Assessing clustering methods using Shannon's entropy

Anis Hoayek, Didier Rullière

Assessing clustering methods using Shannon's entropy

Anis Hoayek* and Didier Rullière*

November 9, 2023

Abstract

Unsupervised clustering techniques are a valuable source of information for determining how to divide a dataset into subgroups. We present a comprehensive analysis of the quality of these algorithms by defining a clustering fuzziness metric. A statistical test and cluster probabilities corrections are provided based on this metric. Some examples demonstrate how it can be used to compare different clustering algorithms or improve the accuracy of various methods. An application for adjusting the number of clusters is also presented. These results are illustrated using both simulated and real-world data.

1 Introduction

In unsupervised learning, clustering methods are very popular methods that associate each observation to a cluster index. Such methods are known as hard clustering approaches. In situations where some observations are difficult to associate with certainty to specific clusters, fuzzy clustering may be used: fuzzy clustering methods, known also as soft clustering methods, associate to each observation the probability to belong to each possible cluster index, or more generally some membership degrees (see e.g Ruspini et al., 2019; Yang, 1993; De Oliveira and Pedrycz, 2007, among many other references). There is a wide diversity of clustering methods: a first subdivision is to distribute clustering methods into two families: 1) partitional clustering algorithm (e.g. k-means, density based clustering, genetic algorithm and many other methods), (see e.g. MacQueen (1967); Kriegel et al. (2011); Forrest (1996); among many other references) in which data is organized into a sequence of groups without any hierarchical structure (Ezugwu et al. (2022)); 2) hierarchical clustering algorithm (e.g. Linkage algorithm, divisive clustering), (see e.g. Dawyndt et al. (2005); Roux (2015)) .

A problem for fuzzy clustering is to compare different available methods. Are they *overconfident*, as in the case where proposed probabilities are all close to 0 or 1? are they *underconfident*, as in the case where each observation can belong to any cluster, with equal probabilities? how to measure this confidence of the clustering method? how to compare clustering methods? how to propose corrections to proposed probabilities in case of over or underconfident clustering method?

The confidence of a clustering method is indeed of practical interest, beyond the quality of a classification. If the clustering method is too uncertain, too fuzzy (underconfident), it may induce avoidable checks (medical investigations for example, with costs or adverse effects). If the clustering method is too categorical (overconfident), it may disable useful alerts (need of medical checks for example). Quantifying this *under/overconfidence*, or equivalently *too fuzzy/not fuzzy enough* characteristic of a clustering method requires the definition of some kind of fuzziness level.

A goal of the present paper is to carefully define the fuzziness of a clustering method. As demonstrated in various sections of this paper, defining such a fuzziness level allows for the comparison of

*Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne France

fuzzy clustering methods, determining whether a clustering method is compliant with a given probabilistic model, improving classification or clustering accuracy on test samples for both supervised and unsupervised learning, and assisting in the selection of a hidden number of clusters.

In the literature, to our knowledge there is very few works on the fuzziness assessment of a clustering method. Some existing indices, as Dave's, Bezdek, and Xi-Beni validity measurement indices (see for example Bataineh et al., 2011), do not directly assess the confidence of the clustering method.

However, some papers in the literature deals with overconfidence issue, see Park et al. (2021), in a specific context of computer vision. Aghababyan et al. (2018) proposed an entropy based metric to evaluate the confidence of clustering algorithm without proposing any correction of the underlying probabilities in case of over/underconfident methods. Yao et al. (2000) introduced a new fuzzy clustering algorithm based on entropy without investigating about the confidence level of the proposed algorithm with respect to other state of the art methods. In addition, to the best of author's knowledge no work has been done on proposing a statistical hypothesis test to decide about accepting or rejecting a clustering method based on its confidence level.

In this paper an entropy based clustering confidence metric is introduced. Based on this metric one will be able to compare the performance of any two clustering algorithms. A statistical test is also introduced to decide about accepting or rejecting a clustering method. Furthermore, in the context of over/underconfident clustering method, a parametric correction of the underlying probabilities is proposed in order to improve the accuracy of a clustering. An application for adjusting the number of clusters is also presented.

The outline of the paper is as follows. We first introduce in Section 2, a metric to measure clustering fuzziness using entropy, for a theoretical mixture model as well for a practical clustering algorithm. A Section 3 is devoted to the comparison of clustering fuzziness level of different clustering algorithms, with some applications based on numeric simulations. A statistical test helping users to decide about accepting or rejecting a given clustering algorithm is also described. Then, in Section 4, we propose a parametric correction method of the underlying probabilities of under/overconfident clustering algorithms. Applications of the whole methodology on real data are shown in Section 5. A conclusion closes the paper.

2 Measuring clustering fuzziness

Let $\mathcal{I}$ be a finite set of cluster indexes. In this work, $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space and for any $i \in \mathcal{I}$, X_i is a $\mathbb{R}^d$ valued random variable defined on Ω with a cumulative distribution function $F_i(\cdot)$, and a probability density function $f_i(\cdot)$ with respect to the Lebesgue measure in $\mathbb{R}^d$.

These random variables X_i, $i \in \mathcal{I}$ have distinct distributions, as they correspond to individuals of different clusters. Assume that one chooses a cluster index with a random variable I taking values in $\mathcal{I}$. One labelled observation of the corresponding individual is given by the couple (X_I, I). The purpose of a clustering procedure is to retrieve the association between observations of the mixture distribution X_I and labels I when the labels are lost.

In this paper, we aim at assessing the fuzziness of different clustering methods, and at comparing it with expected values when they are known.

Notice that the precise value of the labels has no impact, so that a perfect clustering procedure can associate I or any permutation $\sigma(I)$ to the observations X_I: the fuzziness measure should be insensitive to permutations of I. It should also allow to compare clustering that use different numbers of clusters. We present hereafter a measure and some tests based on Shannon's entropy.

An originality of this section is that the fuzziness level associated to a clustering will be represented by the distribution of a real-valued random variable. This will provide a detailed information on the clustering fuzziness, while allowing comparison between these fuzziness levels.

2.1 Theoretical entropy

A labeled clustering is given by the joint distribution of (X_I, I). When the labels are lost, we aim at inferring the values of the random variable I given the observations X_I.

Recall that $\mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}$ is the finite set of all possible cluster indexes. Consider the finite mixture distribution given by:

$$\mathcal{G} := X_I = \sum_{i \in \mathcal{I}} X_i \mathbb{1}_{\{I=i\}}, \text{ with } I \in \mathcal{I} \text{ and each } X_i \sim F_i,\ i \in \mathcal{I}.$$

F_i is the cumulative distribution function (*cdf*) of the random variable (*rv*) X_i.

Then, the *cdf* $F_{\mathcal{G}}$ and the probability density functions (*pdf*) $f_{\mathcal{G}}$ of $\mathcal{G}$ are as follows:

$$\begin{cases} F_{\mathcal{G}}(x) = \sum_{i \in \mathcal{I}} \alpha_i F_i(x), \\ f_{\mathcal{G}}(x) = \sum_{i \in \mathcal{I}} \alpha_i f_i(x), \end{cases}$$

where, f_i is the *pdf* of X_i, with $\alpha_i = \mathbb{P}[I = i]$ and $\sum_{i \in \mathcal{I}} \alpha_i = 1$.

Therefore, the probability that a given observation x is sampled from the underlying *rv* X_i is:

$$\begin{aligned} p_i(x) &:= \mathbb{P}[I = i \mid \mathcal{G} = x], \\ &= \frac{\alpha_i f_i(x)}{\sum_{j \in \mathcal{I}} \alpha_j f_j(x)}. \end{aligned}$$

Now, considering the joint random variables $(\mathcal{G}, I)$, we compute Shannon's entropy of the *rv* I given $\mathcal{G} = x$, measuring the information on the fact that an observation x is sampled from the distribution F_i, $i \in \mathcal{I}$:

$$\mathcal{H}_I(x) = -\sum_{i \in \mathcal{I}} p_i(x) \log_2 p_i(x),$$

under the convention that $0 \log_2 0 = 0$.

Recall that I is the hidden cluster index associated to X_I. This index I will be unknown in practice. The former entropy $\mathcal{H}_I(x)$ measures the uncertainty of the clustering at a given point x. When it is equal to 0, e.g. in the case when $p_i(x) = 1$ for a given i and 0 otherwise, then it is certain that x was sampled from a specific known index i. This entropy is maximal when all $p_i(x)$ are equal, so that one has totally lost the information about the index I that has generated the observation x.
This entropy is insensitive to cluster index permutations, which is a desirable property, as stated in the introduction.

Consider the function $\mathcal{H}_I : x \mapsto \mathcal{H}_I(x) \in \mathbb{R}^+$. Applied to a random argument $\mathcal{G}$, this function defines the random variable $\mathcal{H}_I(\mathcal{G}) \in \mathbb{R}^+$. $\mathcal{H}_I(\mathcal{G})$ measures the uncertainty of the clustering at a random point having the same distribution as $\mathcal{G}$.

We give below an original definition of the theoretical fuzziness level of a known mixture model.

Definition 1 (Theoretical fuzziness level)**.** *For a given mixture distribution $\mathcal{G}$ and associated labels I, i.e. given the joint distribution of $(\mathcal{G}, I)$, the theoretical fuzziness level is defined as the non-negative real-valued random variable $\mathcal{H}_I(\mathcal{G})$.*

The distribution of this random variable shows the discriminating power of the considered clustering approach: e.g., the higher its mean, the more ambiguous the situation is (high fuzziness). However, a mean close to zero reflects a clustering where one expects to easily associate a unique label to each point x (low fuzziness). On the other hand, a low dispersion shows that the difficulty of cluster labeling is the same for all points x, whereas a high dispersion indicates that some points are easier to label

than others.

It shall be noted that, for a given distribution $\mathcal{G}$, several distributions of I and X_i, $i \in \mathcal{I}$ can lead to the same mixture $\mathcal{G} = X_I$.

As an example, picking a uniform r.v. $U(0,1)$ with probability $1/2$, and a uniform r.v. $U(0,0.5)$ with probability $1/2$ leads to the same mixture as an uniform r.v. $U(0,0.5)$ with probability $3/4$ and $U(0.5,1)$ with probability $1/4$, but the conditional distribution of I given $\mathcal{G}$ are changed.

Another example is the (overlapping) mixture of Gaussian distributions, that can also be modelled as a non-overlapping mixture of truncated Gaussian distributions; this leads to what is sometimes called a bias or an inconsistency for k-means unsupervised clustering, as the mean of any resulting truncated Gaussian differs from the one of the initial Gaussian (see Jin and Malthouse (2016)).

Thus in some cases, identifiability problems may occur when the joint distribution $(\mathcal{G}, I)$ is unknown. This problem however depend on the families of possible distributions F_i, $i \in \mathcal{I}$: for example, it does not occur if all F_i are assumed to be Gaussian distributed, but it may occur if all F_i can be overlapping truncated Gaussian distributed. Such problems will be discussed in a further Section 5, when F_i are unknown. A short review on identifiability problem for finite mixture models is given in McLachlan et al. (2019). We assume in this section that the joint distribution $(\mathcal{G}, I)$ is known, so that all F_i are known, $i \in \mathcal{I}$, and no identifiability problem raises here.

To illustrate the function $\mathcal{H}_I(.)$ we consider a few basic examples:

Case A Let $\mathcal{G}$ be a Gaussian mixture distribution in one dimension with a *pdf*:

$$f_{\mathcal{G}}(x) = 0.3 f_{\mathcal{N}(0,1)}(x) + 0.5 f_{\mathcal{N}(10,1)}(x) + 0.2 f_{\mathcal{N}(3,0.1)}(x).$$

where $f_{\mathcal{N}(.,.)}$ denotes the pdf of a Gaussian r.v. with indicated parameters. The left side of Figure 1 represents the *pdf* of $\mathcal{G}$, while the right side illustrates the function $\mathcal{H}_I$. One can remark that the labeling is perfectly accurate when the values x of $\mathcal{G}$ are far from the central area of the distribution (i.e. left and right queues of the mixture distribution). However, the difficulty of cluster labeling is higher for central values where the region is fuzzy in terms of distribution selection.

Case B Let $\mathcal{G}$ be a Dirac mixture distribution in one dimension with a probability distribution:

$$\mathrm{P}[\mathcal{G} = x] = 0.3 \mathbb{1}_{\{x=0\}} + 0.7 \mathbb{1}_{\{x=3\}}$$

where, $\mathbb{1}_{\{x=a\}}$, $a \in \mathbb{R}$ is the indicator function defined by:

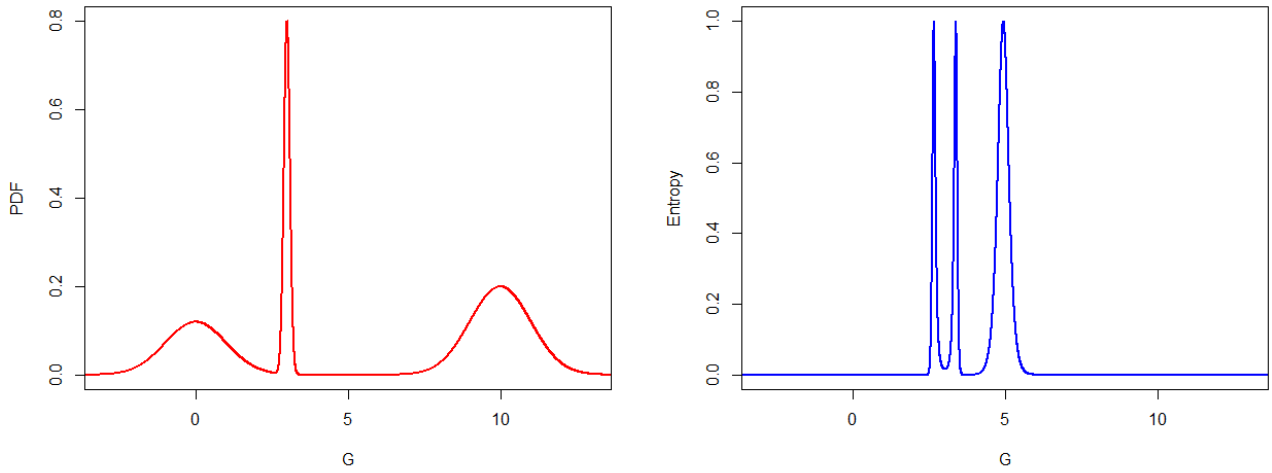
$$\mathbb{1}_{\{x=a\}} = \begin{cases} 0 & \text{if } x \neq a \\ 1 & \text{if } x = a \end{cases}.$$

The left side of Figure 2 represents the probability masses of $\mathcal{G}$, while the right side illustrates the function $\mathcal{H}_I$. One can remark that the labeling is perfectly accurate for all values of x. This is well suited to the Dirac case where the discrimination between different clusters/labels is obvious.

Case C Let $\mathcal{G}$ be a two dimensional Gaussian mixture distribution with a *pdf*:

$$f_{\mathcal{G}}(x) = 0.4 f_{\mathcal{N}(\mu_1, \Sigma_1)} + 0.6 f_{\mathcal{N}(\mu_2, \Sigma_2)},$$

where, $\mu_1 = (0,0)^T$ and $\mu_2 = (4,4)^T$. In addition $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\Sigma_3 = \begin{pmatrix} 2 & 0.4 \\ 0.4 & 2 \end{pmatrix}$.
The left side of Figure 3 represents the contour lines of the *pdf* of $\mathcal{G}$, while the right side illustrates the contour lines of $\mathcal{H}_I$. Once again, one can say that the accuracy of the labeling is lower



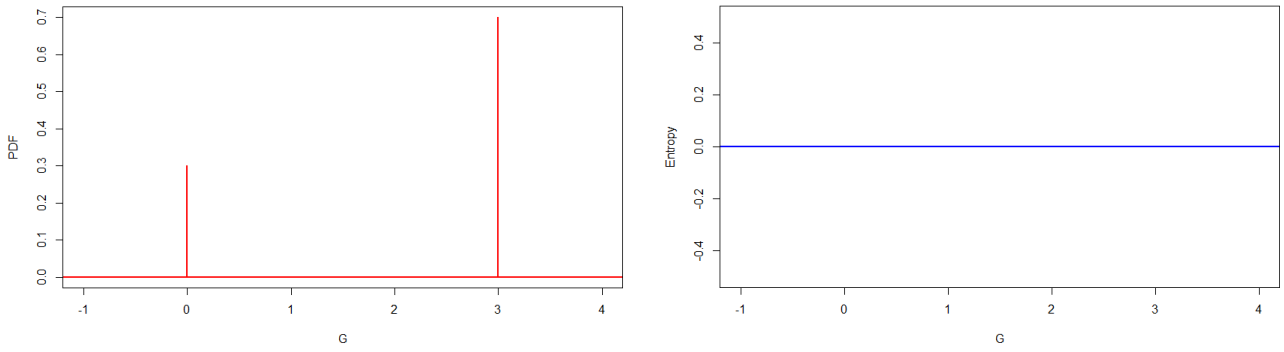
(a) *pdf* of the mixture distribution

(b) entropy $\mathcal{H}_I$

Figure 1: *Gaussian mixture distribution of Case A. Left: pdf of the mixture distribution $\mathcal{G}$, right: entropy $\mathcal{H}_I$ representing the variation of the labeling difficulty.*

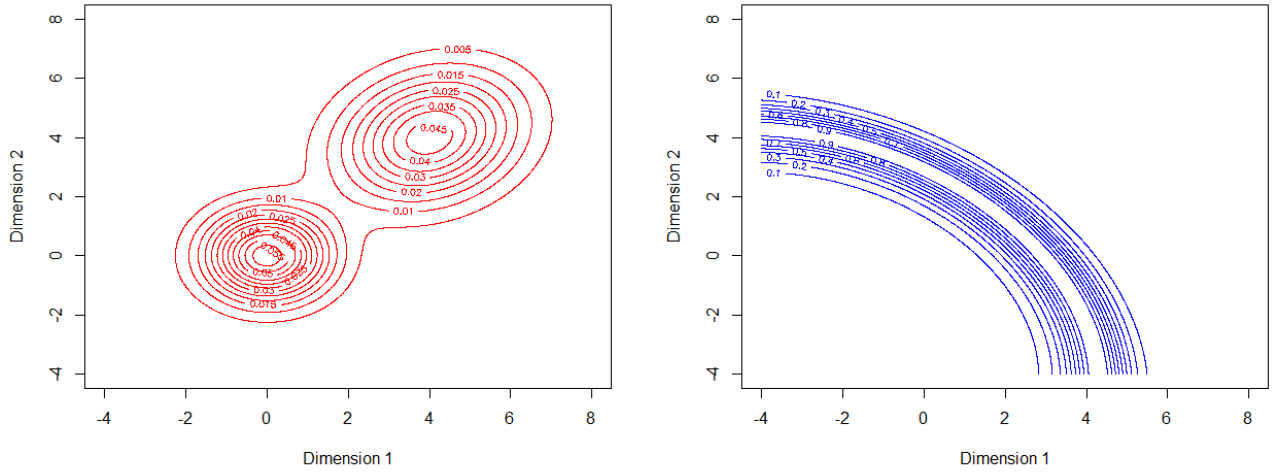
on zones that are with a high fuzziness level (e.g. between the two modes), and thus difficult to label.

In Figure 4 we are showing the impact of increasing the variance of the underlying distribution of a mixture model on the distribution of the entropy by showing the corresponding entropy's boxplot. Hence, the mixture distribution of Case C is considered and the matrices of the underlying distributions are multiplied by a parameter α. The considered values of α are 1, 4 and 8 respectively. One may remark that when the values of α are increasing the fuzziness level is going to be higher with a lower labeling accuracy, which confirms that the entropy is a suitable tool to show the impact of the variance increase on the global fuzziness level.



(a) probability masses (b) zero-valued entropy $\mathcal{H}_I$

Figure 2: *Dirac mixture distribution of Case B. Left: probability masses of the mixture distribution $\mathcal{G}$, right: zero-valued entropy $\mathcal{H}_I$ representing the trivial labeling of any observation of $\mathcal{G}$.*



(a) Contour lines of the *pdf* (b) Contour lines of the entropy $\mathcal{H}_I$

Figure 3: *Two dimensional Gaussian mixture distribution of Case C Left: pdf of the mixture distribution $\mathcal{G}$, right: entropy $\mathcal{H}_I$ representing the variation of the labeling difficulty.*

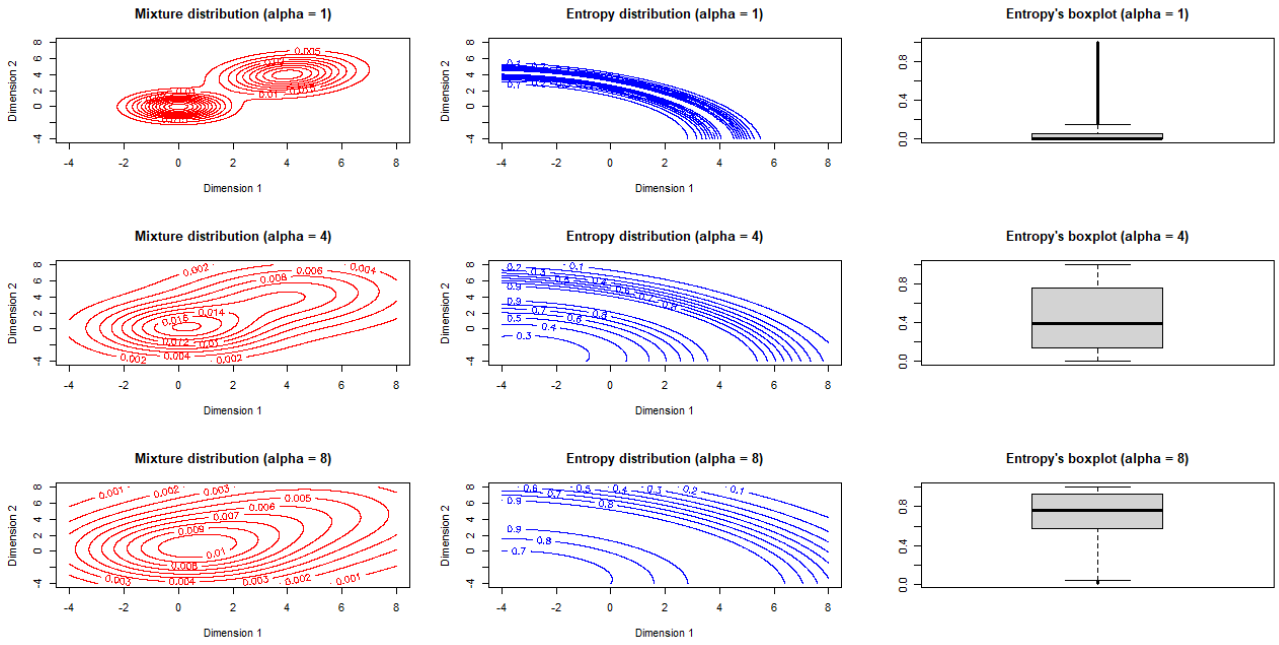


Figure 4: Entropies of two dimensional Gaussian mixture distribution of Case C for different scales

2.2 Empirical entropy

In practice, even when the true mixture $\mathcal{G}$ is unknown, one can derive an associated empirical fuzziness level from both a dataset and a clustering algorithm. Indeed, the dataset provides an empirical distribution of the underlying mixture, and the clustering algorithm provides associated weights, summing to one, that can be used to define an underlying random index. The aim of this section is to define the empirical fuzziness level deriving from both a dataset and a clustering algorithm.

Notice that the weights (or membership grades) in traditional fuzzy clustering methods represent the degree to which data points belong to each cluster. These weights have not originally been meant to be probabilities, or to estimate underlying probabilities. But in some situations, they need to be used as probabilities, or are implicitly used as probabilities. We investigate here the resulting fuzziness level of these weights, in order to compare them to other fuzziness levels. The purpose is not to depreciate one method or another, but to understand the implicit fuzziness level of each method (and associated parameters), and to propose possible clustering weights corrections.

The considered data is an *iid* sequence of random variables $\{G_1, \ldots, G_n\}$, sampled from a mixture distribution $\mathcal{G}$. Let $\mathcal{G}_n$ be a *rv* distributed as this data (e.g. a uniformly randomly chosen element of the sequence).

Even when $\mathcal{G}$ is unknown, a clustering algorithm associate to each data point x the probability that it belongs to different clusters of a set $\mathcal{J}$. Denote by $p_j(x)$ the probability that a data point x belongs to a cluster $j \in \mathcal{J}$, according to the clustering algorithm.

For any data point x, given $\mathcal{G}_n = x$, the probabilities $\{p_j(x) : j \in \mathcal{J},\}$ define the marginal distribution of a *rv* J_n. Hence the joint distribution of a couple of *rv* $(\mathcal{G}_n, J_n)$ can be defined from the data and the clustering algorithm. Based on these probabilities, given $\mathcal{G}_n = x$, the Shannon's entropy of

the random variable J_n is:

$$\mathcal{H}_{J_n}(x) = -\sum_{j\in\mathcal{J}} p_j(x)\log_2 p_j(x).$$

Now, consider the function $\mathcal{H}_{J_n} : x \mapsto \mathcal{H}_{J_n}(x) \in \mathbb{R}^+$. Applied to a random argument $\mathcal{G}_n$, this function $\mathcal{H}_{J_n}(.)$ defines a *rv* $\mathcal{H}_{J_n}(\mathcal{G}_n)$.

For a given clustering algorithm, we give below the resulting original definition of the empirical fuzziness level.

Definition 2 (Clustering empirical fuzziness level)**.** *Let $\mathcal{G}_n$ be a rv distributed as the empirical data, and J_n be the associated random labels according to a given clustering algorithm. Given the joint distribution of $(\mathcal{G}_n, J_n)$, the clustering empirical fuzziness level is defined as the non-negative real-valued random variable $\mathcal{H}_{J_n}(\mathcal{G}_n)$.*

The higher the average values of the random variable $\mathcal{H}_{J_n}(\mathcal{G}_n)$, the higher the fuzziness level of the underlying clustering, the higher the labeling difficulty on a random point of the data.
In the next section, we develop the analysis and compare this fuzziness level to a theoretical one, given an underlying mixture model.

3 Comparing clustering fuzziness

We aim here at comparing two clustering fuzziness levels, or equivalently at comparing two clustering confidence levels. Using the fuzziness levels defined in the former section 2, this ends up in comparing the distribution of two entropies.

Comparing the fuzziness level of theoretical or empirical distributions can be beneficial in many ways: it can aid in the selection of one fuzzy clustering algorithm over another, it can aid in the tuning of some fuzziness parameter, it allows for the comparison of the performance of several algorithms exhibiting the same fuzziness level, and it also allows for the correction of cluster weights in order to comply with some probabilistic model. As detailed in numerical illustration, it will also help improving the accuracy of clustering algorithm on some test samples.

As a first illustration, we present here the case where the theoretical distribution $(\mathcal{G}, I)$ is known, i.e. when one knows the underlying reality of the data. The case of real data with unknown distribution will be treated in a further Section 5. We compare here a theoretical fuzziness level with an empirical fuzziness level. A direct comparison between two empirical fuzziness level, even when the theoretical mixture is unknown, would also be possible, although not treated here.

The idea we develop below is that a fuzzy clustering algorithm, applied to an *iid* sample of $\mathcal{G}$, should end up with an entropy distributed as the random variable $\mathcal{H}_I(\mathcal{G})$. If it concludes with a lower mean entropy, then the clustering algorithm exhibits too low fuzziness, is too overconfident in its associations/labeling. If it concludes with a higher mean entropy, then the clustering algorithm has a too high fuzziness level and is too hesitating. Indeed, the distribution of $\mathcal{H}_I(\mathcal{G})$ reflects the difficulty of the labeling problem. In fact, depending on the underlying structure of $\mathcal{G}$, some clustering algorithms may perform better then others regarding this fuzziness level. We first assume that the joint random distribution $(\mathcal{G}, I)$ is given, so that no identifiability issue raises here. Notice that we will be able to compare algorithms that associate a cluster index probability for each point, and algorithms that associate a unique label to each point.

We aim at comparing the distributions of the two random variables $\mathcal{H}_I(\mathcal{G})$ and $\mathcal{H}_{J_n}(\mathcal{G}_n)$. Therefore, in order to compare the entropy distributions of these random variables, a distance (or a dissimilarity)

is calculated. Furthermore, as the distribution of $\mathcal{H}_I(\mathcal{G}_n)$ is easier to obtain than the one of $\mathcal{H}_I(\mathcal{G})$, which requires a numerical integration, we will consider in practice the following dissimilarity:

$$D_n := D\left(\mathcal{H}_I(\mathcal{G}_n) \,||\, \mathcal{H}_{J_n}(\mathcal{G}_n)\right),$$

where $D(.)$ is the chosen dissimilarity. Note that, the distribution of $\mathcal{H}_I(\mathcal{G}_n)$ represents the theoretical labeling difficulty for the mixture distribution (at data points), whereas $\mathcal{H}_{J_n}(\mathcal{G}_n)$ represents the perceived labeling difficulty of the clustering algorithm on the data.

The fuzzy clustering weights will be more in line with the underlying theoretical model as the distance decreases. A zero distance means that the clustering algorithm delivers the true underlying cluster probability at each data point. The choice of using here $\mathcal{H}_I(\mathcal{G}_n)$ instead of $\mathcal{H}_I(\mathcal{G})$ also relies on this zero distance property.

In the next subsections, we compare clustering empirical fuzziness with some underlying known theoretical one, using either a Kolmogorov-Smirnov distance or a Jensen-Shannon Divergence.

3.1 Comparison using Kolmogorov-Smirnov distance

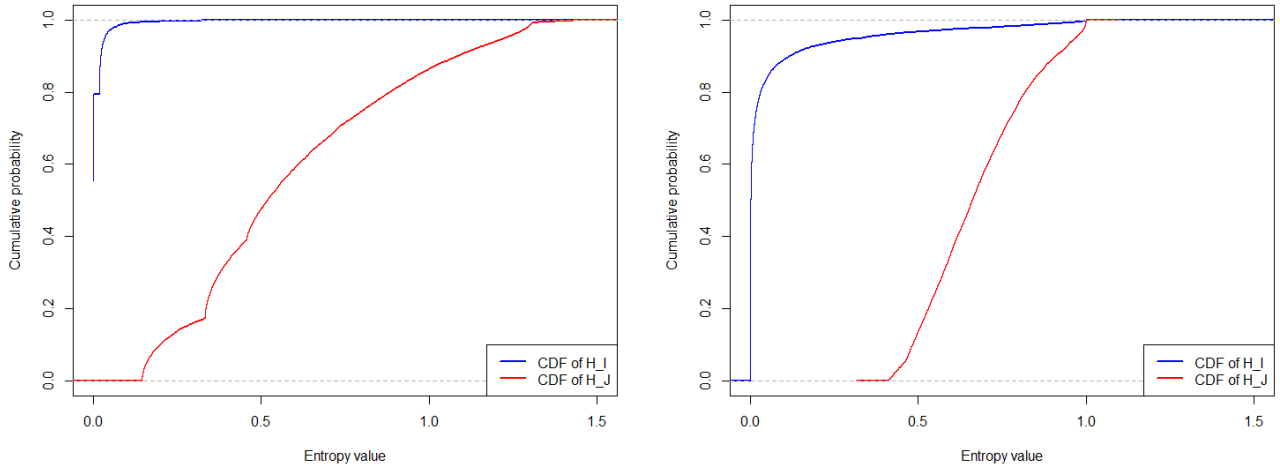
In this subsection, we compare some theoretical and empirical fuzziness level, using Kolmogorov-Smirnov distance. To do this, we compute the following statistic:

$$\mathrm{KS}_n := D\left(\mathcal{H}_I(\mathcal{G}_n) \,||\, \mathcal{H}_{J_n}(\mathcal{G}_n)\right). \tag{1}$$

We reconsider the first two examples of Section 2.1 and we test the proposed approach on two clustering methods, using Kolmogorov-Smirnov distance (KS):

1. In the first application, the k-means clustering approach is applied on a sample of $n = 10\,000$ observations generated from the mixture distribution $\mathcal{G}$. In the one dimensional Gaussian mixture model (Case A), the cardinal of $\mathcal{J}$ was considered to be equal to three. Kolmogorov-Smirnov statistics has a value of $\mathrm{KS}_n = 0.43$ with a $p-$value of $\approx 0\%$. Then, one can say that the two distributions are significantly different and that the proposed clustering method fails to identify the labeling difficulty. This is expected as a hard clustering here is clearly overconfident in its labeling. However, the k-means applied on Dirac mixture underlying distribution (Case B), with $\mathcal{J} = 2$, gives $\mathrm{KS}_n \approx 0$ and a $p-$value of $\approx 100\%$. Then, in the Dirac case, k-means succeed perfectly in re-finding the original labels of the observations. Note that these results fit well with the nature of the k-means algorithm, which proposes a labeling with no degree of fuzziness, known as hard clustering approach. Hence, one can confirm that such kind of clustering algorithm is to be selected when the underlying structure of $\mathcal{G}$ is more categorical, as in the case of a Dirac distribution.

2. In the second application, a soft clustering approach is applied. To do this, we have chosen the fuzzy clustering algorithm called "fanny" which was introduced in Chapter four of Kaufman and Rousseeuw (1990). Here, for each observation, instead of considering only one clustering label, probabilities of different clustering labels is computed. Now, under the same hypothesis of the first application, Kolmogorov-Smirnov statistics has a value of $\mathrm{KS}_n \simeq 1$ with a $p-$value of 0% for both Gaussian and Dirac cases. Then, for the Gaussian underlying distribution, the considered soft approach has no additional value compared to the k-means and the algorithm fails again in identifying the labeling difficulty. However, for the Dirac case the soft approach, unlike the k-means, is not able to identify well original labels, which is logical due to the fact that in Dirac case the mass function is concentrated at only one point. Finally, the comparisons of the *cdf* of both $\mathcal{H}_I(\mathcal{G}_n)$ and $\mathcal{H}_{J_n}(\mathcal{G}_n)$ in Figure 5 shows that, as a result of its high fuzziness, the chosen clustering is under-confident. In fact, the values of $\mathcal{H}_I(\cdot)$ are almost concentrated in the interval $[0, 0.15]$ while those of $\mathcal{H}_{J_n}(\cdot)$ are spread on $[0.15, 1.4]$. In addition, Table 1 shows that, under the Gaussian assumption, the distribution of $\mathcal{H}_{J_n}(\cdot)$ is more volatile with significantly higher mean comparing to $\mathcal{H}_I(\cdot)$ which is another evidence of the high level of uncertainty of the considered soft clustering algorithm.



(a) 1D, Fanny algorithm

(b) 2D, Fanny algorithm

Figure 5: *cdf* of different entropies (under fanny algorithm) in the case of a Gaussian mixture distribution ($n = 10\,000$ observations). On the left: one dimensional case of Case A, on the right: two dimensional case of Case C. High abscissas values correspond to points with high labeling difficulty, or equivalently high uncertainty. The *cdf* of $\mathcal{H}_I(\mathcal{G}_n)$ are blue upper curves, *cdf* of $\mathcal{H}_{J_n}(\mathcal{G}_n)$ are red lower curves.

	1-D Gaussian case		2-D Gaussian case	
	$\mathcal{H}_I(\mathcal{G}_n)$	$\mathcal{H}_{J_n}(\mathcal{G}_n)$	$\mathcal{H}_I(\mathcal{G}_n)$	$\mathcal{H}_{J_n}(\mathcal{G}_n)$
Mean	0.008	0.585	0.056	0.681
Variance	0.0014	0.1004	0.027	0.024
Minimum	0	0.144	$\simeq 0$	0.415
Maximum	0.993	1.534	$\simeq 1$	1

Table 1: Moments of entropy distributions for soft clustering fanny-algorithm ($n = 10\,000$ observations)

3.2 Comparison using Jensen-Shannon Divergence

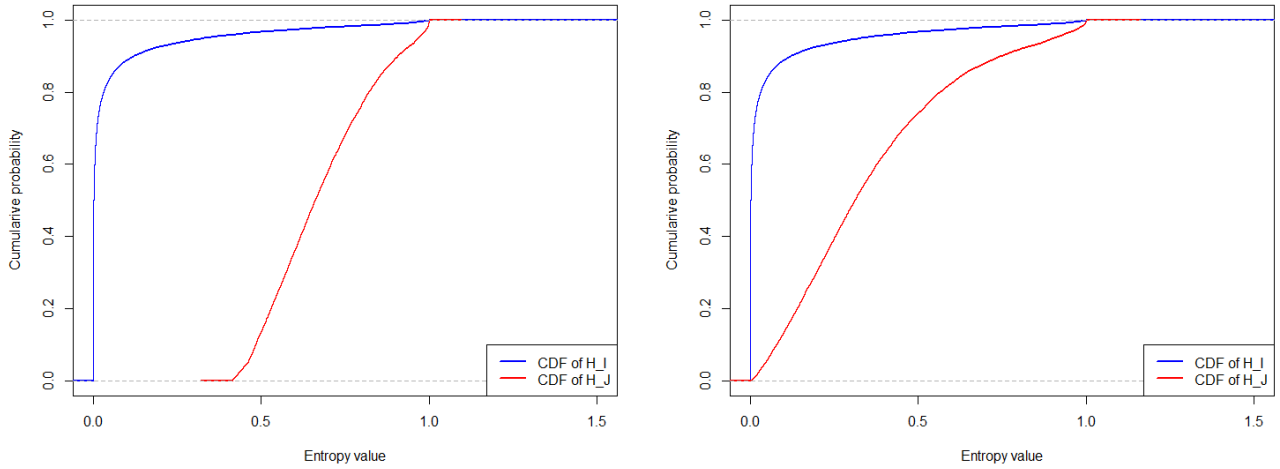
In this subsection, we aim at comparing an empirical entropy deriving from a fuzzy clustering algorithm (and associated parameters), with the known underlying entropy of the known mixture $\mathcal{G}$. To this aim, the distance between two entropy distributions P and Q is calculated based on Jensen-Shannon divergence (JSD) metric defined by:

$$\mathrm{JSD}\left(P \mid\mid Q\right) = \frac{1}{2}\left[\mathrm{KL}\left(P \mid\mid R\right) + \mathrm{KL}\left(Q \mid\mid R\right)\right],$$

where $R = \frac{1}{2}\left(P + Q\right)$ and $\mathrm{KL}\left(P \mid\mid R\right)$ denotes the Kullback-Leibler divergence between probability distributions P and R. Note that the JSD has the following properties: 1) Non negative measure; 2) Symmetric measure and 3) $\mathrm{JSD} \in [0, 1]$, with $\mathrm{JSD} = 0$ if and only if $P = Q$. Hence, this metric is suited to the case of fuzzy underlying distributions when soft clustering algorithm are applied.

We compare here some theoretical and empirical fuzziness level, using the Jensen-Shannon Divergence. To do this, we compute the following statistic:

$$\mathrm{JSD}_n := \mathrm{JSD}\left(\mathcal{H}_I\left(\mathcal{G}_n\right) \mid\mid \mathcal{H}_{J_n}\left(\mathcal{G}_n\right)\right). \tag{2}$$



(a) 2D, Fanny algorithm

(b) 2D, FKM algorithm

Figure 6: *cdf of different entropies in the case of a 2-D Gaussian mixture model, Case C, ($n = 10\,000$ observations). On the left fanny algorithm, on the right FKM algorithm.. The cdf of $\mathcal{H}_I(\mathcal{G}_n)$ are blue upper curves, cdf of $\mathcal{H}_{J_n}(\mathcal{G}_n)$ are red lower curves.*

By applying the JSD on our previous examples introduced in Section 2.1, we get the following results:

1. For the Gaussian mixture distribution and "fanny" clustering algorithm context with a sample size $n = 10\,000$: $\text{JSD}_n = 0.985$ for one dimensional Gaussian (Case A) and $\text{JSD}_n = 0.897$ for the two dimensional (Case C). Then, for this setting and for this criterion, the performance of "fanny" algorithm is better for the two dimensional Gaussian mixture distribution (in terms of fuzziness level). But it is still an under-confident approach (see Figure 5 and Table 1). Note that, in the numerical illustrations of this paper, in order to compute different JSD, random variables $\mathcal{H}_I(\mathcal{G}_n)$ and $\mathcal{H}_{J_n}(\mathcal{G}_n)$ were discretized with a constant step size of 0.001.

2. Now in order to compare the efficiency of two different soft clustering algorithm "fanny" and "Fuzzy k-means (FKM)" introduced by Bezdek (1981) we consider a sample $n = 10\,000$ of the two dimensional Gaussian mixture distribution (Case C). Under these assumptions we get $\text{JSD}_n^{\text{fanny}} = 0.897 > \text{JSD}_n^{\text{FKM}} = 0.658$. Then with this setting, the FKM algorithm is more suited to find similar fuzziness as the true underlying model. The result is illustrated in Figure 6.

To study the impact of the sample size n on the quality and the performance of different clustering algorithms, Table 2a shows JSD_n for different values of n in the case of one and two dimensional Gaussian underlying distribution and in the context of "fanny" clustering algorithm. On the other hand Table 2b shows a comparison between $\text{JSD}_n^{\text{fanny}}$ and $\text{JSD}_n^{\text{FKM}}$ for different values of n in the case of two dimensional Gaussian mixture distribution. In both cases, especially for 2-D Gaussian case, JSD is decreasing for high values of n. Then, one can say that the ability of a clustering algorithm to reconstruct original hidden labels is partly related to the uncertainty on the empirical distribution.

As not to be limited to the comparative aspect of our approach, its important to propose a method that helps setting a kind of threshold or boundary between accepted and rejected clustering method, in terms of labels reconstruction performance, for a given set of data.

n	1-D Gaussian case	2-D Gaussian case
100	1.000	**1.000**
500	0.991	**0.967**
1000	0.997	**0.959**
3000	0.990	**0.921**
5000	0.990	**0.920**

(a) Performance of "fanny" algorithm for different values of n

n	fanny	FKM
100	1.000	**0.960**
500	0.967	**0.846**
1000	0.959	**0.817**
3000	0.921	**0.707**
5000	0.920	**0.664**

(b) Comparing the performance of "fanny" and FKM algorithms for different values of n, Case C

Table 2: JSD for different values of n, the lower the better.

3.3 Statistical test

Using the results of Section 2, we can make some progress on the problem of selecting a good clustering method that fits well the original mixture distribution assumption, when it is known. This way, one can understand if a clustering method tends to be over or underconfident, compared to a given underlying model. We will investigate the case of unknown mixture distribution in Section 5.

Recall that $\mathcal{G}_n$ is a *rv* having the same empirical distribution as an *iid* sample of $\mathcal{G}$, and that $\mathcal{H}_{J_n}(\mathcal{G}_n)$ is the empirical entropy of the clustering method under investigation, as defined in Section 2.2, for the corresponding sample.

We make the assumption that the clustering method under consideration has a specific *consistency* property: more precisely, we assume that there exist joint random variables $(\mathcal{G}, J)$ such that $\mathcal{H}_{J_n}(\mathcal{G}_n) \xrightarrow{d} \mathcal{H}_J(\mathcal{G})$ in distribution, as $n \to \infty$. In other words, the algorithm's empirical fuzziness distribution should converge to some limit distribution, which occurs especially if the clustering weights are not too altered as the sample grows. As an illustration of this assumption, Figure 7 shows that for both fanny and FKM algorithms, and after a reasonable number of observations (e.g. $n = 500$) the *cdf*s of entropies $\mathcal{H}_{J_n}(\mathcal{G}_n)$ seem to empirically converge to some limit distribution.

We consider the context of a statistical test with H_0 : "the entropy of clustering method complies with the original mixture distribution assumption" vs H_1 : "the entropy of the clustering method does not comply with the mixture distribution assumption". In other words one can write:

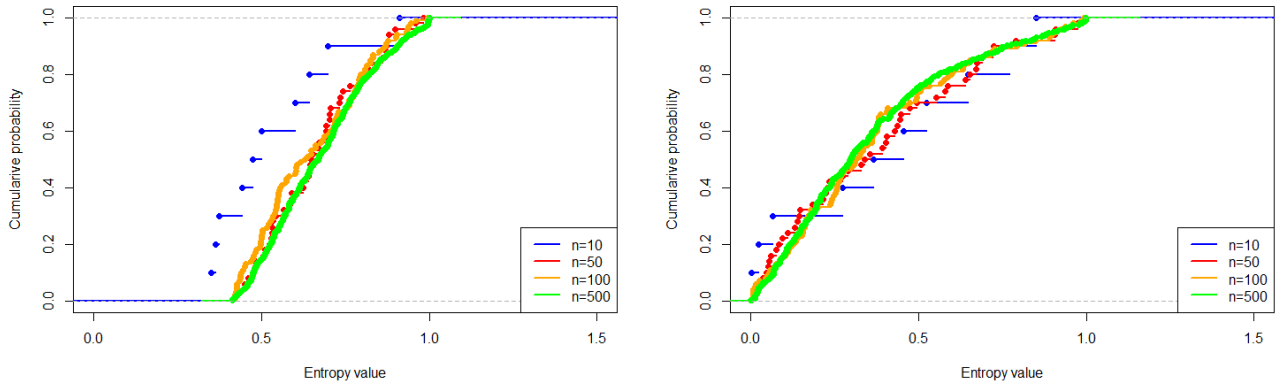
$$H_0 : \ \mathcal{H}_J(\mathcal{G}) \stackrel{d}{=} \mathcal{H}_I(\mathcal{G}) .$$

The considered test compare here a clustering method with an underlying given theoretical mixture distribution, but it would be quite straightforward to adapt it to the comparison of two different clustering algorithms.

Assume observations are an *iid* sequence of n random variables distributed as $\mathcal{G}$, say $G_1, \ldots, G_n$. For this data and for a given clustering algorithm, the deterministic function $\mathcal{H}_{J_n}(.)$ is returned at each of these data points. Under the above mentioned *consistency* assumption, we assume for n large enough that the sequence $\mathcal{H}_{J_n}(G_1), \ldots, \mathcal{H}_{J_n}(G_n)$ is an *iid* sample of $\mathcal{H}_J(\mathcal{G})$.

Now assume that the underlying joint distribution of $(\mathcal{G}, I)$ is known. Given $\mathcal{G} = x$, the value of $\mathcal{H}_I(x)$ is known. For an *iid* sequence of m random variables $G'_1, \ldots, G'_m$ distributed as $\mathcal{G}$, independent from the previous sequence, the sequence $\mathcal{H}_I(G'_1), \ldots, \mathcal{H}_I(G'_m)$ is an *iid* sample of $\mathcal{H}_I(\mathcal{G})$.

Finally, we get two independent *iid* samples of the respective real-valued random variables $\mathcal{H}_I(\mathcal{G})$ and $\mathcal{H}_J(\mathcal{G})$, with respective sample sizes m and n. Under H_0 the two samples derive from the same distribution. Hence any two-sample test of equality of distribution can be applied, like the two-sample Kolmogorov–Smirnov test. This can also be applied if $\mathcal{G}$ is discrete or derived from some particular data.



(a) 2D, Fanny algorithm

(b) 2D, FKM algorithm

Figure 7: *cdf of entropies $\mathcal{H}_{J_n}(\mathcal{G}_n)$ for different values of n in the case of a 2-D Gaussian mixture model, Case C. On the left fanny algorithm, on the right FKM algorithm.*

More specifically, the test statistic can be written

$$T_{m,n} := D\left(\mathcal{H}_I\left(\mathcal{G}'_m\right) \,||\, \mathcal{H}_{J_n}\left(\mathcal{G}_n\right)\right) \tag{3}$$

where D is the chosen dissimilarity between two distributions of real-valued random variables, for example, the Kolmogorov-Smirnov distance, or the JSD divergence, and where $\mathcal{G}'_m$ and $\mathcal{G}_n$ are two independent rv distributed as the respective samples $\{G'_1, \ldots, G'_m\}$ and $\{G_1, \ldots, G_n\}$.

When the distribution of the test statistic is unknown, the problem of obtaining an approximate critical region of the considered test, under H_0, can be solved by applying a numerical simulation method.

Under the assumption H_0, the distribution of $T_{m,n}$ can be estimated by simulation: two independent *iid* samples of m and n observations respectively are generated according to the joint original distribution $(\mathcal{G}, I)$. The distance $T^{H_0}_{m,n}$ between corresponding empirical distributions of $\mathcal{H}_I\left(\mathcal{G}'_m\right)$ and $\mathcal{H}_I\left(\mathcal{G}_n\right)$ is calculated. By repeating this procedure a large number of times, one obtains under H_0 the approximate quantiles of the distribution of $T^{H_0}_{m,n}$, from which one can get a critical region of approximated level α.

Finally, on a sample of n observations, we choose here $m = n$ and a clustering method is accepted if $T_{n,n} < (1-\alpha)^{th}$ quantile of the obtained distance distribution $T^{H_0}_{n,n}$ under H_0.

As an application, using JSD metric, we consider testing H_0 vs H_1 where the underlying distribution is the two dimensional Gaussian mixture defined in Section 2.1. Based on the numerical simulation method described above, with $m = n = 10\,000$, and by repeating the procedure 1000 times, we obtain the approximate quantiles of the corresponding JSD distribution and the critical region of approximated level $\alpha = 5\%$, which is $q^{JSD}_{0.95} = 0.04$. Then for a given clustering algorithm if $T_{n,n} > q^{JSD}_{0.95}$ we reject H_0. Hence the clustering method fails to understand the labeling difficulty of the considered observation. In our case, both fanny and FKM are rejected as both $T^{\text{fanny}}_{n,n}$ and $T^{\text{FKM}}_{n,n}$ are greater than 0.04.

It is important to note that the proposed test only evaluates the compatibility of a clustering method with an underlying probabilistic model in terms of fuzziness. A test rejection does not imply that the clustering method is ineffective; rather, it indicates that it must be modified if it is to be compatible with a given probabilistic model. This is the goal of the following section.

	Training set			Testing set	
	$\text{JSD}(\theta = 1)$	$\hat{\theta}$	$\text{JSD}\left(\hat{\theta}\right)$	$\text{JSD}(\theta = 1)$	$\text{JSD}\left(\hat{\theta}\right)$
Fanny	0.895	5.958	**0.043**	0.918	**0.088**
FKM	0.630	3.348	**0.039**	0.678	**0.084**

Table 3: Comparison of JSD: original vs corrected on testing and training data, the lower the better.

4 Correction of clustering probabilities

In this section, we assume that the true model behind observations is known, i.e. that the joint law of $(\mathcal{G}, I)$ is known. We will discuss the case of unknown model in the further Section 5.

Assume that the statistical test of Section 3.3 is rejecting the clustering method. Before proposing another clustering algorithm, a possible solution is to adjust the underlying cluster probabilities using a parametric transformation.

Recall that $p_j(x_\ell)$ is the probability of assigning observation x_ℓ to cluster j. In order to adjust these probabilities to fit the original mixture distribution we propose the following transformation:

$$p_j(x_\ell)^* = \frac{p_j(x_\ell)^\theta}{\sum_{i \in \mathcal{J}} p_{C_i}(x_\ell)^\theta}, \tag{4}$$

where $\mathcal{J}$ is the proposed set of cluster indices and $\theta \in \mathbb{R}^+$ a parameter, to be estimated, reflecting the reason behind the bad clustering quality.

Then, based on these new probabilities a new clustering entropy is calculated. We denote it by $\mathcal{H}_{J_n}(\mathcal{G}, \theta)$. Hence, one may assess the effect of this transformation by recomputing $\text{JSD}\left(\mathcal{H}_I(\mathcal{G}_n) \mid\mid \mathcal{H}_{J_n}(\mathcal{G}_n, \theta)\right)$.

Now, the optimal value of the parameter θ is the one minimizing the previous JSD, which will be given by:

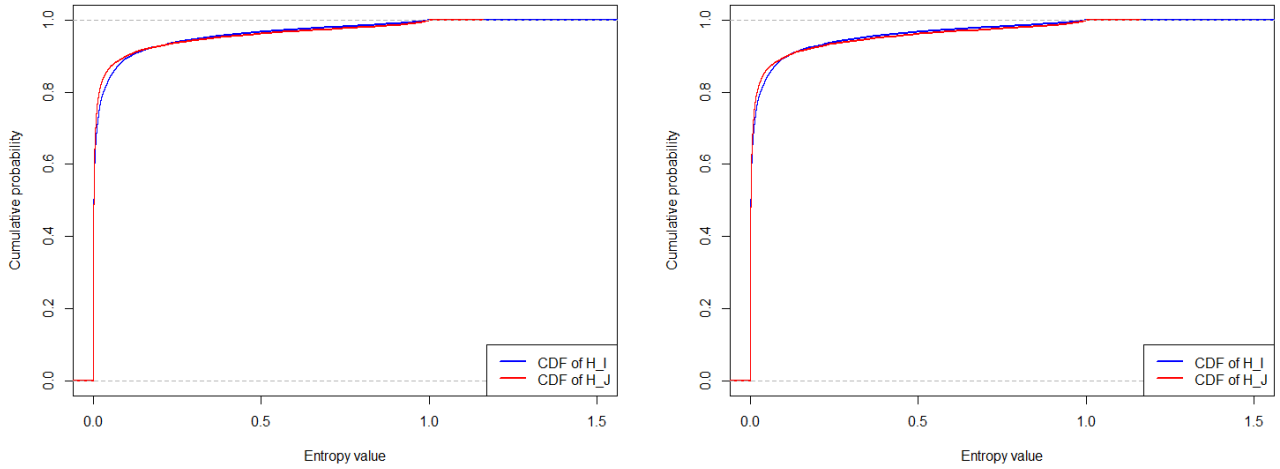
$$\hat{\theta} = \arg\min_\theta \text{JSD}\left(\mathcal{H}_I(\mathcal{G}_n) \mid\mid \mathcal{H}_{J_n}(\mathcal{G}_n, \theta)\right).$$

So, in case $\hat{\theta} > 1$, one can say that the proposed clustering algorithm is not categorical enough and has a higher fuzziness level then the one of the original mixture distribution. Otherwise, if $\hat{\theta} < 1$, the current clustering algorithm is too categorical and we need to regularize it by injecting some source of fuzziness. Due to this interpretation, the parameter $\hat{\theta}$ can be considered as an indicator of fuzziness.

To study the efficiency of this approach, we apply it on a previous example considering two different soft clustering algorithm "fanny" and "Fuzzy k-means (FKM)". We generate a sample $n = 10\,000$ of the two dimensional Gaussian mixture distribution with the same parameters as the one in Section 2.1. Based on this training sample we calculate the estimator $\hat{\theta}$ of θ. Then, we use the value of this estimator to correct the probabilities proposed by each of the clustering algorithm on a testing sample of 3000 observations and we compare the impact of this correction on the quality of the clustering on both training and testing sets by calculating the corresponding JSD. Under these assumptions we get results summarized in Table 3.

Then, by applying the parametric transformation proposed in Equation (4) we are able to reduce enormously the JSD and correcting the behavior of both algorithms, so that they become more suited to find the hidden labels of the original data. Even on the testing set, the correction has a considerable favorable influence on both algorithms' clustering quality. In addition, comparison of the *cdf* of different entropies $\mathcal{H}_I(\mathcal{G}_n)$ vs $\mathcal{H}_{J_n}\left(\mathcal{G}_n, \hat{\theta}\right)$ are illustrated in Figure 8 for training set and Figure 9 for testing.

Note that a more complex transformation than the one proposed in (4) can be done using more than one parameter and focusing on local parts of the distribution. By adapting the same approach described in this section we can select the best transformation according to, say, the least JSD criterion. Also, as the application is done on data with gaussian underlying distributions, it is obvious to expect that the best performance, in terms of clustering, will be for the Gaussian Mixture Model (GMM) based clustering method, introduced by McLachlan and Basford (1988), where gaussian distributions



(a) 2D, Fanny algorithm

(b) 2D, FKM algorithm

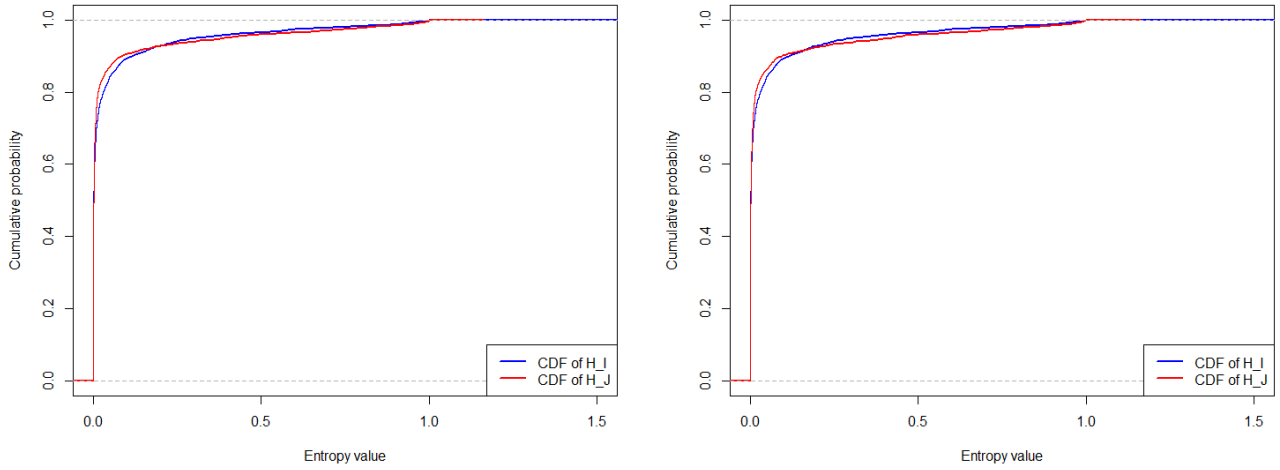
Figure 8: *cdf of different entropies in the case of a 2-D Gaussian mixture model ($n = 10\,000$ observations, Case C). On the left fanny algorithm ($\hat{\theta} = 5.958$), on the right FKM algorithm ($\hat{\theta} = 3.348$), with modified probabilities applied on training data*

are considered as priors. As expected, even before correction, JSD of the GMM method is about 0.042 for training and 0.089 for testing with a good fit of the *cdf*s of the two entropies (see Figure 10).

Furthermore, after applying the proposed probability transformation, we used the theoretical labels of the 2-D Gaussian mixture distribution generated in Subsection 2.1 to examine the ability of fuzzy clustering algorithms in reconstructing the original clusters of an underlying data. Then, the idea is to make a confusion matrix comparing theoretical labels with the labels proposed by the clustering method. The accuracy of each clustering method is calculated as the proportion of well classified observation and formulated by:

$$Accuracy = \frac{1}{n}\sum_{\ell=1}^{n}\sum_{j\in\mathcal{J}}\mathbb{P}\left[\text{assign observation } \ell \text{ to cluster } j/\text{Theoretically observation } \ell \text{ is in cluster } j\right]. \qquad (5)$$

Tables 4 and 5 show the clustering accuracy as defined in Equation (5) for several soft clustering method applied on 2-D Gaussian mixture distribution generated data, before and after applying the correction of clustering probabilities on training and testing sets respectively. Note that here also θ is estimated on the training set and used to correct clustering probabilities on the testing set. In general, the accuracy of the clustering is improving in a significant way after applying the correction method, except for the GMM clustering method where the accuracy is almost the same as this method is designed to deal with gaussian mixture underlying distributions which is fitting perfectly the generated data. In addition, the estimated value of θ is giving information about the quality of the clustering method. For Fanny and FKM algorithms the fuzziness index $\hat{\theta}$ is far greater than one, so these two methods are not categorical enough and have higher fuzziness level than the original mixture. However, $\hat{\theta}$ for GMM is close to one, which is reasonable because GMM is designed in a way to generate original clusters of a gaussian mixture distribution. Also, an added value of the proposed correction method, is that after performing the correction one can revisit the result of the statistical test that was introduced in Section 3.3 to reassess if after modification, a certain clustering method that was rejected previously is



(a) 2D, Fanny algorithm

(b) 2D, FKM algorithm

Figure 9: *cdf of different entropies in the case of a 2-D Gaussian mixture model ($n = 3\,000$ observations, Case C). On the left fanny algorithm ($\hat{\theta} = 5.958$), on the right FKM algorithm ($\hat{\theta} = 3.348$), with modified probabilities applied on testing data*

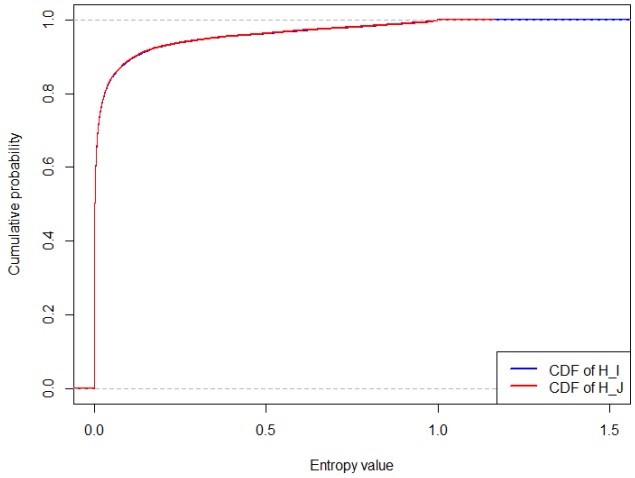


Figure 10: *cdf* of different entropies in the case of a 2-D Gaussian mixture model ($n = 10\,000$ observations, Case C) for GMM algorithm ($\hat{\theta} = 0.970$) applied on training data without modified probabilities

Clustering method	Accuracy before correction	Accuracy after correction	$\hat{\theta}$
GMM	0.968	**0.980**	0.970
Fanny	0.802	**0.968**	5.958
FKM	0.904	**0.969**	3.348

Table 4: Accuracy of different clustering methods calculated on training data before and after correction, the higher the better. θ is estimated based on the training set and used to correct clustering probabilities on both training and testing sets.

Clustering method	Accuracy before correction	Accuracy after correction	$\hat{\theta}$
GMM	**0.978**	**0.97**	0.970
Fanny	0.798	**0.967**	5.958
FKM	0.901	**0.967**	3.348

Table 5: Accuracy of different clustering methods calculated on testing data before and after correction, the higher the better. θ is estimated based on the training set and used to correct clustering probabilities on both training and testing sets.

now accepted. Conclusions of the statistical test applied on several soft clustering methods before and after probability correction are presented in Tables 6 and 7 for training and testing sets respectively. It is clear that after applying the parametric correction of the probabilities, the values of the statistic of the test of Section 3.3 decrease significantly to become much closer to the critical region empirical quantile $\left(q_{0.95}^{JSD} = 0.04\right)$. In fact some of the clustering algorithms that was rejected before correction are accepted after it. Then, correcting clustering probabilities makes: 1) different clustering method more accurate in reconstructing original clusters and 2) the comparison between different clustering method more reasonable in addition to the selection of the best performing one.

	Value of the JSD statistic		Decision about H_0	
Clustering method	Before correction	After correction	Before correction	After correction
GMM	0.042	**0.035**	Accepted	Accepted
Fanny	0.895	**0.043**	Rejected	Accepted
FKM	0.630	**0.039**	Rejected	Accepted

Table 6: Statistical test conclusions on training data. JSD is calculated on training data before and after correction, the lower the better. θ is estimated based on the training set and used to correct clustering probabilities on both training and testing sets.

	Value of the JSD statistic		Decision about H_0	
Clustering method	Before correction	After correction	Before correction	After correction
GMM	0.089	**0.086**	Rejected	Rejected
Fanny	0.918	**0.088**	Rejected	Rejected
FKM	0.678	**0.084**	Rejected	Rejected

Table 7: Statistical test conclusions on testing data. JSD is calculated on testing data before and after correction, the lower the better. θ is estimated based on the training set and used to correct clustering probabilities on both training and testing sets.

5 Application on real data

Up to this point, we have assumed that the hidden mixture model and the $(\mathcal{G}, I)$ distribution were known. Of course, this is not always the case in the real world. Indeed, once the model is understood, no clustering technique is needed.

The goal of this section is to demonstrate that the presented methods are still applicable even when the true mixture model is hidden, that they can aid in comparing different algorithms, that they can improve prediction accuracy on test samples for both labeled and unlabeled data, and that they can aid in selecting the appropriate number of clusters. For each of these purposes, specific examples are provided.

5.1 Supervised learning, improving classification accuracy

We consider here a supervised classification task. We show that the proposed fuzziness correction of Section 4, together with a standard estimation of some bandwidth parameter, improve the clustering accuracy of most fuzzy clustering algorithms, on an unlabeled test sample. To do so, we look into possible estimation procedures for comparing clustering fuzziness to an estimated probabilistic model.

As an application of our approach, we use an open source data from the UCI Machine Learning Repository, Dua and Graff (2017), available at `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29`.

This data classifies $n = 569$ individuals in either malignant (M) or benign (B) breast cancer using 30 numeric features characterizing each person. I.e., a space of dimension $\mathbb{R}^{569\times 30}$, labeled into $\mathcal{I} = \{1 = \text{"M"}, 2 = \text{"B"}\}$ cluster indexes. Therefore, assuming that we are in the context of a mixture distribution $\mathcal{G}$, the probability that a given observation x is sampled from the underlying rv X_i can be estimated for any $i \in \mathcal{I}$, by for example:

$$\widehat{p_i}\,(x) = \frac{\widehat{\alpha}_i \widehat{f^i}\,(x)}{\sum_{j\in\mathcal{I}} \widehat{\alpha}_j \widehat{f^j}\,(x)}, \tag{6}$$

with, $\widehat{f^i}\,(\cdot)$ the kernel density estimation (KDE) of the observations generated by the underlying rv X_i (i.e., observations in the cluster of index i). In other words it is an estimation of the pdf of the rv X_i. Note that, practically one can use kernel density estimation (KDE) functions, already implemented in statistical software, to compute these values. The quantity $\widehat{\alpha}_i$ is the proportion of observations in the same cluster, i.e.,

$$\widehat{\alpha}_i = \frac{\text{Number of observations in cluster } i}{n}.$$

In this application, we used a standard KDE of the R software, provided by the KDE function, corresponding here to a plug-in bandwidth as defined in Chacón and Duong (2010). Results would naturally differ for other bandwidths, but we show here that this classical KDE helps improving the clustering

	Training set		
	JSD($\theta = 1$)	$\hat{\theta}$	JSD$\left(\hat{\theta}\right)$
Fanny	**1**	0.001	**1**
FKM	0.989	9.883	**0.249**
GMM	0.0834	1.652	**0.0556**

Table 8: Comparison of JSD: original vs corrected on training data. θ is estimated on the training set and used to compute JSD (the lower the better) on both training and testing sets.

	Testing set		
	JSD($\theta = 1$)	$\hat{\theta}$	JSD$\left(\hat{\theta}\right)$
Fanny	**1**	0.001	**1**
FKM	1	9.883	**0.352**
GMM	0.0709	1.652	**0.0827**

Table 9: Comparison of JSD: original vs corrected on testing data. θ is estimated on the training set and used to compute JSD (the lower the better) on both training and testing sets.

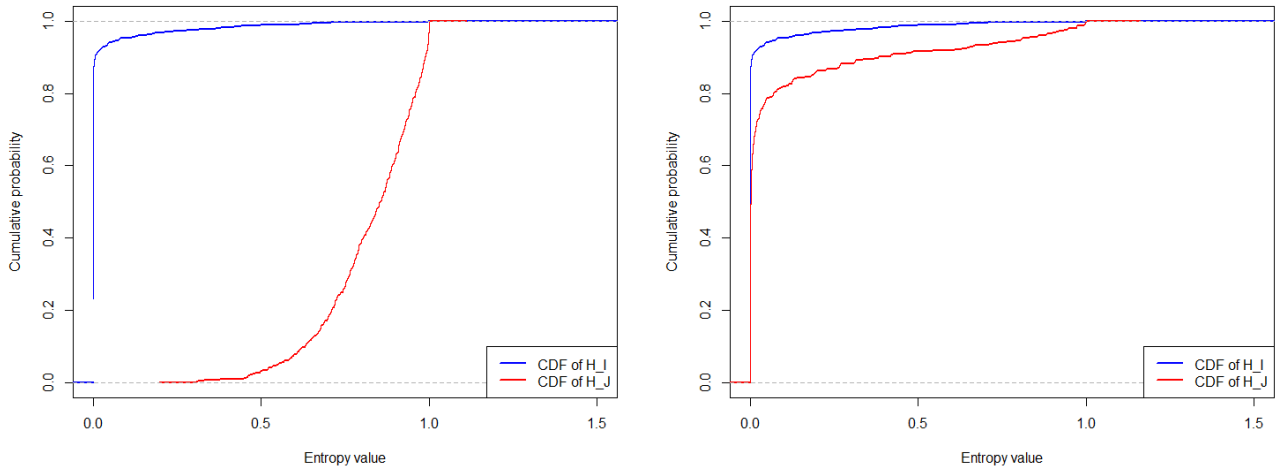
accuracy on a test sample.

For the three different soft clustering methods introduced previously, we compare the impact of the parametric correction proposed in Equation (4) on the quality of the clustering by first estimating the optimal value of θ, on a training set representing 80% of the considered data set, and then calculating the corresponding JSD on both training and testing sets. Results are in Tables 8 and 9.

Considering clustering fuzziness, Fanny is the worst clustering method for the considered data, even after the correction it is affecting the same probability for both clusters (1 and 2) without any discrimination power. On the other hand, the performance of FKM and GMM is improving enormously after the correction with a preference to the GMM method. The significant impact of the parametric correction on FKM and GMM can be seen also when comparing the *cdf* of the entropies before and after correction in Figures 11 and 12. Note that, the impact of the correction for the GMM is limited because even before correction GMM is performing well on reconstructing original labels. In addition, one can remark that the estimated value of θ for the FKM is far higher than one which is equivalent to say that FKM has a higher fuzziness level than the original mixture of the considered cancer data set. However, the fuzziness index $\hat{\theta}$ for GMM is close to one, which is reasonable because GMM is already performing well on discrimination level even before the correction.

Finally, the illustrations show that the correction of clustering weights, together with the use of a standard bandwidth parameter, significantly improve the clustering accuracy on a test sample. The definition of the fuzziness level, at the origin of the proposed correction, appears here to be of practical interest, even when the hidden mixture model is unknown.

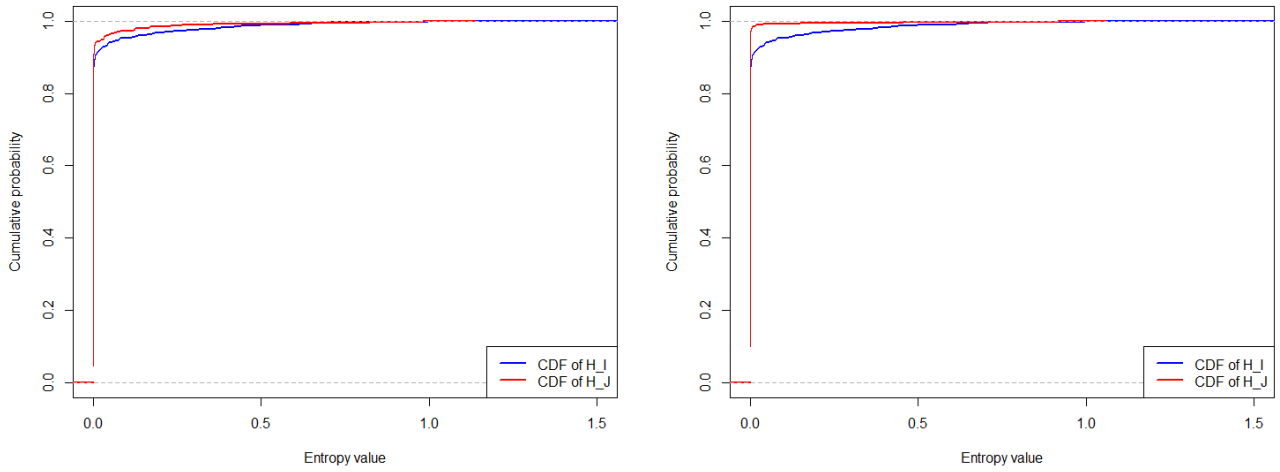
Now, by focusing only on FKM and GMM we can also compute the accuracy of each clustering method before and after the parametric correction based on the approach described in Equation (5). Here also $\hat{\theta}$ is estimated on the training set and used to compute the clustering accuracy for both training and testing sets. Results are in Tables 10 and 11. On both training and testing datasets one may remark that the clustering accuracy of the FKM method is improving significantly after the correction of the corresponding probabilities. However, the correction has no impact on the accuracy of the GMM. This is also reflected by the value of $\hat{\theta}$ which is close to one, indicating that the fuzziness level of the GMM is acceptable and is capturing well the original labels of the clusters. This may be explained by the possibility that the underlying distribution of the considered dataset is following a gaussian mixture distribution.



(a) real data, FKM without correction

(b) real data, FKM with correction

Figure 11: *cdf of different entropies for FKM algorithms. On the left without modified probabilities, on the right with modified probabilities, applied on cancer data*



(a) real data, GMM without correction

(b) real data, GMM with correction

Figure 12: *cdf of different entropies for GMM algorithms. On the left without modified probabilities, on the right with modified probabilities, applied on cancer data*

Clustering method	Accuracy before correction	Accuracy after correction	$\hat{\theta}$
FKM	0.7	**0.91**	9.883
GMM	**0.937**	**0.937**	1.652

Table 10: Accuracy of different clustering methods based on training data, the higher the better. θ is estimated on the training set and used to compute JSD on both training and testing sets.

Clustering method	Accuracy before correction	Accuracy after correction	$\hat{\theta}$
FKM	0.683	**0.89**	9.883
GMM	**0.957**	**0.95**5	1.652

Table 11: Accuracy of different clustering methods based on testing data, the higher the better. θ is estimated on the training set and used to compute JSD on both training and testing sets.

This simple illustrative example could be extended in several ways, that are let as future work.

Firstly, we considered here a supervised learning problem. For unsupervised learning, one option is to perform hard clustering of the data, obtain labels for each point, and then estimate the entropy as in a supervised setting. Repeating the procedure for several hard clusterings would account for both the uncertainty of the initial hard clustering and the uncertainty caused by the intrinsic overlapping of cluster densities for a given labelling and bandwidth.

Secondly, this illustration was heavily depending on the choice of bandwidth parameter. Intuitively, a larger bandwidth results in more fuzziness, hence impacting the correction parameter. Here, a standard bandwidth estimator helped improving the accuracy of the clustering algorithm, but the choice of other bandwidths would be possible.

5.2 Unsupervised learning, finding clusters number

In the context of selecting the optimal number of clusters in unsupervised analysis, several state-of-the-art methods exist. These include the Elbow Method, which identifies the point of maximum curvature in the within-cluster sum of squares plot (Thorndike, 1953); the Silhouette Coefficient, which measures the compactness and separation of clusters (Rousseeuw, 1987); the Gap Statistic (Tibshirani et al., 2001), which compares within-cluster dispersion to that of reference data; methods using the Bayesian Information Criterion (BIC), which balances model fit and parameter complexity (Zhao et al., 2008); and more other approaches like the Davies-Bouldin Index for cluster validation (Davies and Bouldin, 1979). Notable recent references on these methods include studies by Milligan and Cooper (1985) on cluster validity indices, Rousseeuw (1987) on silhouettes, Tibshirani et al. (2001) on data-driven approaches, Sugar and James (2003) for an information theoretic approach, and Halkidi and Koutsopoulos (2019) for advancements in clustering techniques and validation methods. Software packages are also available, see e.g. Charrad et al. (2014).

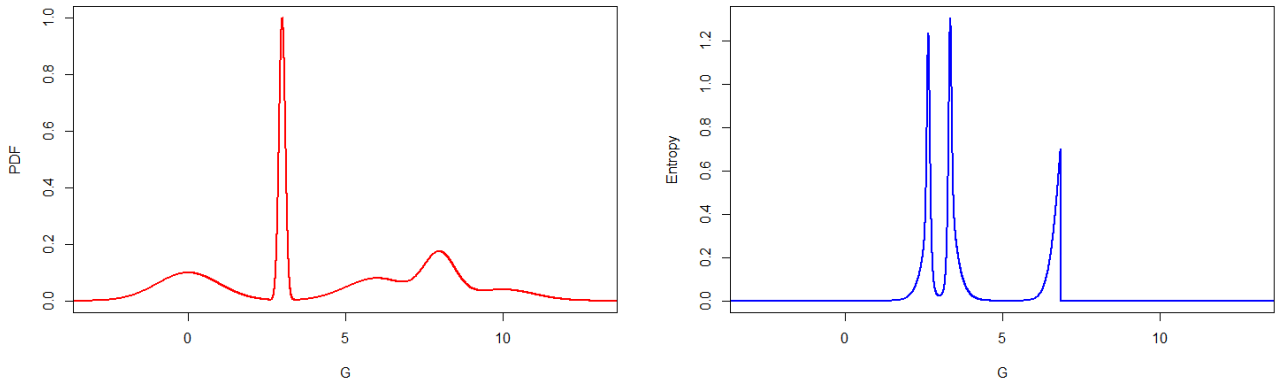
Our work proposes a novel approach that considers the fuzziness level generated by each cluster method, incorporating the degree of uncertainty into the evaluation of the optimal number of clusters for a more comprehensive analysis. In this application, we show that measuring the fuzziness level of a clustering algorithm may help choosing the right number of clusters.

Intuitively, the fuzziness level is related to the right number of clusters: any segmentation of a true cluster in two parts tends to create some fuzziness areas, so that the proposed number of clusters cannot be too high. Proposing a smaller number of clusters may be reasonable when it leads to fuse true clusters, but it is clearly inappropriate when it leads to cut true cluster in parts, creating fuzziness. Based on this idea, we will use the distribution of the empirical entropy to choose the right number of clusters.

For the sake of clarity, let us consider first a toy example. Let $\mathcal{G}$ be a Gaussian mixture distribution in one dimension with a *pdf*:

$$f_{\mathcal{G}}(x) = 0.25 f_{\mathcal{N}(0,1)}(x) + 0.25 f_{\mathcal{N}(3,0.1)}(x) + 0.2 f_{\mathcal{N}(6,1)}(x) + 0.2 f_{\mathcal{N}(8,0.5)}(x) + 0.1 f_{\mathcal{N}(10,1)}(x).$$

where $f_{\mathcal{N}(.,.)}$ denotes the pdf of a Gaussian *rv* with indicated parameters. Associated labels are lost, so that the setting is similar to an unsupervised clustering algorithm.



(a) *pdf* of the mixture distribution $\mathcal{G}$

(b) entropy $\mathcal{H}_I(\mathcal{G})$

Figure 13: *Gaussian mixture distribution. Left: pdf of the mixture distribution* $\mathcal{G}$*, right: entropy* $\mathcal{H}_I(\mathcal{G})$ *representing the variation of the labeling difficulty.*

The left side of Figure 13 represents the *pdf* of $\mathcal{G}$, while the right side illustrates the function $\mathcal{H}_I$. Remark that this is a difficult example: even when the *pdf* is known, the number of clusters is here difficult to find, the truth is five clusters, but the values of entropy indicate the presence of a gray zone, where the level of fuzziness reaches its peak around the center of the given mixture distribution.

The right side of Figure 13 represents the entropy $\mathcal{H}_I(\mathcal{G})$. Once again, one can remark that the labeling is perfectly accurate when the values x of $\mathcal{G}$ are far from the central area of the distribution (i.e. left and right queues of the mixture distribution). However, the difficulty of cluster labeling is higher for central values where the region is fuzzy in terms of distribution selection.

As the maximal value of the entropy depends on the cluster number, we use here a rescaled versions of the empirical entropy, so that all rescaled entropies are belonging to $[0, 1]$:

$$\widetilde{\mathcal{H}}_{J_n}(x) = \frac{1}{\log_2(\mathrm{card}(\mathcal{J}))}\mathcal{H}_{J_n}(x)$$

Now, for FKM algorithm, we draw the distribution of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$, with $n = 25000$, for different numbers of clusters, varying from 2 to 8. Table 12 shows that by considering the number of clusters that minimizes the mean and the $75th$ percentile the preference goes to five clusters which fits with the simulated underlying mixture distribution. In other words, one selects the number of clusters with the lowest level of uncertainty for more than 75% of the available data. On the other hand, based on Figure 14 the entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ show that with 5 clusters, the entropy tends to be smaller, which is another hint on the optimal number of clusters.
Now, beyond considering the entropy mean as a primary indicator, it is valuable to delve into additional insights provided by the $95th$ percentile which is an extreme percentile. This extreme percentile can shed light on situations where a higher number of clusters may be justified.
In conclusion, while the mean of the entropy typically serves as a good indicator, examining non-extreme percentiles, e.g. $75th$, can provide useful information. However, taking into account extreme percentiles of the entropy can offer also valuable insights, particularly in situations where the complexity of the data warrants a more nuanced approach to cluster analysis. This multifaceted analysis enhances the decision-making process, allowing for a more informed determination of the optimal number of clusters in various cluster analysis applications.

The example here is chosen to be very clear to understand and visualize, but as the dimension d increases, the visual assessment of the right number of clusters is very difficult, whereas the considered

Number of clusters	Mean of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$75th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$
2	0.307	0.398	0.886
3	0.236	0.412	0.796
4	0.248	0.469	0.658
5	**0.201**	**0.365**	0.612
6	0.214	0.399	0.619
7	0.209	0.386	**0.587**
8	0.250	0.400	0.599

Table 12: Empirical mean, $75th$ percentile and $95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on fanny algorithm.

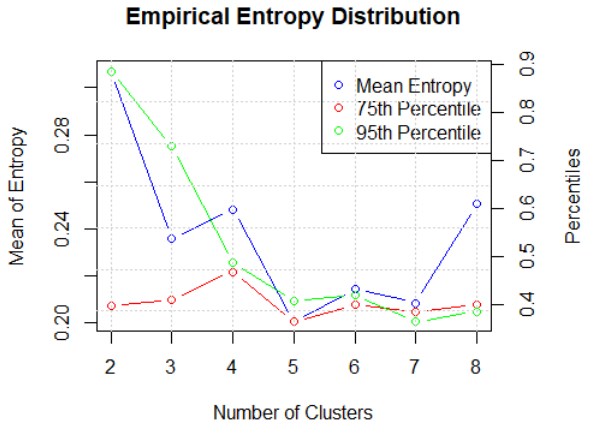


Figure 14: Mean, $75th$ and $95th$ percentile of entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on FKM algorithm and using simulated data

Number of clusters	Mean of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$75th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$
2	**0.823**	**0.937**	0.996
3	0.879	0.946	0.997
4	0.900	0.956	**0.991**
5	0.906	0.956	0.991
6	0.919	0.961	0.993

Table 13: Empirical mean, $75th$ percentile and $95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on FKM algorithm and applied on real data.

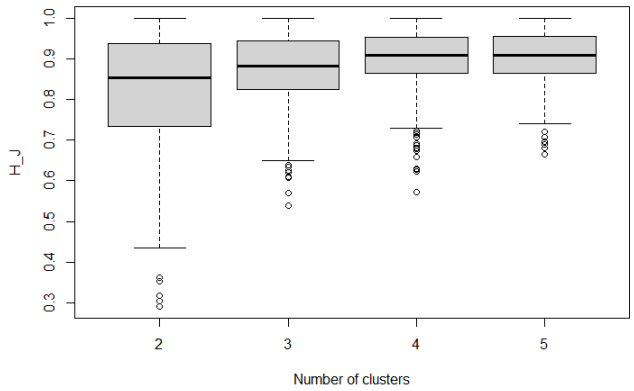


Figure 15: Boxplots of entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on FKM algorithm and using real data

entropies are still real-valued distributed.

Now, the same methodology is applied on the real dataset considered in Subsection 5.1. Based on FKM algorithm, we draw the distribution of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$, for different numbers of clusters, varying from 2 to 5. Table 13 shows that by considering the number of clusters that minimizes the empirical mean and the $75th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ we get $n = 2$, which is matching the data separated into two classes M and B.
On the other hand, based on Figure 15 the box-plots of entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ show that with 2 clusters, the entropy tends to be smaller, which is another hint on the optimal number of clusters. It is important to note that we are observing a similar pattern as in the previous example with simulated data, where the extreme percentile ($95th$ percentile) also highlights a preference for a higher number of clusters.

As a third application in the same unsupervised context, we propose to apply our method for selecting the optimal number of clusters to the MNIST dataset, aiming to demonstrate, once again, its effectiveness in extracting meaningful information within the data.
In fact, the MNIST dataset is a widely recognized benchmark in the field of machine learning and computer vision. It stands for Modified National Institute of Standards and Technology database and is a collection of handwritten digits, from 0 to 9, commonly used for training and evaluating various classification algorithms. The dataset consists of a total of 70,000 grayscale images, each measuring

Number of clusters	Mean of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$75th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$	$95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$
5	0.956	0.992	0.999
6	0.955	0.992	0.999
7	0.952	0.990	0.999
8	0.941	0.984	0.999
9	0.940	0.983	0.998
10	**0.939**	**0.982**	0.998
11	0.941	0.985	0.998
12	0.942	0.985	0.998
13	0.943	0.986	0.998
14	0.945	0.987	0.998
15	0.946	0.988	0.998

Table 14: Empirical mean, $75th$ percentile and $95th$ percentile of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on FKM algorithm and using MNIST data.

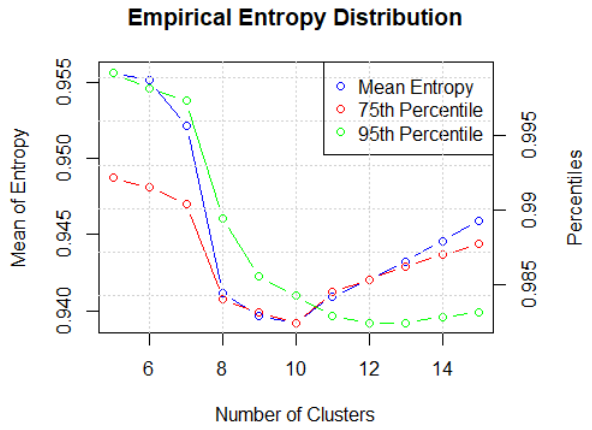


Figure 16: Mean, $75th$ percentile and $95th$ percentile of entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ for different number of clusters based on FKM algorithm and using MNIST data

28x28 pixels. These images are evenly divided into 60,000 training samples and 10,000 test samples. The MNIST dataset has become a standard reference for researchers and practitioners due to its simplicity, availability, and relevance to real-world applications.
Let us discuss the experimental results and outcomes obtained by applying the previous methodology to the MNIST dataset. We started by selecting a random sample of 25,000 observations from the training set of the MNIST dataset. Based on FKM algorithm, we draw the distribution of $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$, for different numbers of clusters, varying from 5 to 15. Once again, Table 14 shows that by considering the number of clusters that minimizes the mean and the $75th$ percentile the preference goes to ten clusters which is matching the data classified into ten digits. On the other hand, based on Figure 16 the entropies $\widetilde{\mathcal{H}}_{J_n}(\mathcal{G}_n)$ show that with 10 clusters, the entropy distribution tends to be skewed more to the left.

6 Conclusion

This paper introduced an innovative metric, based on Shannon's entropy for assessing the quality of clustering algorithms in terms of fuzziness level. The proposed metric can be used to compare the performance of two clustering algorithms in a way to conclude which one is more over/under confident. In addition, a statistical test has been constructed, based on the introduced fuzziness level metric. It helps to make a decision about accepting or rejecting a clustering algorithm. Moreover, a parametric adjustment of the underlying probabilities of a clustering algorithm has been introduced in order to improve the fuzziness level of the corresponding clustering method. According to many numerical simulations and real world data applications, it was noticed that the proposed methodology helps users significantly in getting better discrimination power from a given clustering method. Applications to find the optimal number of clusters were also proposed.
So far, a first perspective of this work is to try to develop a theoretical proof of the probability distribution of the proposed metric. A second perspective is to adapt the proposed methodology on data of higher level of complexity (i.e., data of mixed typology).

References

Aghababyan, A., Lewkow, N., and Baker, R. S. (2018). Enhancing the clustering of student performance using the variation in confidence. In *International Conference on Intelligent Tutoring Systems*, pages 274–279. Springer.

Bataineh, K. M., Naji, M., and Saqer, M. (2011). A comparison study between various fuzzy clustering algorithms. *Jordan Journal of Mechanical & Industrial Engineering*, 5(4).

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms: Advanced applications in pattern recognition*. Plenum Press (New York, NY [ua]).

Chacón, J. E. and Duong, T. (2010). Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *Test*, 19:375–398.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). Nbclust: an r package for determining the relevant number of clusters in a data set. *Journal of statistical software*, 61:1–36.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Dawyndt, P., Meyer, H. D., and Baets, B. D. (2005). The complete linkage clustering algorithm revisited. *Soft Computing*, 9(5):385–392.

De Oliveira, J. V. and Pedrycz, W. (2007). *Advances in fuzzy clustering and its applications*. John Wiley & Sons.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., and Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743.

Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys (CSUR)*, 28(1):77–80.

Halkidi, M. and Koutsopoulos, I. (2019). Qgraph: A quality assessment index for graph clustering. In Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., and Hiemstra, D., editors, *Advances in Information Retrieval*, pages 70–77, Cham. Springer International Publishing.

Jin, C. and Malthouse, E. C. (2016). On the bias and inconsistency of k-means clustering. preprint, doi:10.13140/RG.2.1.4300.5528.

Kaufman, L. and Rousseeuw, P. J. (1990). *An introduction to cluster analysis*. John Wiley and Sons, Incorporated.

Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(3):231–240.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York.

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019). Finite mixture models. *Annual Review of Statistics and Its Application*, 6(1):355–378.

Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179.

Park, S., Han, S., Kim, S., Kim, D., Park, S., Hong, S., and Cha, M. (2021). Improving unsupervised image clustering with robust learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12278–12287.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Roux, M. (2015). A comparative study of divisive hierarchical clustering algorithms. *arXiv preprint arXiv:1506.08977*.

Ruspini, E. H., Bezdek, J. C., and Keller, J. M. (2019). Fuzzy clustering: A historical perspective. *IEEE Computational Intelligence Magazine*, 14(1):45–55.

Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763.

Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika*, 18(4):267–276.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Yang, M.-S. (1993). A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11):1–16.

Yao, J., Dash, M., Tan, S., and Liu, H. (2000). Entropy-based fuzzy clustering and fuzzy modeling. *Fuzzy sets and Systems*, 113(3):381–388.

Zhao, Q., Hautamaki, V., and Fränti, P. (2008). Knee point detection in bic for detecting the number of clusters. In Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., and Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems*, pages 664–673, Berlin, Heidelberg. Springer Berlin Heidelberg.