \hat{\mu}_j)^T \Sigma_k^{-1} (x_i - \hat{\mu}_k)
\]

in the denominators of the fixed-point \( \mu \) equations are replaced by their square root, corresponding to the original Tyler’s \( M \)-estimators.
4) Version 4: we implement the same \textit{acceleration procedure} on top of the algorithm of Version 2.

For concreteness, in Algorithm 1, we describe the complete algorithm in Versions 1 and 4. In the particular case described in Section II-C, where all \( g_{ik} \) functions are known, the \( p_{ik} \) should be computed with the Bayes expression as in (19).

The plots in Fig. 1 show the convergence of the fixed-point equations for the estimation of \( \mu \) and \( \Sigma \) for the different versions of the algorithm in two different setups with two distributions

![Fig. 1. Convergence speed of the fixed-point equations for the estimation of \( \mu \) (left) and \( \Sigma \) (right). The results in the Gaussian case are plotted on the top and the bottom, and the ones for a mixture of \( t \)-distributions are shown on the bottom. Each line represents the median of the values obtained on each iteration of the fixed-point iteration for all the iterations of FEMCAs and all clusters. The grey areas represent the quartile range of each iteration of each version.](image-url)
Algorithm 1: Scheme of FEMCA.

Input: Data \( \{x_i\}_{i=1}^n \), \( K \) the number of clusters
Output: Clustering labels \( Z = \{z_i\}_{i=1}^n \)
1: Set initial random values \( \theta^{(0)} \);
2: \( l \leftarrow 1; \)
3: while not convergence do
4: \( \text{E-step:} \) Compute \( p_{ik}^{(l-1)} = P_{i, \theta^{(l)}}(Z_i = k|x_i = x_i) \) for \( 1 \leq k \leq K \)
5: \( p_{ik}^{(l)} = \pi_k^{(l)} \left( \frac{\Delta_{ik}^{(l-1)} - m/2}{|\Sigma_k^{(l-1)}|^{1/2}} \right) \)
\( \sum_{k'=1}^K \frac{\pi_k^{(l)} \left( \Delta_{ik}^{(l-1)} - m/2 \right)}{|\Sigma_k^{(l-1)}|^{1/2}} \)
\( \Delta_{ik}^{(l-1)} = (x_i - \mu_{ik}^{(l-1)})^T (\Sigma_{k}^{(l-1)})^{-1}(x_i - \mu_{ik}^{(l-1)}) \)
6: For each \( 1 \leq k \leq K; \)
7: Update \( \hat{\pi}_k^{(l)} = \frac{1}{n} \sum_{i=1}^n p_{ik}^{(l)} \) and compute \( w_{ik}^{(l)} = \frac{\pi_k^{(l)}}{\sum_{j=1}^K p_{ij}^{(l)}} \) Set \( \mu_k^{(l)} = \mu_k^{(l-1)} \) and \( \Sigma_k' = \Sigma_k^{(l-1)} \);
8: \( \text{while not convergence do} \)
9: \[ \begin{align*}
\mu_k'' &= \frac{\sum_{i=1}^n p_{ik}^{(l)} x_i}{\sum_{i=1}^n p_{ik}^{(l)}} \left( \sum_{i=1}^n p_{ik}^{(l)} x_i - \mu_{ik}^{(l)} \right)^T (\sum_{i=1}^n p_{ik}^{(l)})^{-1} (x_i - \mu_{ik}^{(l)}) \\
\Sigma_k'' &= m \sum_{i=1}^n p_{ik}^{(l)} w_{ik}^{(l)} (x_i - \mu_{ik}^{(l)})^T (x_i - \mu_{ik}^{(l)}) \\
\Sigma_k' &= m \sum_{i=1}^n p_{ik}^{(l)} w_{ik}^{(l)} (x_i - \mu_{ik}^{(l)})^T (\sum_{i=1}^n p_{ik}^{(l)})^{-1} (x_i - \mu_{ik}^{(l)})
\end{align*} \]
10: or
11: \[ \begin{align*}
\mu_k'' &= \frac{\sum_{i=1}^n p_{ik}^{(l)} x_i}{\sum_{i=1}^n p_{ik}^{(l)}} (x_i - \mu_{ik}^{(l)})^T (\sum_{i=1}^n p_{ik}^{(l)})^{-1} (x_i - \mu_{ik}^{(l)}) \\
\Sigma_k'' &= m \sum_{i=1}^n p_{ik}^{(l)} w_{ik}^{(l)} (x_i - \mu_{ik}^{(l)})^T (x_i - \mu_{ik}^{(l)}) \\
\Sigma_k' &= m \sum_{i=1}^n p_{ik}^{(l)} w_{ik}^{(l)} (x_i - \mu_{ik}^{(l)})^T (\sum_{i=1}^n p_{ik}^{(l)})^{-1} (x_i - \mu_{ik}^{(l)})
\end{align*} \]
12: \( \text{end} \)
13: Update \( \mu_k^{(l)} = \mu_k'' \) and \( \Sigma_k^{(l)} = \Sigma_k' \) (with the trace normalisation \( \text{tr}(\Sigma_k^{(l)}) = m \)) and \( p_{ik}^{(l)} \);
14: \( l \leftarrow l + 1; \)
15: \( \text{end} \)
16: Set \( z_i \) as the index \( k \) that has the maximum \( p_{ik} \) value;

each. We separately study the two parameters to see if the variations differently affect each convergence. The first one is a simple case with two well-separated Gaussian distributions in dimension \( m = 10 \) (means equal to \( 0_m \) and \( 2*1_m \), and covariance matrices are the identity \( I_m \) and a diagonal matrix with elements \( 0.25, 3.5, 0.25, 0.75, 1.5, 0.5, 1, 0.25, 1, 1 \)). The second one is a mixture of two \( t \)-distributions with heavy tails and the same parameter \( (\nu_1 = \nu_2 = 3) \). As one can see in both cases, the convergence is reached for all algorithm versions after approximately twenty iterations of the fixed-point loop. Fig. 1 shows that the convergence speed is improved for Versions 3 and 4, as expected. On the other hand, we study in Fig. 2 the evolution of the log-likelihood in these two different scenarios. In the case of the multivariate \( t \)-distributions, we computed the likelihood with the true degrees of freedom \( (\nu_k = 3) \). This Figure shows an increased likelihood in all cases and a faster convergence of Version 1 of the model because the correct values for the mean/scatter estimators are reached faster even though its computation takes a bit longer than for Versions 3 and 4. The estimation accuracy of Versions 1 and 2 is better by construction (ML-based) than for Versions 3 and 4. Based on these Figures and previous studies about fixed-point fast convergence (see e.g., [53]), Version 1 is kept since it follows the original proposal, and although slightly slower than Version 2 for the fixed-point loop, it is faster for the convergence of the algorithm. Furthermore, we fixed the number of iterations to 20 in all the experiments. Notice that increasing this number does not result in a significant increase in terms of clustering performance.

Let us now discuss initialization and thresholds used in the proposed algorithm. The mean parameters are initialized as the means resulting from the k-means algorithm. When k-means output clusters of only one point, we rerun k-means, leaving out the isolated points. Due to singularity problems, we take the initial scatter matrix as the identity matrix. We set the initial value of all \( \tau \) parameters to one. For the convergence flag, we consider \( 10^{-6} \) for the threshold of the \( \ell_2 \)-norm difference of consecutive estimators, and the maximum number of iterations of the fixed-point loop length is set to 20 based on the previous discussion. We obtain the same final clustering results for each run using the initialisation described above. In the low-dimensional case, we truncate the \( \tau \) value to avoid numerical issues induced by points very close to the mean. That is, if \( \tau \) is smaller than \( 10^{-12} \), we change its value to the selected threshold. The implementation in Python of the algorithm is available at the repository https://github.com/violetr/frem.

Furthermore, it is important to remark that, in our approach, the constraint on the trace of \( \Sigma \) (\( \text{tr}(\Sigma) = m \)) does not act as a regularization procedure, as it is usually the case in EM-like algorithms [12], [21]. As mentioned in Remark 3, the trace constraint does not affect the clustering results.

Finally, regarding the algorithm’s complexity, it happens to be the same as one of the classical EM algorithms for a mixture of Gaussian distributions. The E-step has the same complexity as the usual algorithm. For the M-step, even though a nested loop is included to solve the fixed-point equations, the complexity is not increased since the number of iterations is constant and the main cost of each iteration corresponds to the scatter matrix inversion as in EM for GMM.
### III. Experimental Results

This section presents experiments obtained with both synthetic and real data. Section III-A illustrates how the proposed algorithm can adapt to heavier tail distribution and outliers on simulated data, while Section III-B deals with real data. We study the convergence of the fixed point equations and the estimation error in the case of synthetic data (for which we know the true parameter values). We compare our results to the ones of the classical EM for GMM, EM for multivariate t-distributions, TCLUST [12], and RIMLE [21]. Additionally, for the real data, we compare the clustering results with the ground truth labels for k-means, HDBSCAN, and spectral clustering [54]. The comparison between the former three and our algorithm is straightforward because they all have in common only one main parameter (the number of clusters) that we fix and suppose known in our experiments. Regarding the implementations, we use Scikit-learn [55] for k-means and the Gaussian Mixture and the R package EMMIXskew [56] for the mixture of t-distributions. Concerning TCLUST and the RIMLE algorithms, we set the number of clusters and use the default values for the rest of the parameters. We avoided the artificial constraint on the TCLUST algorithm solution caused by the eigenvalue constraint threshold when possible. We used the OTRIMLE version of RIMLE that selects the main parameter of the model with a data-driven approach [57]. For both of them, we use the R implementation provided by the authors. In the case of spectral clustering, we run the Scikit-learn implementation, where it is necessary to tune an extra parameter in order to build the neighbourhood graph. We set the number of neighbours in the graph equal to the number that maximizes the silhouette score [58]. A fair comparison with HDBSCAN is even more difficult to set because the parameters to tune are completely different and less intuitive than those of the other algorithms. Once again, we select the best silhouette score pair of parameters by sweeping a grid of selected values.

We then quantify the differences in performance by using the usual metrics for the clustering task known as the adjusted mutual information (AMI) index and the adjusted rand (AR) index [59]. One also provides the correct classification rate. One also provides the correct classification rate. Additionally, the πₖ vectors, corresponding to the distribution proportions, are randomly chosen from a set of possibilities that avoid trivial and giant clusters. These cases are avoided due to the ill-posed clustering problem that it implies.

In these experiments, we include all the considered algorithms that estimate parameters. Thus, we exclude the comparison k-means, spectral clustering, and HDBSCAN. Table III shows the estimation error when estimating the main parameters of the model for all the setups. Furthermore, we report the clustering

| Setup | m | n | distribution 1 | distribution 2 | distribution 3 |
|-------|---|---|---------------|---------------|---------------|
| 1     | 8 | 1000 | t, df = 3 | t, df = 3 | t, df = 3 |
| 2     | 8 | 1000 | t, df = 10 | t, df = 10 | t, df = 10 |
| 3     | 40 | 1300 | K, df = 3 | t, df = 6 | N' |
| 4     | 8 | 1200 | N' | N' | N' |
| 5     | 6 | 1200 | 0.7N + 0.3G N', s = 0.1 | 0.6N + 0.4t, df = 2.3 | N' |

The distribution of each of the three clusters in each setup is specified. The distribution can be multivariate Gaussian (N'), generalized Gaussian (GN'), t-distribution or k-distribution. In the case of the latter three distributions, the extra parameters (do f or s) are indicated.

When estimating μ, the l₂ norm of the error is computed. The πₖ vectors, corresponding to the distribution proportions, are randomly chosen from a set of possibilities that avoid trivial and giant clusters. These cases are avoided due to the ill-posed clustering problem that it implies.

In these experiments, we include all the considered algorithms that estimate parameters. Thus, we exclude the comparison k-means, spectral clustering, and HDBSCAN. Table III shows the estimation error when estimating the main parameters of the model for all the setups.
metrics in Table IV. In a complementary manner, the plots of Fig. 3 visually summarize the distribution of these measures with boxplots. In most cases, the EM for GMM (GMM-EM) method has poor results and high variance.

In setups 1 and 2, the distributions are multivariate $t$–Student, and the difference between them is only in the degrees of freedom. In these setups, the proposed algorithm is referred to as the flexible EM algorithm (FEMCA), and $t$-EM error values are smaller than GMM-EM values. This increase in the predictive performance can be explained by the robustness of the estimators in the case of heavy-tailed distributions or the presence of outliers. It is interesting to confirm that, as in Setup 2, the considered distributions have larger degrees of freedom (tails are lighter), and GMM-EM performs much better than in Setup 1. However, while TCLUST and RIMLE perform similarly in Setup 1, RIMLE has a huge variance and
worse estimation in Setup 2. This phenomenon is due to the overestimation of points as noise/outliers. On the other hand, FEMCA and t-EM perform very similarly in both settings, with a slight improvement of FEMCA in the $\Sigma$ estimation. As shown in both tables, our robust algorithm performs similarly on average even in the $t-$distributed case where the t-EM algorithm is completely adapted. We remark that FEMCA performs very well, as expected, even if, in Setup 1, the traces are very different. Then, for Setups 3, in the case of a mixture of three different distributions ($t$-distribution, $t$-distribution, and Gaussian distribution), FEMCA outperforms the other algorithms in most runs. As Fig. 3 shows, there are only very few runs where FEMCA had a bad performance. Thus, it is important to notice that the model assumptions used to derive FEMCA, i.e., unknown $\tau_{ik}$’s and different distributions for each observation are very general. It allows successful handling of the case of mixtures of different distributions without additive computational cost, which appears to be an important contribution of this work. In Setup 4, in which uniform background noise in the cube $[0, 1]^m$ is included, the best performances are the ones from TCLUS and RIMLE, which appear reasonable since their design matches the data generation process. After them, FEMCA has a very good performance, considering that we do not reject outliers and, consequently, that those are intrinsically misclassified. When we exclude the noise for the metric computation, the classification performance is equally good for these three algorithms. However, the TCLUS algorithm is computed with the true proportion of outliers. Besides, the parameter estimation is equally good for FEMCA, RIMLE, and TCLUS. The performance analysis in this Setup (for which RIMLE and TCLUS are designed to provide the best performance) highlights the proposed algorithm’s flexibility and robustness. Finally, Setup 5 displays very good behaviour of FEMCA and RIMLE compared to the rest of the algorithms. The performance of EM-GMM is really bad there because it cannot deal with outliers coming from heavy tails. Combining two distributions for one cluster is difficult to fit for t-EM and TCLUS. The model is too general for t-EM, and TCLUS probably suffers from an insufficient noise rate to avoid the heavy tails.

To conclude, the proposed algorithm shows by design very stable performance among a wide range of cases. Indeed, when the data perfectly follows a specific model such as e.g., a mixture of $t$-distributions, the best algorithm will be the ML-based one (in this case, the t-EM algorithm). However, FEMCA does perform almost as well as the t-EM. But in various other scenarios (data drawn from different models, outliers in the data), FEMCA will clearly outperform traditional model-based algorithms that are not adaptive.

### B. Real Data

The proposed FEMCA algorithm has been tested on three different real data sets: MNIST [41], small NORB [42] and 20newsgroup [43]. The MNIST hand-written digits (Fig. 4) data set has become a standard benchmark for classification/clustering methods. We apply FEMCA to discover groups in balanced subsets of similar pairs of digits (3-8 and 1-7) and the set of digits (3-8-6). We additionally contaminate the later subset with a small proportion of noise by randomly adding some of the remaining different digits.

As in many application examples in the literature, we first applied PCA to work with some meaningful features instead of the original data [62]. We make a trade-off between explained variance and the curse of dimensionality effects. The dimension of the reduced data is shown in Table V under the column $m$. Because of the stochastic character of the algorithms, we run each of them multiple times $(nrep = 50)$, and we report the median value of the metrics. The metrics for FEMCA are almost always the same, which explains why we do not report the variance.

As can be seen in Tables VI, VII, and VIII, one obtains, in most cases, better values for all the metrics than those produced...
by the other partitioning techniques. This can be explained by the increment in flexibility and the smaller impact of outliers in the estimation process. More precisely, FEMCA does not provide the best results in these scenarios:

- MNIST 7-1 scenario for AMI and AR indices, where the \( t \)-EM performs the best,
- MNIST 3-8-6 and its noisy variation for the three criteria where the spectral clustering and TCLUST respectively perform the better.

The loss in performance of FEMCA is, in most cases, around or less than 1\%, highlighting the robustness of the approach: “better or strongly better than existing methods in most cases and comparable in other cases”. Moreover, those scenarios always correspond to the simpler scenarios, without noise and with well-separated clusters or completely designed to be managed by the best algorithm (MNIST 3-6-8 plus noise for the TCLUST).

We collected the clustering results from the HDBSCAN algorithms fed with a grid of values for its two main parameters. All the computed metrics comparing the results with the ground truth were poor, close to 0. We show the best clustering result of the 3-8 MNIST subset in Fig. 7, where the algorithm classifies a lot of data points as noise. The clustering is almost perfect if the metric is computed only in the non-noise labelled data points. This behaviour might be explained by the dimension of the data, which seems to be too high for HDBSCAN to deal with.

Additionally, we tested dimensional reduction techniques UMAP and t-SNE before the clustering task. All metrics were improved after carefully tuning the parameters. In this scenario, the proposed method performs similarly to the classical GMM-EM because these embedding methods tend to attract outliers and noise to clusters. However, these non-linear visualization approaches are not recommended to extract features before clustering because fictitious effects might appear depending on the choice of the parameters.

For the NORB dataset (some representatives are shown in Fig. 5), k-means, GMM-EM, spectral clustering, and UMAP+HDBSCAN do not perform satisfactorily since they end up capturing the luminosity as the main classification aspect. In contrast, \( t \)-EM and FEMCA highly outperform them, as can be seen in Tables VI, VII and VIII. This can be emphasized thanks to the results of Fig. 6, where label-colored two-dimensional embeddings of the data based on the classification produced by the different methods are shown. The effect of extreme light values seems to be palliated by the robustness properties of the estimators.

Finally, the 20news
group dataset is a bag of words constructed from a news corpus. Each piece of news is classified by topic modelling into twenty groups. Once again, we compare the performance of our methods with the ones of k-means, EM, \( t \)-EM, TCLUST, RIMLE, and spectral clustering algorithms after applying PCA. The corresponding results are also presented in Tables VI, VII and VIII. One can see that k-means, TCLUST, RIMLE and spectral clustering perform poorly, while GMM-EM and \( t \)-EM outperform them. Nevertheless, the proposed FEMCA has strongly better results than the others. It is unclear why spectral clustering is performing so poorly on this data set; it could be due to the lack of separation between clusters and/or the presence of noise that breaks the performance. Finally, the very poor capability of the RIMLE algorithm in this dataset is
explained by the choice of the parameter that highly overestimates the noise.

IV. CONCLUDING REMARKS

In this paper, we presented a robust clustering algorithm that outperforms several state-of-the-art algorithms for both synthetic and real diverse data. Its advantages stem from a general model for the data distribution, where each data point is generated by its own elliptical symmetric distribution. The good theoretical properties of this proposal have been studied and supported by simulations. The flexibility of this model makes it particularly suitable for analyzing heavy-tailed distributed and/or noise-contaminated data. Interestingly, under mild assumptions on the data, the estimated probabilities of membership do not depend on the data distributions, making the algorithm simpler (no need to re-estimate the likelihood at each step), flexible and robust. Moreover, the original approach of estimating one scale parameter for each data point makes the algorithm competitive in relatively high-dimensional settings.

On simulated data, we obtained accurate estimations and good classification rates. Of course, the best model is the one that perfectly coincides with the distribution of the data, e.g., when the mixture is actually Gaussian, GMM-EM outperforms all other methods, including ours, but only marginally. Our method is also algorithm competitive in relatively high-dimensional settings.

REFERENCES

[1] X. Liu, “Hyperparameter-free localized simple multiple kernel k-means with global optimum,” IEEE Trans Pattern Anal Mach Intell., vol. 45, no. 7, pp. 8566–8576, Jul. 2023.

[2] X. Liu, “SimpleMKM: Simple multiple kernel k-means,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 4, pp. 5174–5186, Apr. 2023.

[3] M. Zhang, N. Wang, Y. Li, and X. Gao, “Deep latent low-rank representation for face sketch synthesis,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 10, pp. 3109–3123, Oct. 2019.

[4] M. Zhang, N. Wang, Y. Li, and X. Gao, “Neural probabilistic graphical model for face sketch synthesis,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 7, pp. 2623–2637, Jul. 2020.

[5] M. Zhang, R. Zhang, J. Zhang, J. Guo, Y. Li, and X. Gao, “Dim2clear network for infrared small target detection,” IEEE Trans. Geosci. Remote Sens., vol. 61, pp. 1–14, 2023.

[6] C. Hennig, “Clustering strategy and method selection,” in Handbook of Cluster Analysis, C. Hennig, M. Meila, F. Murtagh, and R. Rocci Eds., Boca Raton, FL, USA: CRC Press, 2015.

[7] Scikit-learn developers, “Clustering—scikit-learn v0.20.3,” 2019. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#clustering

[8] G. McLachlan, “9 The classification and mixture maximum likelihood approaches to cluster analysis,” in Classification Pattern Recognition and Reduction of Dimensionality, ser. Handbook of Statistics, vol. 2, Amsterdam, The Netherlands: Elsevier, 1982, pp. 199–208. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016761618200124

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Roy. Stat. Soc., Ser. B, vol. 39, pp. 1–38, 1977. [Online]. Available: http://web.mit.edu/6.435/www/Dempster77.pdf

[10] C. Fraley and A. E. Raftery, “Model-based clustering, discriminant analysis, and density estimation,” J. Amer. Stat. Assoc., vol. 97, no. 458, pp. 611–632, 2002. [Online]. Available: http://www.jstor.org/stable/2533502

[11] R. A. Maronna, “Robust M-Estimators of multivariate location and scatter,” Ann. Statist., vol. 4, no. 1, pp. 51–67, 1976. [Online]. Available: http://www.jstor.org/stable/2957994

[12] L. A. García-Escudero, A. Górriz, M. Marín, and A. Mayo-Iscar, “A general trimming approach to robust cluster analysis,” Ann. Statist., vol. 36, no. 3, pp. 1324–1345, 2008. [Online]. Available: https://doi.org/10.1214/07-AOS1154

[13] C. Bouveyron and C. Brunet-Saumard, “Model-based clustering of high-dimensional data: A Review,” Comput. Stat. Data Anal., vol. 71, pp. 52–78, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947312004422

[14] G. Celeux and G. Govaert, “Gaussian parsimonious clustering models,” Pattern Recognit., vol. 28, no. 5, pp. 781–793, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320394001256

[15] P. McNicholas, Mixture model-based classification, Chapman and Hall/CRC, 2016.

[16] D. Peel and G. J. McLachlan, “Robust mixture modelling using the t distribution,” Statist. Comput., vol. 10, no. 4, pp. 339–348, Oct. 2000. [Online]. Available: https://doi.org/10.1023/A:1009891500181

[17] Y. Wei, Y. Tang, and P. D. McNicholas, “Mixtures of generalized hyperbolic distributions and mixtures of skew-t distributions for model-based clustering with incomplete data,” Comput. Stat. Data Anal., vol. 130, no. C, pp. 18–41, 2017.

[18] R. P. Browne and P. D. McNicholas, “A mixture of generalized hyperbolic distributions,” Can. J. Statist., vol. 43, no. 2, pp. 176–198, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cjs.11246

[19] S. Lee and G. J. McLachlan, “Finite mixtures of multivariate skew t-distributions: Some recent and new results,” Statist. Comput., vol. 24, no. 2, pp. 181–202, Mar. 2014. [Online]. Available: https://doi.org/10.1007/s11222-012-9362-4

[20] J. D. Banfield and A. E. Raftery, “Model-based Gaussian and non-Gaussian clustering,” Biometrics, vol. 49, no. 3, pp. 803–821, 1993. [Online]. Available: http://www.jstor.org/stable/2532201

[21] P. Coretto and C. Hennig, “Consistency, breakdown robustness, and algorithms for robust improper maximum likelihood clustering,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 5199–5237, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3127009.3128023

[22] K. Yu, X. Dang, H. Bart, and Y. Chen, “Robust model-based learning via spatial-EM algorithm,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 6, pp. 1670–1682, Jun. 2015.

[23] J. D. González, V. J. Yohai, and R. H. Zamar, “Robust clustering using Tau-scales,” Jun. 2019, arXiv:1906.08198.

[24] J. D. González, “Métodos de clustering robustos,” Ph.D. dissertation, Universidad de Buenos Aires, 2019.

[25] N. A. Campbell, “Mixture models and atypical values,” J. Int. Assoc. Math. Geol., vol. 16, no. 5, pp. 465–477, 1984. [Online]. Available: https://doi.org/10.1007/BF01886327
[26] S. Tadjudin and D. A. Landgrebe, “Robust parameter estimation for mixture model,” IEEE Trans. Geosci. Remote Sens., vol. 38, no. 1, pp. 439–445, Jan. 2000.

[27] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, “EM algorithms for weighted data clustering with application to audio-visual scene analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 12, pp. 2402–2415, Dec. 2016. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2522452

[28] R. Nock and F. Nielsen, “On weighting clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 8, pp. 1223–1235, Aug. 2006.

[29] G. Boezaart, M. S. Barnard, and D. E. Tyler, “A characterization of elliptical distributions and some optimality properties of principal components for functional data,” J. Multivariate Anal., vol. 131, pp. 254–264, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0047259X14001618

[30] E. Ollila, D. Tyler, V. Koivunen, and H. Poor, “Complex elliptically symmetric distributions: Summary, new results and applications,” IEEE Trans. Signal Process., vol. 60, no. 11, pp. 5597–5625, Nov. 2012.

[31] E. Conte, A. De Maio, and G. Ricci, “Covariance matrix estimation for adaptive CFAR detection in compound-Gaussian clutter,” IEEE Trans. Aerosp. Electron. Syst., vol. 38, no. 2, pp. 415–426, Apr. 2002.

[32] F. Gini, M. V. Greco, M. Diiani, and L. Verrazzani, “Performance analysis of two adaptive radar detectors against non-Gaussian real sea clutter data,” IEEE Trans. Aerosp. Electron. Syst., vol. 36, no. 4, pp. 1429–1439, Oct. 2000.

[33] F. Gini and A. Farina, “Vector subspace detection in compound-Gaussian clutter. part I: Survey and new results,” IEEE Trans. Aerosp. Electron. Syst., vol. 38, no. 4, pp. 1295–1311, Oct. 2002.

[34] E. Conte and M. Longo, “Characterisation of radar clutter as a spherically invariant random process,” IEEE Proc. F – Commun., Radar Signal Process., vol. 134, no. 2, pp. 191–197, 1987.

[35] E. Conte, A. De Maio, and G. Ricci, “Recursive estimation of the covariance matrix of a compound-Gaussian process and its application to adaptive CFAR detection,” IEEE Trans. Signal Process., vol. 50, no. 8, pp. 1908–1915, Aug. 2002.

[36] F. Pascal, L. Bombrun, J.-Y. Tourneret, and Y. Berthoumieu, “Parameter estimation for multivariate generalized Gaussian distributions,” IEEE Trans. Signal Process., vol. 61, no. 23, pp. 5960–5971, Dec. 2013.

[37] V. Roisman, M. Jonckheere, and F. Pascal, “Robust clustering and outlier rejection using the mahalanobis distance,” in Proc. 28th Eur. Signal Process. Conf., 2021, pp. 2448–2452.

[38] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, “Hierarchical density estimates for data clustering, visualization, and outlier detection,” ACM Trans. Knowl. Discov. Data, vol. 10, no. 1, pp. 5:1–5:51, 2015. [Online]. Available: http://doi.acm.org/10.1145/2733381

[39] L. McInnes and J. Healy, “Accelerated hierarchical density based clustering,” in Proc. IEEE Int. Conf. Data Mining Workshops, 2017, pp. 33–42.

[40] L. McInnes, J. Healy, and S. Astels, “HDDBSCAN: Hierarchical density based clustering,” J. Open Source Softw., vol. 2, no. 11, 2017, Art. no. 205.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[42] Y. LeCun and L. Bottou, “Learning methods for generic object recognition with invariance to pose and lighting,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2004, pp. II–104.

[43] T. M. Mitchell, Machine Learning, 1st ed., New York, NY, USA: McGraw-Hill, 1997.

[44] Z. Liao and R. Couillet, “A large dimensional analysis of least squares support vector machines,” IEEE Trans. Signal Process., vol. 67, no. 4, pp. 1065–1074, Feb. 2019.

[45] K. Yao, “A representation theorem and its applications to spherically invariant random processes,” IEEE Trans. Inf. Theory, vol. 19, no. 5, pp. 600–608, Sep. 1973.

[46] E. Ollila and D. E. Tyler, “Distribution-free detection under complex elliptically symmetric clutter distribution,” in Proc. IEEE 7th Sensor Array Multichannel Signal Process. Workshop, 2012, pp. 413–416.

[47] H. Jeffreys, “An invariant form for the prior probability in estimation problems,” Proc. R. Soc. London, vol. 186, pp. 453–461, 1946.

[48] C. Wu, C. Yang, H. Zhao, and J. Zhu, “On the convergence of the EM algorithm: A data-adaptive analysis,” 2016, arXiv:1611.00519.

[49] D. E. Tyler, “A distribution-free –estimator of multivariate scatter,” Ann. Statist., vol. 15, no. 3, pp. 1394–1412, May 1987.

[50] J. Frontera-Pons, M. Vazquezano, F. Pascal, and J.-P. Ovarlez, “Hyperspectral anomaly detectors using robust estimators,” IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens., vol. 9, no. 2, pp. 720–731, Feb. 2016.

[51] M. Bilodeau and D. Brenner, Robustness, New York, NY, USA: Springer, 1999, pp. 206–242. [Online]. Available: https://doi.org/10.1007/978–0–387–22616–3_13

[52] J. T. Kent et al., “Rescoring –estimates of multivariate location and scatter,” Ann. Statist., vol. 19, no. 4, pp. 2102–2119, 1991.

[53] F. Pascal, Y. Chiotour, J.-P. Ovarlez, F. Forster, and P. Lazard, “Covariance structure maximum-likelihood estimates in compound Gaussian noise: Existence and algorithm analysis,” Trans. Sig. Proc., vol. 56, no. 1, pp. 34–48, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1109/TSP.2007.901652

[54] A. T. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm” in Proc. 14th Int. Conf. Neural Inf. Process. Syst. Natural Synthetic, Cambridge, MA, USA: MIT Press, 2001, pp. 849–856. [Online]. Available: http://dl.acm.org/citation.cfm?id=2980539.2980649

[55] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[56] K. Wang, S. Ng, and G. J. McLachlan, “Multivariate skew t mixture models: Applications to fluorescence-activated cell sorting data,” in Proc. Digit. Image Comput.: Techn. Appl., 2009, pp. 526–531.

[57] P. Coretto and C. Hennig, “OTrimle: Robust model-based clustering,” R package version, 2.0.20.

[58] P. J. Rousseuw, “Silhouettes: A graphical aid to the interpretation and validation of clustering analysis,” J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377042787901257

[59] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” J. Mach. Learn. Res., vol. 11, pp. 2837–2854, Dec. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1953024

[60] L. M. Weber and M. D. Robinson, “Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data,” Cytometry Part A, vol. 89, no. 12, pp. 1084–1096, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23030

[61] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” J. Open Source Softw., vol. 3, no. 29, p. 1861, 2018.

[62] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. Nov, pp. 2579–2605, 2008.