Array 19 (2023) 100319

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

SSFuzzyART: A Semi-Supervised Fuzzy ART through seeding initialization and a clustered data generation algorithm to deeply study clustering solutions

Siwar Jendoubi *, Aurélien Baelde, Thong Tran

UPSKILLS R&D, 16 Rue Marc Sangnier 94600, Choisy-Le-Roi, France[1]

ARTICLE INFO

Keywords:
Semi-supervised learning
FuzzyART
Initialization by seeding
Clustering
SSFuzzyART
Clustered binary data generation

ABSTRACT

Semi-supervised clustering is a machine learning technique that was introduced to boost clustering performance when labeled data is available. Indeed, some labeled data are usually available in real use cases, and can be used to initialize the clustering process to guide it and to make it more efficient. Fuzzy ART is a clustering technique that is proved to be efficient in several real cases, but as an unsupervised algorithm, it cannot use available labeled data. This paper introduces a semi-supervised variant of the FuzzyART clustering algorithm (SSFuzzyART). The proposed solution uses the available labeled data to initialize clusters centers. In another hand, to deeply evaluate the characteristics of the proposed algorithm, a clustered binary data generation algorithm with controlled partitioning is also introduced in this paper. Indeed, the controlled generated clusters allows studying the characteristics of the proposed SSFuzzyART. Furthermore, a set of experiments is carried out on some available benchmarks. SSFuzzyART demonstrated better clustering prediction results than its classic counterpart.

1. Introduction

Clustering techniques are useful to group similar unlabeled data instances together. It is used to explore unlabeled datasets and to discover similarity relationships between its instances. In some real cases, it is possible to have access to labeled data instances. These instances are, usually, not enough to learn a classifier. In fact, the available labels could represent a subset of possible labels, on only a tiny fraction of the dataset, preventing the training of a learner. However, usual clustering algorithms are not designed to consider this information about labels. In such cases, semi-supervised techniques can use the labeled instances to guide the clustering process.

Fuzzy Adaptive Resonance Theory (Fuzzy ART) is a classic clustering algorithm. It was introduced by Carpenter et al. [1]. Fuzzy ART is an unsupervised learning algorithm that incorporates the min operator from the fuzzy set theory into an ART based neural network. This algorithm can learn stable clusters from continuous input patterns. Besides, it is effective for large scaled data, and it runs fast compared to other clustering algorithms [2,3]. These characteristics make Fuzzy ART able to detect clusters with different sizes. Then, small clusters, having few instances, could be detected and separated from larger clusters.

Fuzzy ART is a clustering algorithm that demonstrated good performance in many real cases [2,4,5]. However, in the case where some labeled data instances are available, Fuzzy ART cannot consider this information to improve the clustering quality. This paper presents a simple and efficient solution to initialize Fuzzy ART with available labeled data instances. Indeed, the proposed approach is a semi-supervised Fuzzy ART (SSFuzzyART) that considers available labeled data instances to initialize clusters.

Binary data is data composed of vectors of arbitrary size whose categories are drawn in a two-valued distribution only. With modalities 0 or 1, a vector of binary data can be 010111001010. They are used in various fields of application and to model several types of data or phenomena. For example, a text document can be modeled by a binary vector representing the presence or absence of terms or characters. In addition, several automatic learning techniques have been introduced for the exploitation of this particular data, [6–9]. In this article, we consider the binary controlled clustered data to deeply study the characteristics of the proposed SSFuzzyART algorithm. For that purpose, we need to be able to manage the data characteristics and clusters separability to evaluate the SSFuzzyART algorithm. Binary clustered data is adaptable for such cases. In fact, binary data has only two

* Corresponding author.
E-mail addresses: siwar.jendoubi@upskills.ai (S. Jendoubi), thong.tran@upskills.ai (T. Tran).
[1] http://www.upskills.com

https://doi.org/10.1016/j.array.2023.100319
Received 14 March 2023; Received in revised form 30 August 2023; Accepted 6 September 2023
Available online 9 September 2023

modalities, which make it easier to manage and generated clusters are better controlled.

In this article, an algorithm for generating binary data grouped by clusters of controlled variances is introduced. This algorithm allows the generation of binary data sets having characteristics known and mastered beforehand. These datasets are used to test and validate the effectiveness of binary data partitioning solutions. Being able to control the characteristics of the data generated also makes it possible to study the applicability of these methods and their weaknesses. These characteristics are also useful for choosing a machine learning technique. Indeed, the choice will be guided by the characteristics of the data and tests in circumstances similar to real cases. In addition, a study comparing dimension reduction algorithms was carried out to show the usefulness of the generated data in the choice of a dimension reduction algorithm, under the angle of the conservation of the characteristics of the clusters in space.

The contributions of this paper are the following: (1) the proposition of a semi-supervised version of the Fuzzy ART, SSFuzzyART, clustering algorithm. SSFuzzyART uses a set of available labeled instances to initialize the clusters centers. Also, a set of experiments is made to evaluate the proposed algorithm and to study its strength and weakness. (2) the proposition of a data generation algorithm and the study of the characteristics of the generated data and the use of the generated data to deeply study SSFuzzyART algorithm characteristics.

This paper is organized as follows: Section 2 presents an overview of the current state of the art on the semi-supervised clustering, Fuzzy ART and clustering data generation. Section 3 details the classic Fuzzy ART algorithm and the proposed SSFuzzyART. Section 4 presents the clustered binary data generation algorithm, and evaluates the generated clusters through a set of experiments. Section 5 introduces the considered evaluation measures, evaluates the characteristics of SSFuzzyART using the generated data, compares SSFuzzyART clustering to FuzzyART clustering and discusses the experiment results. Finally, Section 6 concludes the paper.

2. Related works

2.1. Semi-supervised clustering

Semi-supervised clustering combines elements of both semi-super vised learning and clustering analysis. By incorporating class labels, semi-supervised clustering directs the clustering procedure, leading to improve performance. Over the past years, many remarkable initiatives have surfaced, focusing on both exploration and practical implementation in various fields. [10].

Adaptive Resonance Theory (ART) and Semi-supervised learning has attracted several researchers' attention. Kim et al. [11] introduced a label propagation ART algorithm. It is a semi-supervised approach. Another semi-supervised Message Passing Adaptive Resonance Theory (MPART), [12], was also introduced. Recently, [13] proposed HANFA-ART with ACA which is a Hybrid Adaptive Neural-Fuzzy Algorithm based on Adaptive Resonant Theory with Adaptive clustering Algorithm. This model has nine major components, and the Adaptive Resonant Theory is one of them. Another interesting clustering solution was recently introduced by Masuyama et al. [14]. This work proposes a solution for better estimating the specification of a similarity threshold (vigilance parameter) which is an interesting parameter on which the prediction of the ART based models is highly dependent.

Fuzzy Adaptive Resonance Theory algorithm is a classic online clustering algorithm that is known to be sensitive to data order. To remedy this issue, several research papers introduced solutions for this problem. Elnabarawy et al. [15] used evolutionary computation to improve Fuzzy ART. Liew et al. [4] used a genetic algorithm for the same purpose.

Clusters initialization by seeding and semi-supervised clustering are usual machine learning techniques that are always used to make the clustering prediction more efficient. Basu et al. [16] used labeled data to initialize clusters for k-means algorithm. They assume that for each possible cluster in a given problem, they have at least one instance belonging to that cluster. Besides, they initialize each cluster with the mean of instances belonging to that cluster. Taghizabet et al. [17] proposed "ConvexClust". It is a semi supervised clustering k-means based solution that uses labeled data and the geometric point of view to improve the clustering results.

According to [18,19] that present detailed surveys about the semi-supervised clustering, most available semi-supervised clustering solutions consider the k-means clustering approach. However, to the best of our knowledge, there is no existing work that focus on a semi-supervised FuzzyART clustering approach. Indeed, Fuzzy ART is a specific clustering algorithm that showed its performance to detect clusters with different sizes and to overcome the case of imbalanced data. Then, a semi-supervised FuzzyART inherit from Fuzzy ART its characteristics and improves its results using labeled data.

2.2. Clustered data generation

Data is ubiquitous and is often a resource in various situations. However, generating such a value depends on higher quality data. In addition, in situations involving sensitive confidential data such as business data or background, it is imperative to maintain data confidentiality by adhering to quality standards. The demand for top-notch data and privacy-friendly practices has become increasingly evident in recent years, as companies and researchers use it more extensively.

The literature presents several approaches of synthetic data generation for clustering like [20–24] are introduced. Each of these approaches is generally trying to replicate some real world phenomena and to generate the needed data to evaluate and validate some specific solutions. In addition, the generated data could have some special characteristics related to the generated clusters shapes or features, for example. Let us consider the work of [24]. It presents clustered data generation approach called "repliclust" that allows to generate clusters with different geometric characteristics. However, their solution is not useful to generate binary data and does not allow managing clusters sparsity.

Several binary data generation algorithms, [23,25], aim to simulate correlation phenomena between binary variables. These works seek to simulate a phenomenon observed in several fields of research such as the correlation between genetic variables in bioinformatics, binary questionnaires (yes/no) carried out for sociological studies, etc. Several algorithms for generating correlated binary modalities with varied marginal probabilities have been introduced, [23,25]. The main objective of these algorithms is to be able to generate binary data with high dimensions and with variables presenting controlled correlations between them. These data generation algorithms are suitable for the study of data correlation. However, the data generated with these solutions is not suitable for the study of data clustering approaches.

Methods for generating longitudinally correlated binary data have been studied by [26]. These methods are useful for simulating large empirical surveys resulting in correlated longitudinal binary data where some dichotomous outcomes are measured on the same experimental units over time. For example, such data frequently arise in clinical trials where the same independent set of patients may be assessed for the presence or absence of a particular disease at different time points. The generation of this data considers the "cluster" aspect in the generation process. A cluster/group is defined by the fact that the measures within each cluster are correlated, but the measures of different clusters are independent. These generation methods are different from our solution. Indeed, these approaches do not control intra- and inter-cluster distances, but rather the correlation of data in each cluster.

Synthetic binary data has been used to experiment and validate clustering algorithms, like in the work of [8]. However, the generated

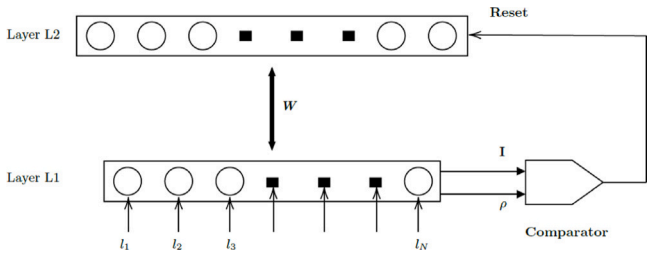

Fig. 1. Fuzzy ART architecture.

data used did not allow significant experimentation in terms of data partitioning. Indeed, the binary data generation method used only allowed the generation of high-dimensional data, but these data do not present detectable clusters. Thus, these datasets only allowed the study of the scalability of the proposed clustering algorithm.

To the best of our knowledge, the literature does not present a clustered data generation solution that could be useful to analyze all the characteristics of the proposed SSFuzzyART algorithm or that allows to generate similar clustered binary data to the generated data using the proposed algorithm. Our need for clustered binary data with the possibility to manage the intra and inter clusters distances and the data sparsity motivated us to introduce the data generation algorithm proposed in Section 4

3. Semi-supervised Fuzzy ART (SSFuzzyART)

In real world problems, some labeled data instances may be available. Fuzzy ART algorithm is not able to consider this available data in its clustering process. To resolve this problem and make Fuzzy ART output more efficient in such cases, this paper introduces the SSFuzzyART algorithm.

This section details first the Fuzzy ART algorithm. Next, it introduces the SSFuzzyART.

3.1. Fuzzy ART algorithm

Fuzzy ART [1] is a classic clustering algorithm that extends the ART1 algorithm. Indeed, ART1 algorithm categorizes only binary input instances. Then, Fuzzy ART extends this algorithm to categorize both binary and analog instances. For each input instance, the Fuzzy ART network looks for the "nearest" cluster that "resonates" with that instance. Next, it updates the center of the selected cluster with the input instance. The vigilance parameter, $\rho \in [0,1]$, is an input parameter of Fuzzy ART that determines the similarity of instances in the same cluster. When the vigilance parameter increases, the similarity of instances in the same cluster grows, which leads to a larger number of clusters, [27].

Fuzzy ART network is composed of two layers L_1 and L_2. Fig. 1 shows the architecture of Fuzzy ART network. The layer L_1 is the input presentation layer. It has as many neurons as the number of features in the input instance. The layer L_2 is the category representation layer. Then, neurons in the L_2 layer are added while the learning progresses. Indeed, for each new detected category, a new neuron is appended to the L_2 layer. Besides, the layer L_2 maintains one additional neuron in the uncommitted state. The uncommitted neuron is useful to maintain the vigilance criterion (equation (3)) satisfied when a new category arises [28].

When a new data instance, I, is presented in the layer L_1, each neuron in the layer L_2 computes a score for the category choice, using the following equation:

$$T_j = \frac{|I \wedge w_j|}{\alpha + w_j} \tag{1}$$

Where $\wedge$ is the Fuzzy AND operator defined as: $x \wedge y = \min(x, y)$, $|x|$ is defined as $|x| = \sum_k x_k$, $\alpha > 0$ is a parameter of Fuzzy ART, and w_j is the center of the j category (neuron). After computing the score, T_j, for all neurons in the layer L_2, the neuron, J, having the maximum score is selected:

$$T_J = \max_j(T_j) \tag{2}$$

Once a neuron (category) is selected from the layer L_2, Fuzzy ART checks the expectation between the input instance, I, and the selected category. This expectation is compared to the vigilance threshold, ρ, as follows:

$$\rho \leq \frac{|I \wedge w_J|}{I} \tag{3}$$

If this condition is satisfied, then the category neuron center is updated with the instance I using the following equation:

$$w_J = \beta(I \wedge w_J) + (1 - \beta)w_J \tag{4}$$

Where $\beta \in [0,1]$ is the learning rate.

If the condition of the formula (3) is not satisfied, Fuzzy ART reruns the neuron selection process until finding a neuron that satisfies the expectation condition (equation (3)), next, it updates that neuron using formula (4). In the case where no committed neuron fits with the expected condition, then, the data instance represents a new category and the uncommitted neuron is selected and updated to fit the new detected category, then, its weight vector is initialized with the input data instance. Next, a new uncommitted neuron is created. This process is repeated on all available data instances until getting a stable partition in which no instance changes its category. To limit the running time, a maximum number of epochs could be defined.

The next section introduces the proposed semi-supervised Fuzzy ART algorithm.

3.2. SSFuzzyART algorithm

The semi-supervised Fuzzy ART (SSFuzzyART) is introduced to consider the case where some labeled data instances are available. These instances may represent all possible labels in a given problem, or just a subset of these labels. Consequently, this data is useful to make Fuzzy ART clustering more efficient and to get better quality results.

The SSFuzzyART has the same steps as the classic Fuzzy ART algorithm, and it differs from it in the algorithm initialization. In fact, Fuzzy ART starts the clustering process with one neuron in the L_2 layer, which is the commitment neuron. This initialization makes it sensitive to the order of data entry. Indeed, the first coming instances will be used to initialize the clusters and then, will influence the clustering performance. The SSFuzzyART uses the labeled instances to initialize the categories in the L_2 layer. Then, each available category in the labeled data will be used to initialize its corresponding cluster, as shown in Fig. 2.

For a given category in the labeled data, a neuron in the L_2 layer is created to represent that category. Then, available data instances will be used to initialize the created neuron by computing its weights. Consequently, weights are computed by applying Eq. (4) on the set of data instances for a given category. Table 1 presents a computation example.

Three possible scenarios for the use of the labeled instances after the initialization step are possible:

1. In the first scenario, the labeled data is used to initialize category neurons (layer L_2). Next, it is no longer used in the clustering process.
2. In the second scenario, the labeled data is used to initialize category neurons (layer L_2). Next, their labels are discarded, and initialization instances are clustered with the rest of data regardless of their initial labels.

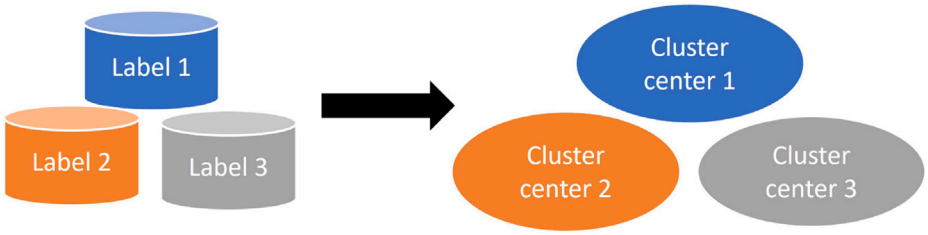

Fig. 2. SSFuzzyART initialization.

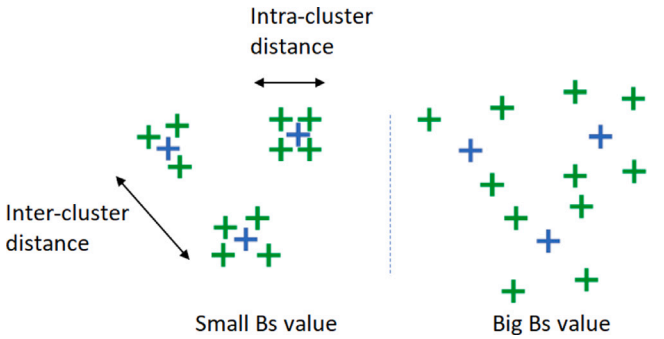

Fig. 3. An explanatory example of intra and inter-cluster distances and the impact of Bs on the generated clusters.

Table 1
Category weights computation example through applying the Fuzzy min operator on four instances of a given label, the self-learning rate $\beta = 1$ in this example.

Instance 1	000111
Instance 2	101011
Instance 3	010111
Instance 4	100011
Category weights	**000011**

3. In the third scenario, the labeled data is used to initialize category neurons (layer L_2). Next, initialization instances are clustered with the rest of the data, but their labels are set and cannot change during the clustering process, thus guiding the clustering process.

The choice of the scenario depends on the data to be clustered and the problem to be solved. For example, if there is some doubt about the labeled instances, it could be useful to apply the second scenario and to keep SSFuzzyART cluster these instances again. In cases where the quality of labeled instances is high, then it is recommended to consider the first or the third scenario.

In this paper, **the second scenario** is considered. Then, labels are no longer considered in the clustering process and the instances used for initialization are clustered with the rest of the data regardless of their initial labels. This choice allows studying the ability of SSFuzzyART to correctly put the labeled instances in their clusters.

4. Clustered binary data generation with controlled partitioning

This section present and explain the proposed data generation algorithm for binary clustered data. Besides, we present an empirical study of the generated clusters' separability.

4.1. Data generation algorithm

This section introduces the data generation algorithm and the parameters that allow controlling the characteristics of generated clusters. Algorithm 1 details the proposed data generation algorithm. The values of the binary modalities considered are either 1 or 0, but other pairs of values are also possible without losing generality. We designate by *bit* a modality that has exactly two possible values: 1 or 0.

Input
- C: Number of clusters
- Nc: $Nc = (n_1, n_2, \ldots, n_m)$ list of clusters sizes
- D: Number of features, i.e. generated vector size
- Bs: Bit swapping rate of generated vectors
- One_pr: Probability of having one value generated in the prototype vector. The default value of this parameter is 0.5

for *cluster* i *in* C **do**
  Randomly generate a prototype of dimension D whose values are taken from a Bernoulli distribution of expectation One_pr $Ber(One\_pr)$.
  for *vector* j *in* Nc_i **do**
    Copy the generated prototype and swap the state of $Bs\%$ randomly chosen bits
  end
end

Algorithm 1: Clustering binary data generation algorithm with controlled variance

The input parameters of the algorithm are: the number of clusters (C), the list of the number of vectors per cluster ($Nc = (n_1, n_2, \ldots, n_m)$), the dimension of the binary vectors D and the bit swapping rate "Bs". The bit-swapping rate controls the proximity of the vectors in their clusters (intra-cluster distance), while the average distance between clusters (inter-cluster distance) is defined by the prototypes. In this algorithm, Bs is the same for all clusters, as shown in Fig. 3. This means that the intra-cluster distance is on average the same for each cluster. The percentage of one value One_pr is an optional parameter that is set to 0.5 by default. This parameter could be specified to manage the percentage of one value in the generated prototype vector.

The data generation algorithm is detailed in the algorithm 1. Thus, C random vectors are randomly generated and are used as "prototypes" or "centers" of future clusters. For each cluster i, Nc_i vectors are generated by duplicating the prototype vector, then swapping the value of an arbitrarily defined $Bs\%$ fraction of randomly chosen bits. For example, given the following 10-bit reference vector 1001101101 and a $Bs = 20\%$, the following two vectors can be generated by randomly swapping two bits of the initial vector: 1001**11**0001 and 10**1**1101**0**01.

4.2. Evaluation of the generated clusters separation

The binary data generation algorithm introduced in the 4.1 section is evaluated in this section. Indeed, in order to use these artificially generated data with confidence, it is important to control the quality of their generation. This quality control is defined through the control of the distance between the vectors in the same cluster (intra-cluster distance), and the distance between clusters (inter-cluster distance). The set of generated vectors is called E. This set is partitioned into C clusters, the ith of which is named C_i.

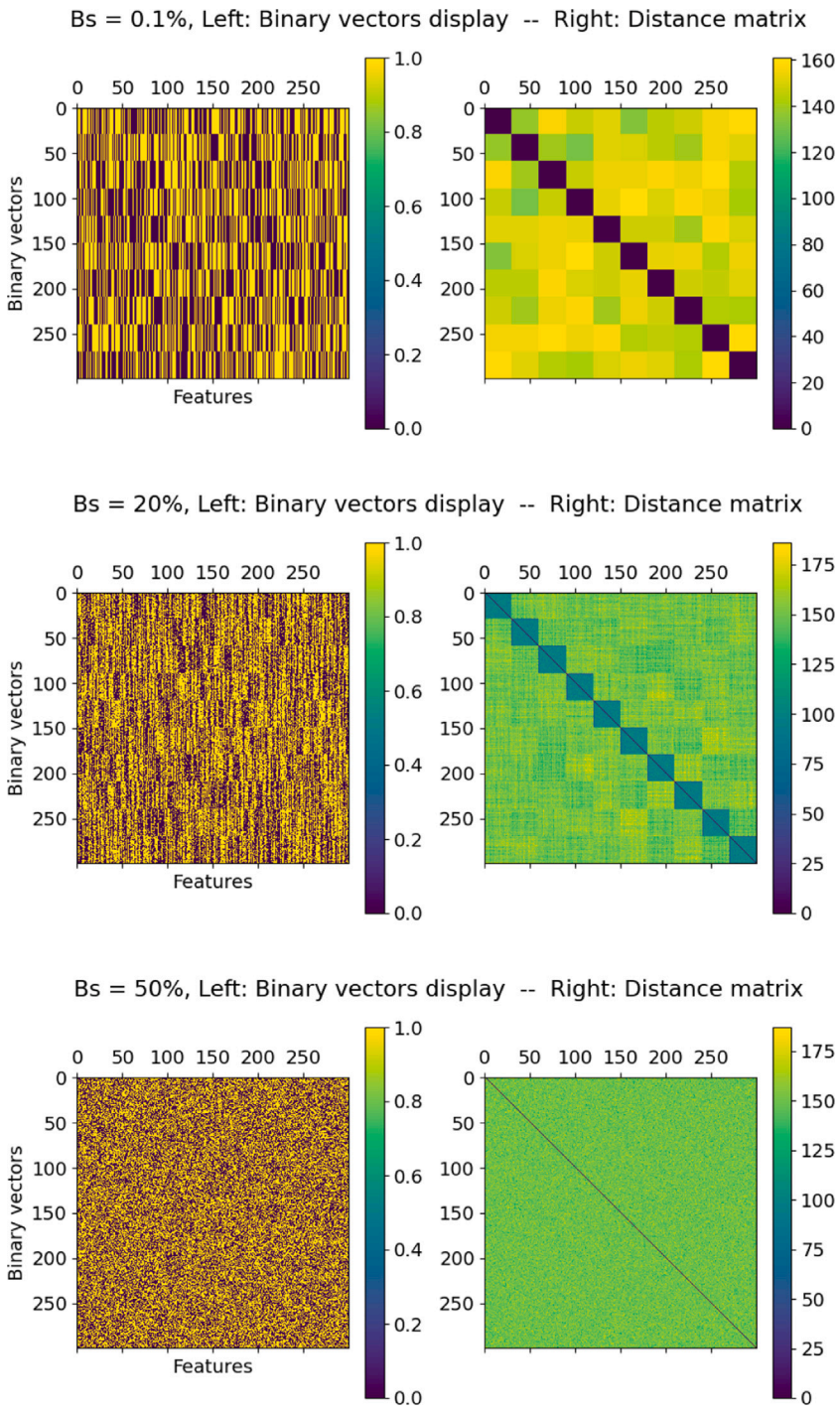

Fig. 4. Display of the generated binary vectors and their corresponding distance matrices for three values of Bs: 0.1%, 20% and 50% respectively. Number of clusters $C = 10$, Dimension $D = 300$ and $Nc = 30$. The Manhattan distance is used.

The binary data generation algorithm is, also, compared to a commonly used solution in the literature that is considered in this paper as a baseline solution. This solution consist of generating real-valued random clustered data and then binarizing the generated clusters against a threshold. For this purpose, we used the available function make_blobs[2] from the python library scikit learn. The generated samples follow the normal distribution. Then, the idea is to generate a set of real-valued random clusters. Next, a threshold is used to transform these real-values to binary values. Then, if a given real value is lower than the threshold, the binary value will be 0 else it will be 1. This algorithm is called Baseline data generation in the experiments.

We note that the algorithms and all the experiments are implemented with Python.

[2] make_blobslink

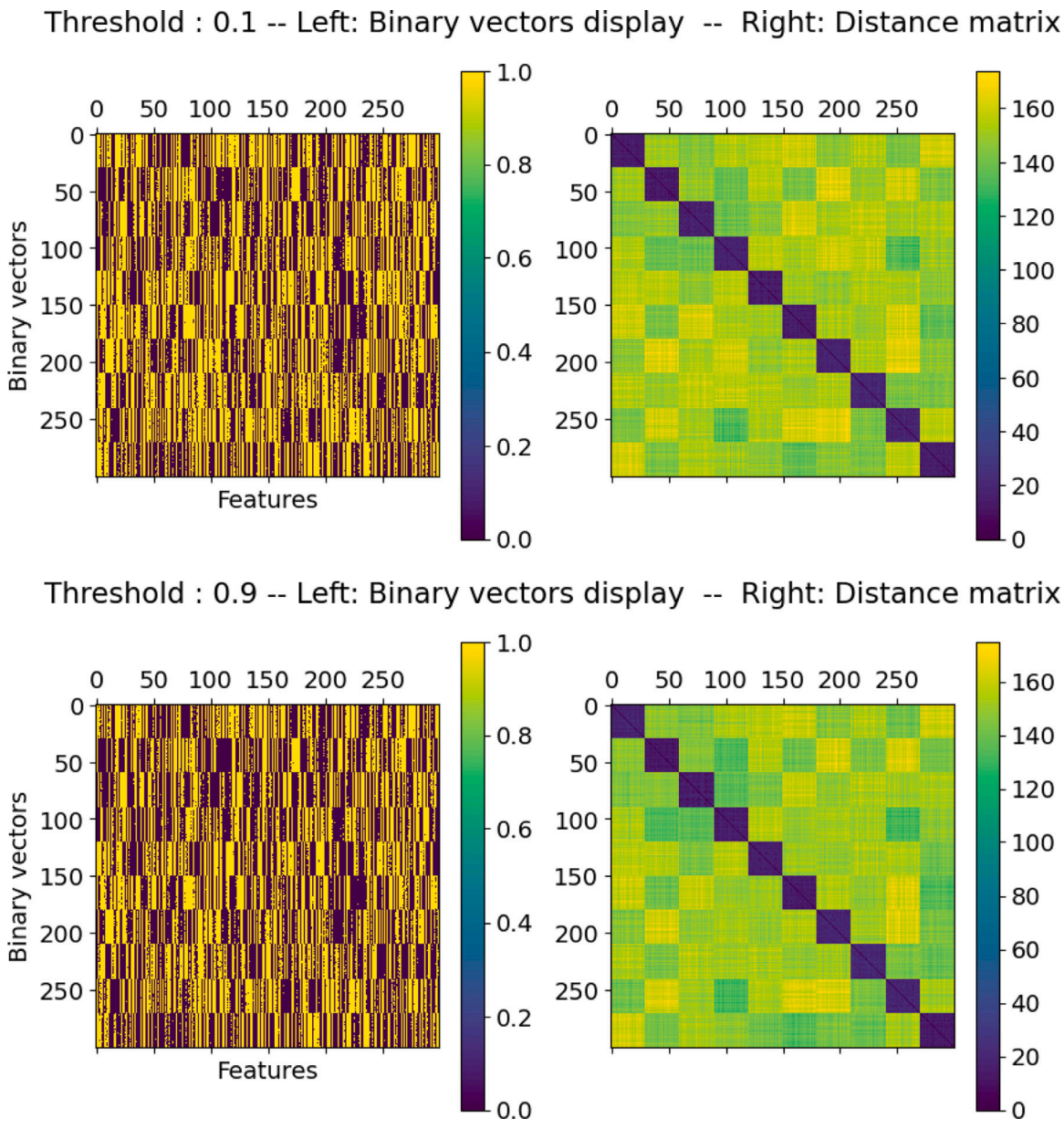

Fig. 5. Display of the generated binary vectors using the Baseline data generation algorithm and their corresponding distance matrices for two values of threshold: 0.1 and 0.9 respectively. Number of clusters $C = 10$, Dimension $D = 300$ and $Nc = 30$. The Manhattan distance is used.

Fig. 4 shows a visualization of three data sets and corresponding distance matrices with three-bit swapping rates $Bs = 0.1\%$, $Bs = 20\%$ and, $Bs = 50\%$ respectively. Besides, Fig. 5 shows a visualization of two data sets generated with the Baseline data generation algorithm. In this experiment, we generated one real valued data set as explained above, and we used two different binarization threshold values, 0.1 and 0.9 respectively. The distance matrix consists of calculating the distance between each pair of vectors (x, y). The distance used in this paper is the Manhattan distance:

$$Man(x, y) = \sum_{i=1}^{D} x_i - y_i \tag{5}$$

The vectors in Fig. 4 have a dimension $D = 300$. Besides, 10 clusters containing 30 vectors each are generated. When the bit-swapping rate is low ($Bs = 0.1\%$), the clusters are easily visible and separable. In coherence, the distance matrix shows that the data vectors belonging to the same clusters are close to each other. If the bit swapping rate increases to $Bs = 20\%$ the clusters remain visible but the average intra-cluster distance increases, leading to significantly more different vectors from each other. With Bs of 50%, there are no more visible clusters in the matrices. All the vectors are random, the average intra-cluster distance is equal to the average inter-cluster distance: the notion of the cluster has no longer any meaning in this case. The same configuration was considered in Fig. 5, then the generated vectors have a dimension $D = 300$. Besides, 10 clusters containing 30 vectors each are generated. and two threshold values are considered, 0.1 and 0.9. According to this figure, the clusters are easily visible and separable and the distance matrices show that the data vectors belonging to the same clusters are close to each other. However, we have the same visualization for the two threshold values. Consequently, the binarization threshold has no effect on the clusters. The Baseline data generation algorithm does not offer the possibility to manage the clusters properties, as we need to study the clustering algorithm characteristics.

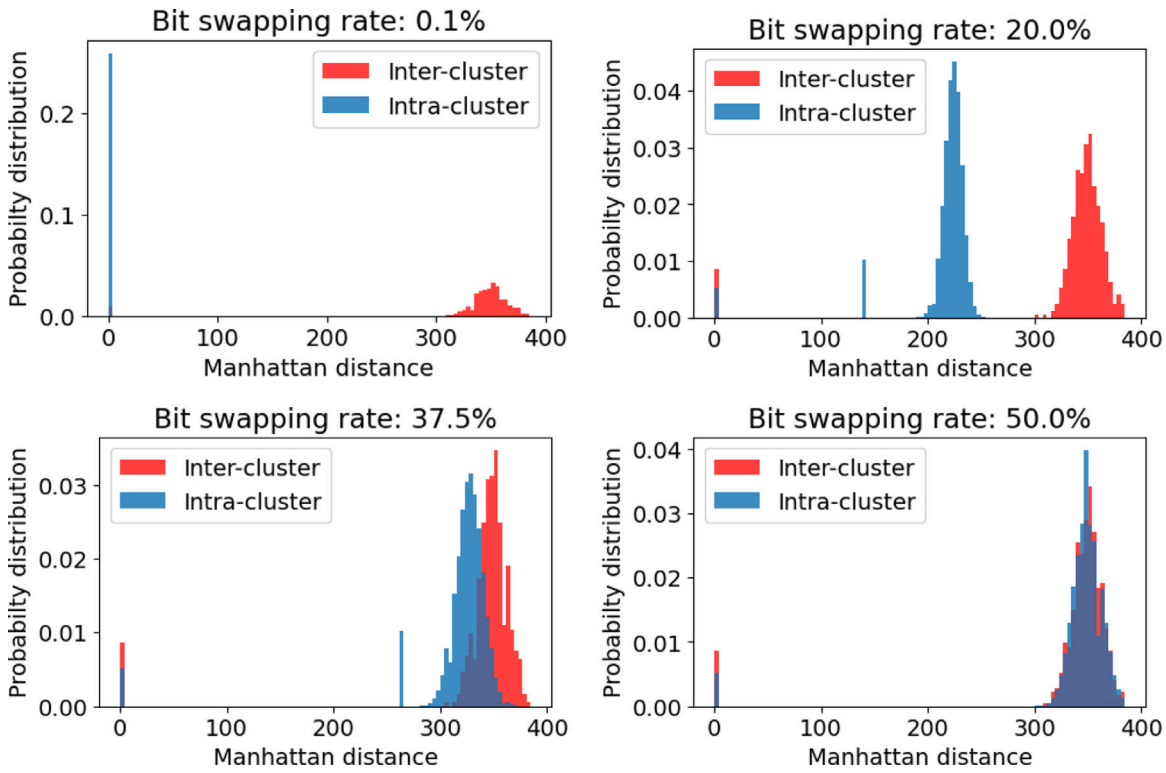

Fig. 6. Intra- and inter-cluster distance distributions Evolution in the function of bit swapping rate. $C = 30$, $D = 700$, and $Nc = 50$. The distance Manhattan is used.

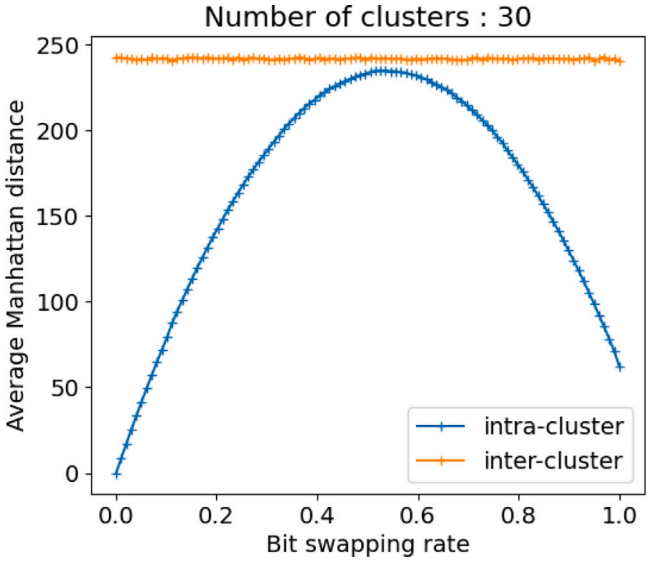

Fig. 7. Evolution of the intra and inter-cluster average distance according to the rate of bit swapping. Number of clusters $C = 30$, number of vectors per cluster $Nc = 15$ and dimension of the binary vector = 500. The Manhattan distance used.

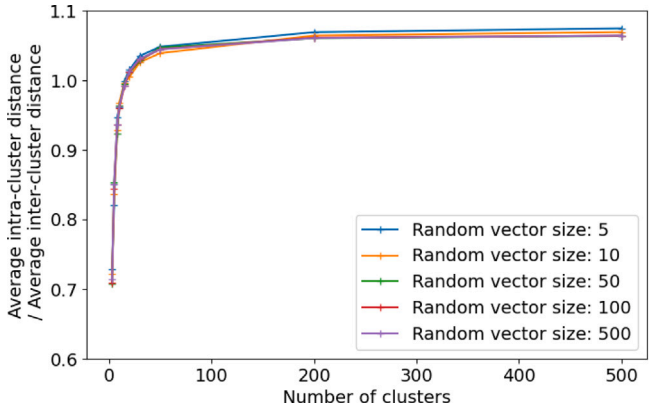

Fig. 8. Ratio between the average value of the inter-cluster distance and the maximum of the intra-cluster distance (several values of Bs were used to determine the maximum and the average), according to the number of clusters C and the dimension of the vector D. The Manhattan distance is used.

The experiment illustrated by Fig. 6 confirms these observations. This experiment consists in plotting the histograms of the intra- and inter-cluster distances as a function of the average bit-swapping rate. An intra-cluster distance is a distance between two vectors belonging to the same cluster C_i, defined as follows:

$$D_{intra}(x, y) = Man(x, y) \mid \forall C_i \in C,\ x,\ y \in C_i \tag{6}$$

Conversely, inter-cluster distances relate to vectors belonging to different clusters. Let C_i and C_j be two different clusters, this distance is expressed as follows:

$$D_{inter}(x, y) = Man(x, y) \mid \forall C_i, C_j \in C,\ C_i \neq C_j,\ x \in C_i\ et\ y \in C_j \tag{7}$$

Indeed, with a low Bs, the histogram of intra-cluster distances is located at 0, because the vectors are extremely similar to each other. As the rate of bit swapping Bs increases, the histogram of the intra-cluster distances widens and approaches the histogram of the inter-cluster distances. Indeed, the clusters grow because the number of swapped bits increases. For a high bit swapping rate ($Bs = 50\%$), the two histograms are merged: there are no more clusters, only a uniform cloud of points. Moreover, we note that the histogram of the inter-cluster distances does not depend on the Bs rate. In fact, these distances are fixed during the generation of the initial prototypes and,

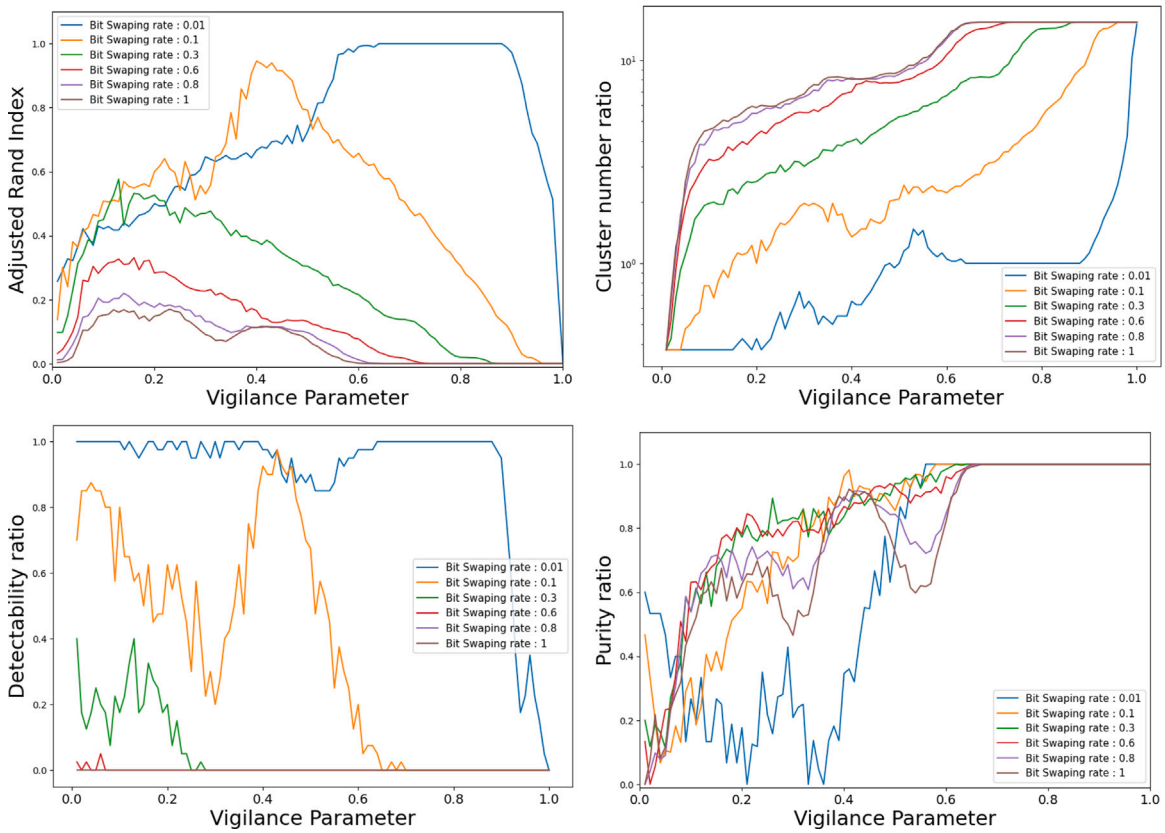

Fig. 9. SSFuzzyART clustering quality (compared to ground truth) versus the vigilance parameter for several bit-swapping rates.

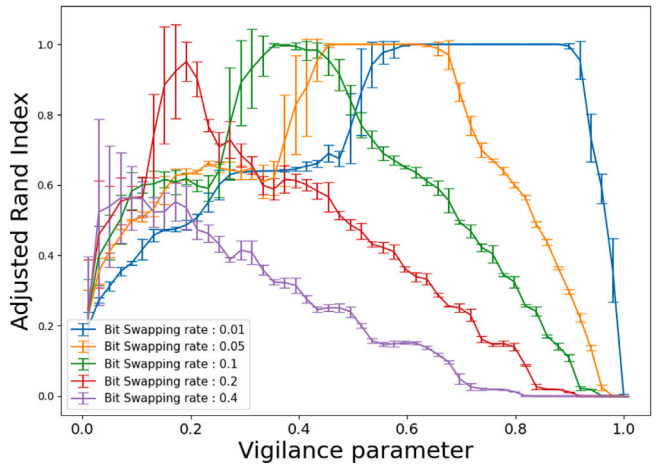

Fig. 10. Adjusted Rand Index versus the vigilance parameter for several values of bit swapping rate and binary vector feature number.

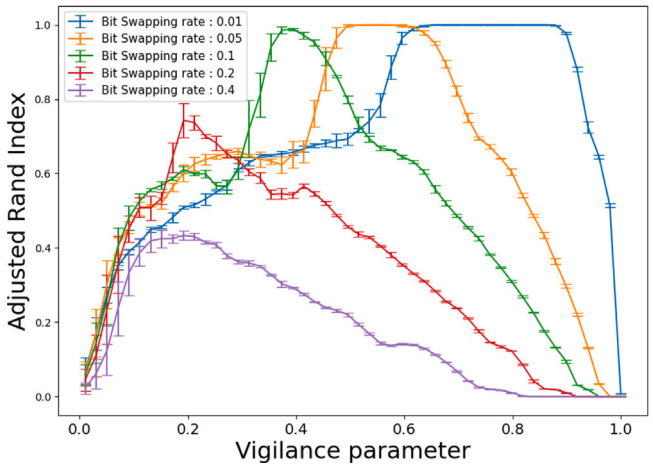

Fig. 11. Adjusted Rand Index versus the vigilance parameter for several values of bit swapping and cluster number. The error bar shows the variability associated with the cluster number.

in coherence with the definition of the algorithm, are independent of the rate Bs.

To better understand the impact of Bs on intra- and inter-cluster distances, we introduce the notions of average distances by the following definitions:

$$\overline{D_{intra}} = \frac{1}{N_{couples\ intra}} \sum_{x,y \in E} D_{intra}(x,y) \tag{8}$$

$$\overline{D_{inter}} = \frac{1}{N_{couples\ inter}} \sum_{x,y \in E} D_{inter}(x,y) \tag{9}$$

These average distances correspond to the arithmetic mean of the histograms presented in Fig. 6.

The Fig. 7 shows the impact of the bit swapping rate Bs on the intra and inter-cluster average distances. In coherence with Fig. 6, the inter-cluster average distance is independent of the bit swapping rate Bs. Moreover, the intra-cluster average distance increases until $Bs = 50\%$ then decreases beyond. Moreover, its maximum is almost confused with the value of the inter-cluster average distance. For low Bs, the intra-cluster distance is much lower than the inter-cluster distance: the clusters are very compact and far from each other. As the bit swapping rate increases, the clusters grow at a constant inter-cluster distance. When $Bs = 50\%$, the intra-cluster distance reaches its maximum, the radius of the clusters is close to the inter-cluster distance, so there are

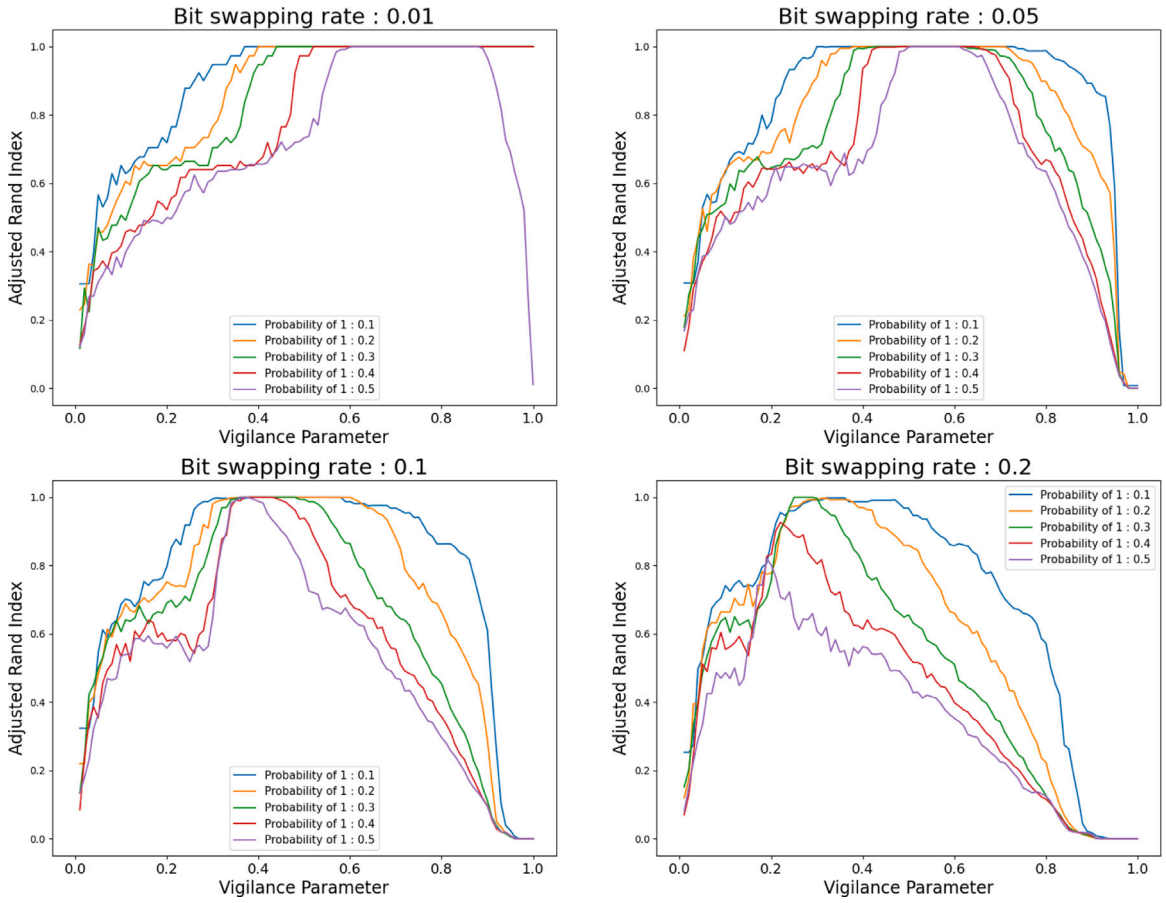

Fig. 12. Adjusted Rand Index for several bit swapping rates, for several occurrence probabilities of 1 in binary vectors.

no more clusters defined. The value of the maximum at $Bs = 50\%$ is linked to the arbitrary choice of the expectation of Bernoulli's law at 0.5 used for the generation of the prototypes. As Bs continues to increase, the inter-cluster distance decreases because the bit-swap potential has been exceeded, and any further bit change leads to the complementary vector of the initial vector. In the case where $Bs = 100\%$, all the vectors are equal to the complement of the prototype, except the prototype itself, hence the fact that the intra-cluster distance is not strictly zero.

Next, Fig. 8 illustrates the impact of the number of clusters C and the dimension of the vectors D on the intra- and inter-cluster distances. This figure presents the ratio between the average value of the inter-cluster distance and the maximum of the average intra-cluster distance, defined as follows:

$$R = \frac{\overline{D_{inter}}}{max_{x,y \in E}\ D_{intra}(x, y)} \tag{10}$$

depending on the number of clusters C and the dimension of the vector, D.

This figure shows that the vector dimension has no impact on the inter and intra-cluster distances. On the other hand, R depends on the number of clusters only if this number is less than 100. When the R indicator is less than 1, the clusters disappear for the bit swapping rates Bs less than 50%. Controlling the clusters concentration is therefore difficult in this case. In the case where the R indicator is greater than 1, the clusters disappear from a Bs equal to 50%. Therefore, the clusters can be controlled independently of the D dimension of the vectors. Moreover, the more the number of generated clusters increases, the easier it is to control the concentration of the vectors.

5. Experiments

Several experiments are made to evaluate the proposed SSFuzzyART. All experiments presented in this paper are implemented with Python3. Then a set of evaluation measures is introduced in Section 5.1. A first set of experiments evaluates the characteristics of SSFuzzyART (Section 5.2. Finally, an evaluation of SSFuzzyART and a comparison with FuzzyART using three available benchmarks is presented in Section 5.3.

5.1. Evaluation measures

Four evaluation measures are considered to evaluate the proposed SSFuzzyART algorithm, and to compare it to the classic Fuzzy ART. The considered measures are defined as follows:

- *Adjusted Rand index*[3]: it is a similarity score between two clustering in the range $[-1, 1]$. It considers all pairs of samples in the prediction and target clustering and checks if pairs are in the same or different clusters. When this score is equal to 1, then the clustering is perfect. When it is near zero, then the clustering is near the random clustering.
- *Number of cluster ratio*: the ratio of predicted clusters number by the target clusters number. If this ratio is equal to 1, then the

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html

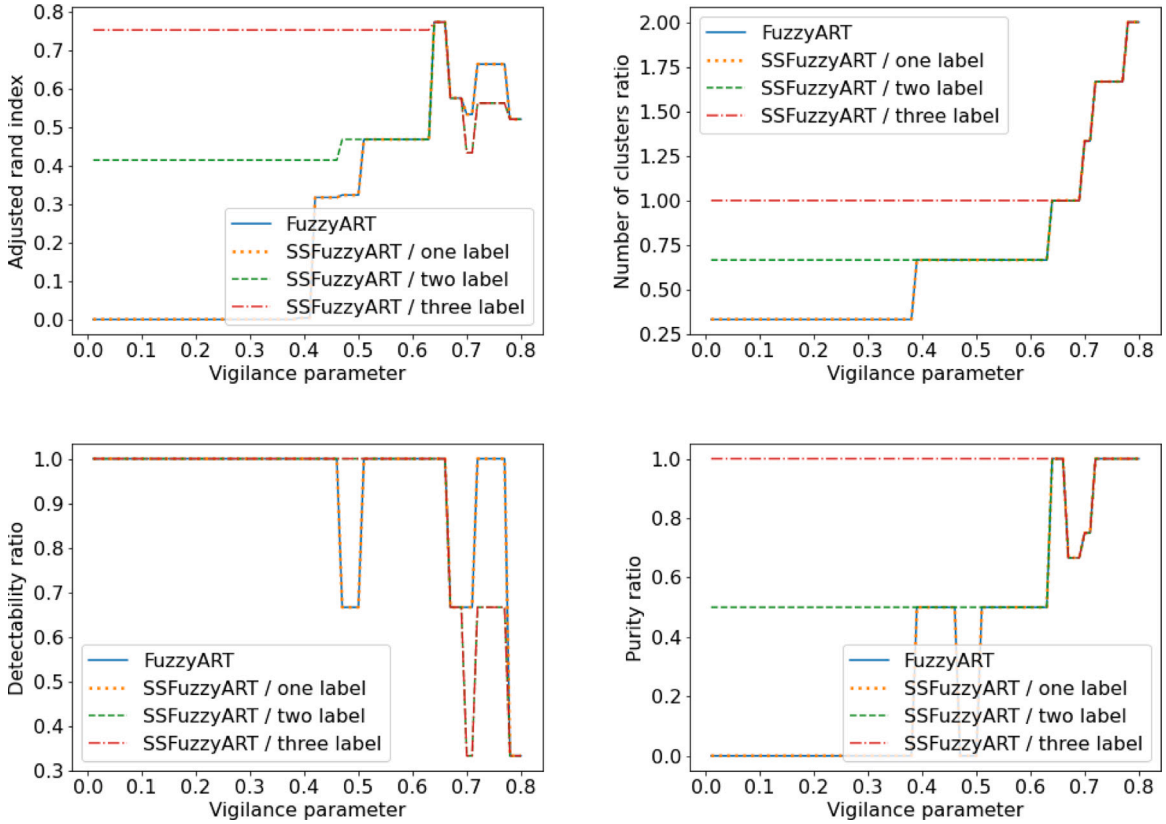

Fig. 13. SSFuzzyART VS Fuzzy ART on Iris dataset. In this experiment, three initialization samples were considered. The first one contains 5 instances having the same label. The second one contains 10 instances having two labels (five instances per label). The third one contains 15 instances having three labels (five instances per label).

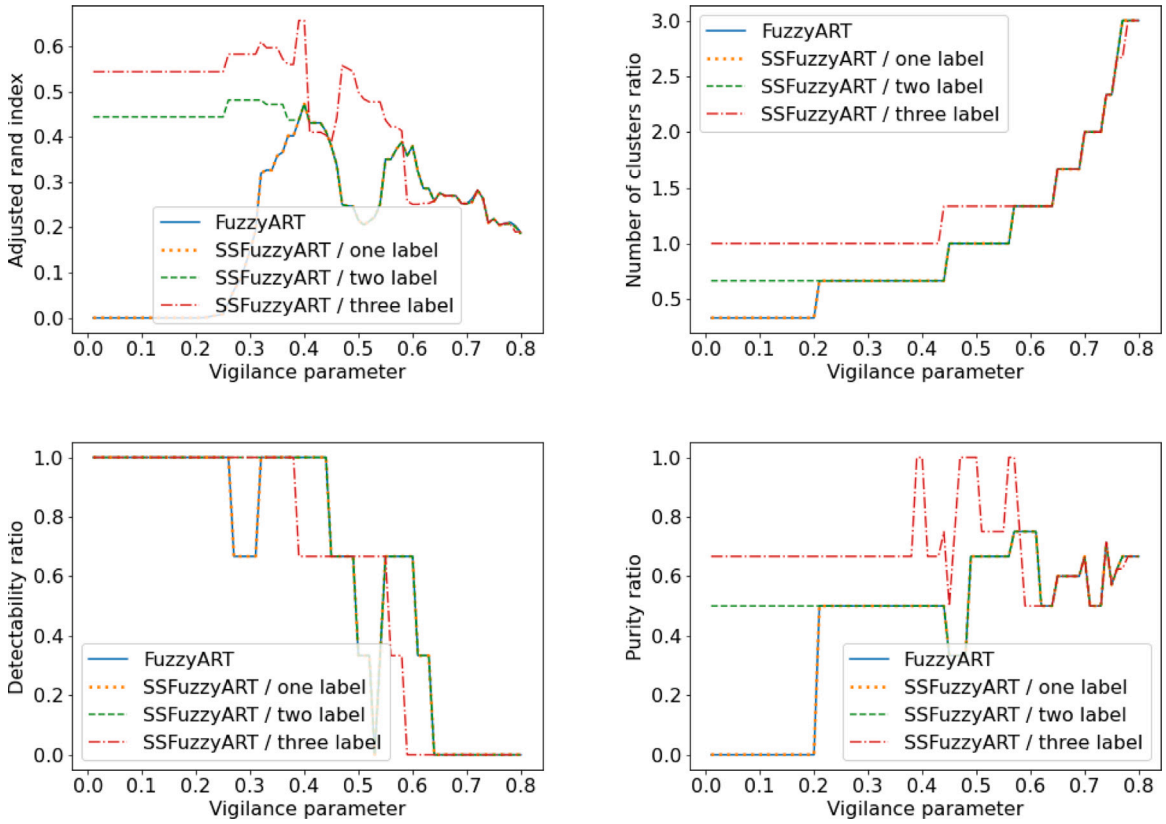

Fig. 14. SSFuzzyART VS Fuzzy ART on Wine dataset. In this experiment, three initialization samples were considered. The first one contains 5 instances having the same label. The second one contains 10 instances having two labels (five instances per label). The third one contains 15 instances having three labels (five instances per label).

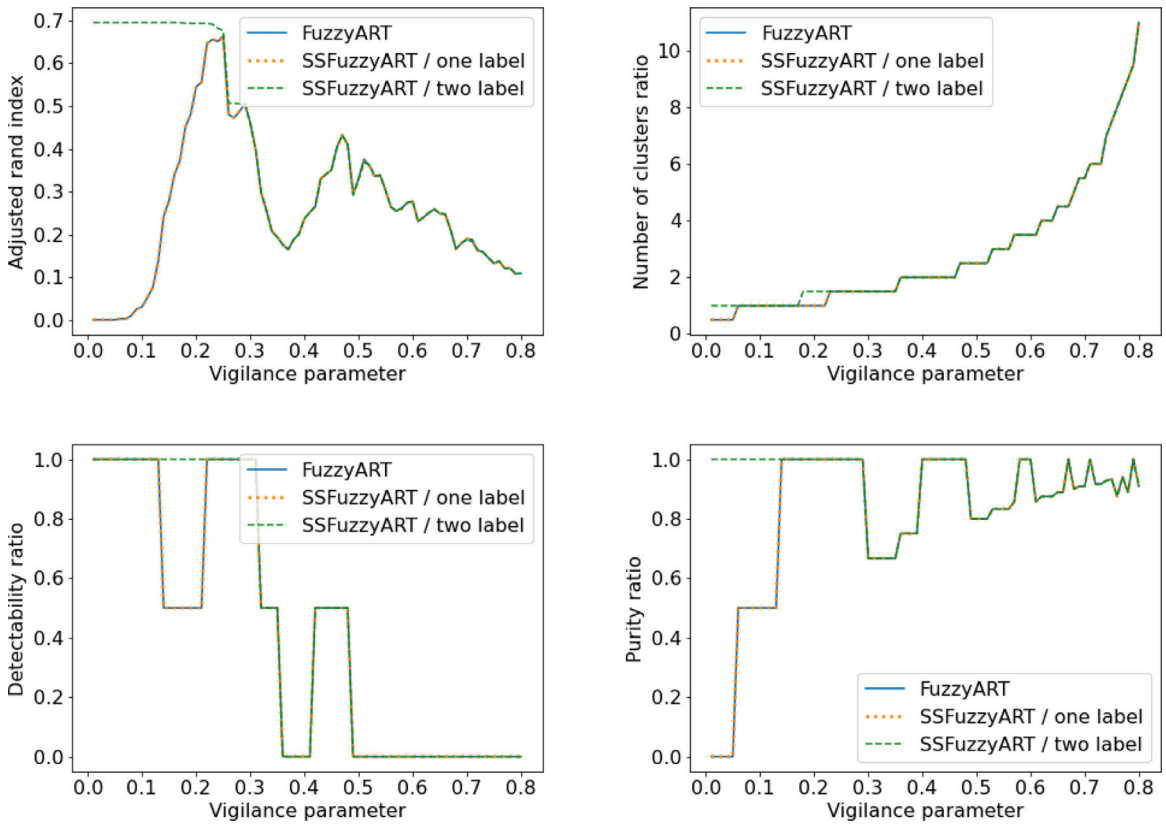

Fig. 15. SSFuzzyART VS Fuzzy ART on Breast cancer dataset. In this experiment, two initialization samples were considered. The first one contains 30 instances having the same label. The second one contains 60 instances having two labels (30 instances per label).

predicted clustering has exactly the same number of clusters as the target clustering. This measure is useful to evaluate Fuzzy ART and SSFuzzyART, as both do not use the number of clusters as input.

- *Cluster purity ratio*: the ratio of pure clusters, where a cluster is considered pure if at least 70% of its elements have the same label.
- *Label detectability ratio*: the ratio of detectable labels, where a label is considered detectable if at least 70% of data instances having that label are in the same cluster.

The considered cluster purity ratio and label detectability ratio have some tolerance on the definition of pure cluster and detectable label, respectively. This tolerance is defined by the fact that a cluster is pure if at least 70% of its elements have the same label, and a label is detectable if at least 70% of data instances having that label are in the same cluster. This tolerance allows these two measures to accept a clustering error of 30%. The choice of the tolerance value is defined according to the clustering problem, then for some problems it could be higher and for other problems it could be lower than 70%. In this paper, the clustering tolerance is fixed to 70%. A good prediction should have high cluster purity ratio and high label detectability ratio. This tolerance is useful to find the best compromise between these two measures. These two measures are influenced by the number of clusters. Consequently, when the number of clusters is high, these clusters are usually pure. However, in that case, the detectability of labels is low. With a high number of clusters, a given label instances could be scattered in several clusters.

The next section presents and discusses experiment results and the impact of the layer L_2 initialization on the final prediction of the SSFuzzyART.

5.2. SSFuzzyART characteristics study on generated data

SSFuzzyART is tested on a simulated dataset generated using the algorithm introduced in Section 4.2. It consists of binary vectors grouped in clusters whose variance can be controlled with the bit-swapping parameter. This parameter ranges from 0 to 1 and the cluster variance is the highest when the bit-swapping rate is equal to 0.5 (in most "normal" cases). This means that clusters are created with an assigned root cause.

As the binary vector dataset is simulated, the cluster in which each point belong is known. This allows for assessing the ability of SSFuzzyART to effectively cluster binary vectors. The quality of the SSFuzzyART clustering is assessed using the Adjusted Rand Index, which compares the SSFuzzyART clustering with the known vector assignment from the generation process.

The Adjusted Rand Index, the number of cluster ratio, the detectability ratio, and the purity ratio are computed for several values of bit swapping rates and vigilance parameters. The maximum number of epochs for SSFuzzyART is set at 20. The number of created clusters is 40 with 15 vectors each and the vectors have 200 features. Besides, 75 initialization vectors having 15 labels (5 vectors per label) are considered in the initialization set of SSFuzzyART. Results are presented in Fig. 9.

According to the adjusted rand index curves in Fig. 9, for low values of bit swapping (meaning clusters are well separated), SSFuzzyART can perfectly cluster the binary vectors for a large range of vigilance parameters (from 0.6 to 0.9). If the vigilance parameter is too small, then the created clusters are too coarse and contain several individuals/vectors having different types. If the vigilance parameter is too close to 1, then the clusters are too granular and individuals/vectors having the same type are spread on multiple clusters. If the bit swapping rate increases,

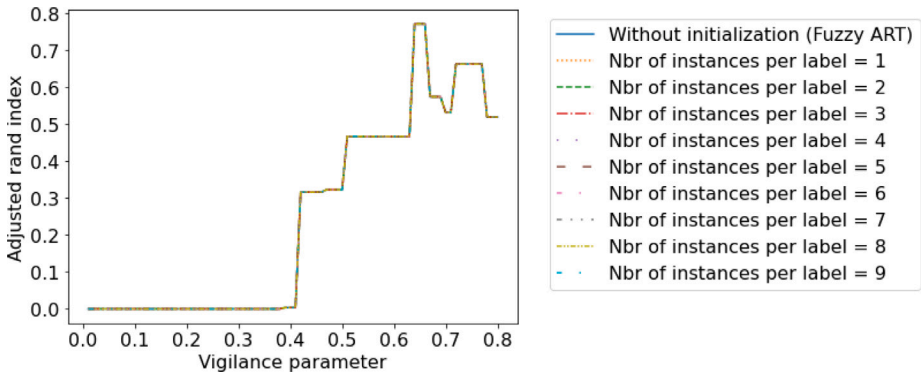

Fig. 16. Impact of the number of considered instances per label in the initialization sample of the SSFuzzyART compared to Fuzzy ART (without initialization). Iris dataset is used, only one label is considered in this experiment.

meaning that clusters are bigger and closer to one another, SSFuzzyART is less able to accurately cluster the data. Moreover, the maximum of the Rand index moves to the left: as the generated cluster is bigger and closer, some begin to merge and the vigilance parameter allows for the maximum Rand index is lower to allow coarse clusters to be identified.

According to the cluster number ratio curves in Fig. 9, whatever the variance of clusters (the bit swapping rate), the number of clusters created is globally increasing with the vigilance parameter. This is coherent with the internal working of SSFuzzyART. Moreover, the vigilance parameter values allow the creation of the exact same number of clusters than the ground truth decreases when the bit-swapping rate increases. This means that for coarser cluster structures, using a lower vigilance parameter allows for better retrieval of the cluster structure.

According to the detectability ratio curves in Fig. 9, the label detectability decreases when the vigilance parameter increases. The more clusters are created, the more the labels are spread in numerous clusters, therefore reducing their detectability.

According to the purity ratio curves in Fig. 9, cluster purity is high when the vigilance parameter is high. This makes sense because when a lot of clusters are created, they contain only a few individuals that are likely to have the same root cause.

To summarize, provided clusters are well-defined and separated, SSFuzzyART can retrieve the clustering structure of binary vectors

In the second experiment, the effect of the number of features (the binary vector length) is studied by computing the Adjusted Rand Index for several values of the feature number (200, 500, 1000, 2000) and different values of bit swapping. The generated dataset contains 40 clusters with 15 vectors each, and the vectors have 200 features. Besides, 75 initialization vectors having 15 labels (5 vectors per label) are considered in the initialization set of SSFuzzyART. Results are displayed in Fig. 10.

According to Fig. 10, the feature number has a relatively small impact on the Adjusted Rand Index, mainly for vigilance parameter values that are less than 0.5. Consequently, the SSFuzzyART clustering process can be achieved for different values of feature numbers. The feature number has no significant impact on clustering results.

The effect of the cluster number to detect (the number of root causes) is displayed in Fig. 11. The number of features of the binary vector is 200, the number of vectors per cluster is 15, and 75 initialization vectors having 15 labels (5 vectors per label) are considered in the initialization set of SSFuzzyART. According to Fig. 11, the number of clusters has little impact on the behavior of SSFuzzyART. Consequently, SSFuzzyART's capacity to cluster data is not dependent on the number of clusters, but more on the relative distance between clusters (accounted by the bit-swapping parameter).

In Fig. 12, we investigate the effect of the occurrence probability of 1 in binary vectors. In fact, real data are known to be sparse, with more 0 values than 1 values in their feature vectors. Fig. 12 displays the adjusted Rand index for several occurrence probabilities of 1 in random vectors, for several bit swapping rate. Having vectors biased toward 0 (meaning having less 1 values than 0 values), characterized by low values of occurrence probability of 1, allows to SSFuzzyART to find optimal partitioning of the vectors for a larger or different range of vigilance parameters.

Table 2
Considered benchmark datasets.

Benchmark	Number of categories	Number of instances	Number of features
Iris	3	150	4
Wine	3	178	13
Breast cancer	2	569	30

5.3. SSFuzzyart evaluation and comparison with FuzzyART

To evaluate the proposed SSFuzzyART algorithm and compare its performance to the classic FuzzyART, a set of experiments is made to compare Fuzzy ART results to the SSFuzzyART results using three benchmark datasets. The considered datasets were selected from UCI machine learning repository.[4] Table 2 presents the considered datasets.

To evaluate SSFuzzyART, a set of experiments on three benchmarks (Table 2) is carried out. The clustering results of SSFuzzyART are compared to Fuzzy ART results according to the four considered evaluation measures. In each experiment, a sample from the dataset is selected to initialize the L_2 layer in SSFuzzyART.

Fig. 13, Fig. 14 and Fig. 15 presents the experiment results on respectively Iris, Wine and Breast cancer datasets. In these experiments, the Adjusted Rand score, the number of clusters ratio, the purity ratio and the detectability ratio were measured in function of the vigilance parameter ρ to compare the SSFuzzyART to Fuzzy ART.

Moreover, to study the impact of the number of labels/category used in the initialization sample, a set of experiments was carried out using different samples containing a different number of labels, depending on the selected dataset. The number of selected instances depends also on the number of labels, to ensure that labels are equally present in the sample. For example, the Iris dataset contains three labels. The first sample out of the Iris Dataset contains five instances all having the same label, the second sample contains ten instances belonging to any of two different labels (five instances per label) and the third sample contains fifteen instances from the three available labels (five instances per label). In a given sample, all considered labels are represented with the same number of instances.

[4] https://archive.ics.uci.edu/ml/datasets.php

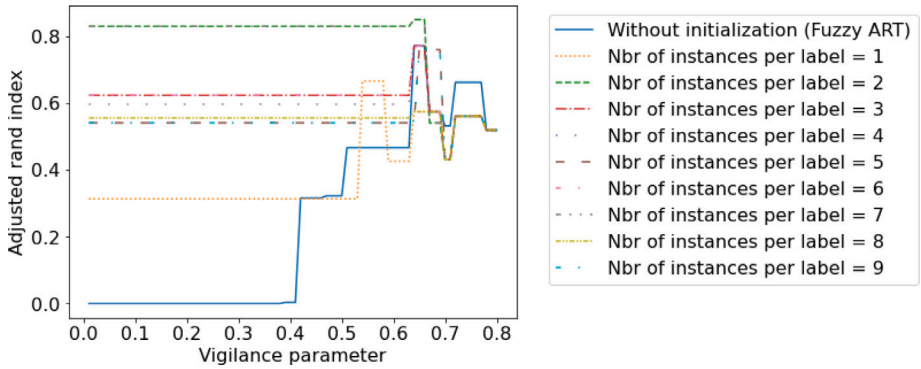

Fig. 17. Impact of the number of considered instances per label in the initialization sample of the SSFuzzyART compared to Fuzzy ART (without initialization). Iris dataset is used, and three labels are considered.

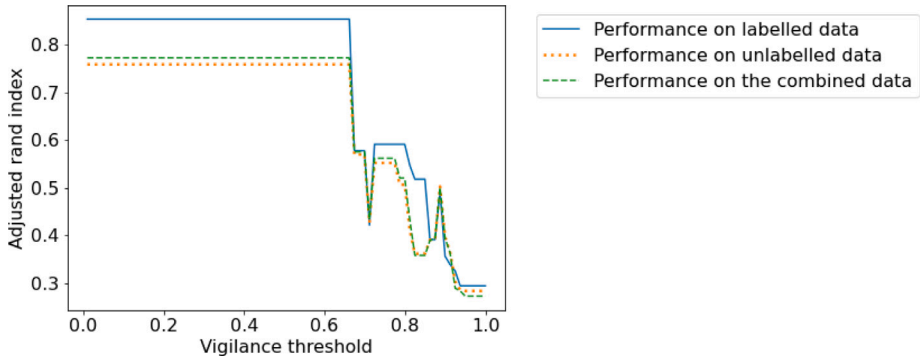

Fig. 18. SSFuzzyART Adjusted Rand index on the labeled data (seeds), the unlabeled data and the combined dataset. Iris dataset is used, and three labels are considered with 7 random instances per label.

According to the results in Fig. 13, Fig. 14 and Fig. 15, the SSFuzzyART leads to better results than Fuzzy ART when the vigilance parameter, ρ, values are low, and they yield similar results when ρ values are higher. Indeed, when the ρ parameter is high, the tolerance of the algorithm to create a new cluster is higher, which increases the number of clusters, then the initialization that is made to guide the clustering process is loosed. However, when the ρ parameter is low, the tolerance to create a new cluster is low too. Then, the number of cluster is also low, and the algorithm will not be allowed to add many clusters, especially when most possible labels are represented with corresponding neurons in the L_2 layer.

Another important conclusion from these experiments is that the available number of labels in the initialization sample has an impact on the output prediction. When only one label is available in the initialization sample, the SSFuzzyART and Fuzzy ART have almost the same output results. However, when the number of available labels in the initialization sample increases, the quality of the clustering increases too. Besides, according to the results in Fig. 13 and Fig. 14, even with two available labels from three in the labeled data, the SSFuzzyART is able to improve the clustering performance compared to the classic Fuzzy ART results.

Furthermore, the classic Fuzzy ART algorithm is sensitive to data order and its results are impacted with it. Indeed, clusters are created gradually with the arriving of data instance, as explained in above. With the initialization step, the algorithm is provided with some information that helps to better detect the clusters. Consequently, SSFuzzyART improves the clustering performance thanks to this initialization.

A second experiment is made to study the impact of number of instances in the initialization sample if there is only one available label in the data. Nine different samples were considered, and the number of instances in these samples is in the range $[1, 9]$ and all samples are selected from instances that belong to only one label. Fig. 16 presents the results of this experiment. Fig. 16 shows overlapping curves, i.e., all curves are equals. These results mean that a sample with only one label is not having any impact on the clustering results, even with many instances having that label. This experiment confirms the results of Fig. 13. In fact, having only one label in the initialization sample have no impact on the clustering results of SSFuzzyART. To have some improvement, the number of available labels must be greater than one.

A third experiment is carried out to study the impact of the number of instances used during initialization on the clustering results. Fig. 17 shows the Adjusted Rand index with different vigilance threshold values and nine different samples. All these samples contain random instances from the three labels. The number of instances per label is in the range $[1, 9]$. According to Fig. 17, increasing the number of initialization instances always improves the Adjusted Rand index. Indeed, even with one instance per label, SSFuzzyART has a Rand index that is greater than or equals to the Fuzzy ART Rand index. Furthermore, a number of instances per label that is greater than 2 has always a positive impact on the Rand index, according to these results.

In the previous experiments, the labeled instances that are used to initialize SSFuzzyART are clustered with the rest of the data.

Then, a last experiment is carried out to compare the performance of SSFuzzyART on the labeled data (labeled instances that are used as seeds), the unlabeled data (the rest of the dataset) and the combined dataset. Iris dataset is considered in this experiment. Fig. 18 presents the results of this experiment. According to the results in Fig. 18, the highest Adjusted Rand score is obtained on the labeled data. This is an expected result as the labeled data is used to initialize SSFuzzyART second layer neurons. The Adjusted Rand score on the unlabeled data is almost equal to the Adjusted Rand score of the entire dataset. These results confirm that SSFuzzyART has a positive impact on the performance of the clustering. Besides, the initialization of the layer L_2 is useful to obtain a better clustering prediction.

6. Conclusion

This paper introduces a SSFuzzyART algorithm. It is a Fuzzy ART semi supervised clustering algorithm that can use a set of available labeled data to improve the clustering prediction. SSFuzzyART uses the labeled instances to initialize its categories in the L_2 layer. This initialization is useful to guide the clustering process, and it influences positively the clustering prediction. Besides, the classic Fuzzy ART algorithm is sensitive to the choice of the vigilance parameter that manage the tolerance of the algorithm to create a new cluster, this sensitivity has been inherited by SSFuzzyART. Then, it is generally recommended choosing a relatively low ρ to get good performance. The ρ value could depend on the number of categories that are present in the initialization data. Then, if all categories are present, a lower ρ is recommended.

A set of experiments was made and discussed to deeply study the SSFuzzyART characteristics and to compare it to the classic Fuzzy ART algorithm. According to these experiments, SSFuzzyART initialization improved the clustering performance, especially with smaller ρ values. Besides, the number of clusters present in the initialization sample has an important impact on the algorithm performance. Finally, the number of instances per label in the initialization sample has no impact on the clustering performance.

In the future work, it will be interesting to test SSFuzzyART in a real world problem and to study its impact on real data. A second perspective is to compare the three possible scenarios introduced in Section 3.2, and to study their impact on real world problems. A Third perspective of this work could be to find a solution to influence the Fuzzy ART clustering process with available labeled data.

CRediT authorship contribution statement

Siwar Jendoubi: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Investigation, Writing – review & editing. **Aurélien Baelde:** Software, Conceptualization, Validation, Investigation, Writing – review & editing. **Thong Tran:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Carpenter GA, Grossberg S, Rosen DB. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Netw 1991;4(6):759–71.

[2] Bingwen C, Wenwei W, Qianqing Q. Infrared target detection based on fuzzy ART neural network. In: 2010 Second international conference on computational intelligence and natural computing. IEEE; 2010, http://dx.doi.org/10.1109/cinc.2010.5643745.

[3] Ilhan S, Duru N, Adali E. Improved fuzzy art method for initializing K-means. Int J Comput Intell Syst 2010;3(3):274. http://dx.doi.org/10.2991/ijcis.2010.3.3.3.

[4] Liew WS, Loo CK, Wermter S. Emotion recognition using explainable genetically optimized fuzzy ART ensembles. IEEE Access 2021;9:61513–31. http://dx.doi.org/10.1109/access.2021.3072120.

[5] Djellali C, adda M, Moutacalli MT. A comparative study on fuzzy clustering for cloud computing. Taking web service as a case. Procedia Comput Sci 2021;184:622–7. http://dx.doi.org/10.1016/j.procs.2021.04.024.

[6] Bouguila N. On multivariate binary data clustering and feature weighting. Comput Stat Data Anal 2010;54(1):120–34. http://dx.doi.org/10.1016/j.csda.2009.07.013.

[7] Wang X, Kaban A. Finding uninformative features in binary data. In: Lecture notes in computer science. Springer Berlin Heidelberg; 2005, p. 40–7. http://dx.doi.org/10.1007/11508069_6.

[8] Ordonez C. Clustering binary data streams with K-means. In: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery - DMKD '03. ACM Press; 2003, http://dx.doi.org/10.1145/882082.882087.

[9] Kaban A, Girolami M. Initialized and guided EM-clustering of sparse binary data with application to text based documents. In: Proceedings 15th international conference on pattern recognition. IEEE Comput. Soc; 2000, http://dx.doi.org/10.1109/icpr.2000.906182.

[10] Cai J, Hao J, Yang H, Zhao X, Yang Y. A review on semi-supervised clustering. Inform Sci 2023.

[11] Kim T, Hwang I, Kang GC, Choi W-S, Kim H, Zhang BT. Label propagation adaptive resonance theory for semi-supervised continuous learning. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing. IEEE; 2020, p. 4012–6.

[12] Kim T, Hwang I, Lee H, Kim H, Choi W-S, Lim JJ, et al. Message passing adaptive resonance theory for online active semi-supervised learning. In: International conference on machine learning. PMLR; 2021, p. 5519–29.

[13] Akpan VA, Agbogun JB. Hybrid adaptive neural-fuzzy algorithms based on adaptive resonant theory with adaptive clustering algorithms for classification, prediction, tracking and adaptive control applications. Am J Intell Syst 2022;12(1):9–33.

[14] Masuyama N, Amako N, Yamada Y, Nojima Y, Ishibuchi H. Adaptive resonance theory-based topological clustering with a divisive hierarchical structure capable of continual learning. IEEE Access 2022;10:68042–56.

[15] Elnabarawy I, Tauritz DR, Wunsch DC. Evolutionary computation for the automated design of category functions for fuzzy ART. In: Proceedings of the genetic and evolutionary computation conference companion. ACM; 2017, http://dx.doi.org/10.1145/3067695.3082056.

[16] Basu S, Banerjee A, Mooney R. Semi-supervised clustering by seeding. In: Proceedings of 19th international conference on machine learning. Citeseer; 2002.

[17] Taghizabet A, Tanha J, Amini A, Mohammadzadeh J. A semi-supervised clustering approach using labeled data. Sci Iran 2023;30(1):104–15.

[18] Bair E. Semi-supervised clustering methods. Wiley Interdiscip Rev Comput Stat 2013;5(5):349–61. http://dx.doi.org/10.1002/wics.1270.

[19] Qin Y, Ding S, Wang L, Wang Y. Research progress on semi-supervised clustering. Cogn Comput 2019;11(5):599–612. http://dx.doi.org/10.1007/s12559-019-09664-w.

[20] Pei Y, Zaïane O. A synthetic data generator for clustering and outlier analysis. Educ Res Arch 2006.

[21] Dahmen J, Cook D. SynSys: A synthetic data generation system for healthcare applications. Sensors 2019;19(5):1181.

[22] Lahariya M, Benoit DF, Develder C. Synthetic data generator for electric vehicle charging sessions: modeling and evaluation using real-world data. Energies 2020;13(16):4211.

[23] Jiang W, Song S, Hou L, Zhao H. A set of efficient methods to generate high-dimensional binary data with specified correlation structures. Amer Statist 2020;1–13. http://dx.doi.org/10.1080/00031305.2020.1816213.

[24] Zellinger MJ, Bühlmann P. repliclust: Synthetic data for cluster analysis. 2023, arXiv preprint arXiv:2303.14301.

[25] Lunn A, Davies SJ. A note on generating correlated binary variables. Biometrika 1998;85(2):487–90. http://dx.doi.org/10.1093/biomet/85.2.487.

[26] Farrell PJ, Rogers-Stewart K. Methods for generating longitudinally correlated binary data. Internat Statist Rev 2008;76(1):28–38. http://dx.doi.org/10.1111/j.1751-5823.2007.00017.x.

[27] Sengupta S, Ghosh T, Dan PK, Chattopadhyay M. Hybrid fuzzy-ART based K-means clustering methodology to cellular manufacturing using operational time. 2012, arXiv preprint arXiv:1212.5101.

[28] da Silva LEB, Elnabarawy I, Wunsch II DC. A survey of adaptive resonance theory neural network models for engineering applications. Neural Netw 2019;120:167–203.