


LFA Mini-Project

Fuzzy Semi-Supervised Learning Model

Francois QUELLEC - Arthur PASSUELLO

June 10, 2018

1 Project Introduction

For this project, no model is given *a priori*, we will have to do our own research to determine which model would be the most appropriate to our constraints :

- Combine both supervised an unsupervised learning
- Keep a good interpretability factor
- Use supplied dataset

From there on, every choice is our own.

1.1 Semi-Supervised Learning

Semi-supervised learning always implies two parts, one supervised and the other unsupervised. The interaction between the two approaches and their respective importance in the process may be varied in numerous different ways but the idea is always the same : the supervised influence in the model guides the unsupervised learning, aiming to an understandable result.

Among the multiples approaches that exist, two are described below :

Supervised Iterative Clustering This method uses unsupervised learning to divide the data into cluster. Then, a supervised algorithm is used to determine whether those clusters separate correctly the data set into classes (e.g. each cluster contains 95% items of the same class), based on a known reference subset on the input (balanced training set). If not, it does another iteration of clustering on this element alone, etc. until all clusters satisfy the correctness condition. This trained model can then be used on the whole dataset.

Data Omission If a lot of data is available and it is dividable into a lot of classes, then we use a supervised algorithm to classify the data using only some of the known classes, keeping the others unknown, for a subset of samples. Once a satisfactory result has been reached, we use an unsupervised algorithm to speculate on classifying the rest of the classes.

1.2 Interpretability

Using *Fuzzy Logic* paradigms to construct a model has, as one of its main goal, to be understandable by a human being. On the opposite of *Machine Learning*, which models are basically black boxes from which we only expect a result, the scope of this project also aims for the model to be understandable in its execution and decisions. This is why, although we might need *Machine Learning* models as a base to our project, we **will** need to improve its interpretability by simplifying its structure and computing methods, *fuzzifying* the attributes, etc.

1.3 DataSet

The dataset is, at the beginning of this project, unknown to us. This implies our solution will have to be, at least to some degree, generic and adaptable to various types of data. We still expect to obtain a dataset with some known entries in order to add a supervised to the model. Depending on the approach though, those may not need to be equally distributed among the different desired classes.

2 Background

The literature and scientific background around clustering is well furnished to say the least. Clustering is a very wide area, belonging to the *classification* problem but with an *unsupervised* approach. The main advantages of unsupervised learning algorithms is that, as opposed to *supervised classification* algorithms, they do not require a well defined and balanced training set, hence it is often the chosen approach when confronted to a large set of data of which there is no good representative training set available. But there is no magic there and those advantages have a cost. The main downside of such an approach is that there is absolutely no control over the way the model ends up classifying the data, meaning although the results might be satisfying, it has to be used as is, with no re-usable information about *how* and *why* it ends up that way, except that it works.

Improving this issue is the main motivation behind *Fuzzy Clustering* methods and *semi-supervised clustering*. The first enables a better representation of the data and the available knowledge about it by allowing multi-valued membership grades and the insertion of *fuzzy rules* and *fuzzy classes*, which are way more understandable to us simple humans. The latter implies some human control over the direction taken by the algorithm in its clustering process as it will be described below which, again, improves the chances to get a result close to what we would expect.

2.1 Fuzzy c-Means Clustering

`Fuzzy c-Means Clustering` (FMC) is a clustering method which, unlike Hard k-Means Clustering, allows a data sample to more or less belong to one or more clusters. All elements belong to all clusters according to a certain *degree* and this value is called the *membership grade*, the matrix of all *membership grades* for given dataset of n elements and number c of cluster is noted as follow [1] [2]

$$U = \{u_{ki} \mid u_{ki} \in [0,1], \sum_{i=1}^{c} u_{ki} = 1,\ i = 1...c, k = 1...n\}$$

With this definition, we define the dataset X of dimension p as follow

$$X = \{x_k \mid x_k = (x_{k1}\ ...\ x_{kp})^T \in \mathbb{R}^p,\ k = 1...n\}$$

And the set of c clusters centers C as follow

$$C = \{c_i \mid c_i = (c_{i1}\ ...\ c_{ip})^T \in \mathbb{R}^p,\ i = 1...c\}$$

Using those definitions, we are now able to define the central concept of `FMC`, the unknown *objective function*. This function determines the relation between those 3 sets with a *fuzziness index* m ($\in \mathbb{N}^*$) and represents in a way the "efficiency" of the current model. It is the computation of the mean square error of the set of fuzzy clusters and it is used to iteratively compute new values for the *membership grades* matrix U and the clusters centers set C. It is defined as follow

$$J_m(U,C) = \sum_{k=1}^{n}\sum_{i=o}^{c} u_{ki}^m ||x_k - c_i||^2, i = 1...c, k = 1...n \tag{1}$$

The very goal of such a model is to find an optimal value for J_m starting from random *membership grades* values, which will have the consequence of finding an optimal clustering -optimal cluster centers with optimal membership grades- for our dataset X and it will be iteratively computed until such a value is reached. But how do we even know we reach it, and more importantly, how do we compute the next iteration ?
The answer is a bit of a math trick : let J_m^a be the optimal (i.e. minimal) value for our *objective*

function, reached at iteration a, and let us trace a curve describing the clustering efficiency for all $J_m^{(i)}$ iterations, then the curve would reach a local minima in J_m^a, which means

$$\frac{\delta J_m^a}{\delta c_i} = 0 \ , \ \frac{\delta J_m^a}{\delta u_{ki}} = 0$$

From this and without getting into the mathematical calculation (we fully trust our sources) to get there, we get the formulas to compute our values for the *membership grades* and the cluster centers :

$$u_{ki} = \Big[\sum_{j=1}^{c} \big(\frac{||x_k - c_i||}{||x_k - c_j||}\big)^{\frac{2}{m-1}}\Big]^{-1}, \qquad c_i = \frac{\sum_{k=1}^{n} u_{ki}^m * x_k}{\sum_{k=1}^{n} u_{ki}} \tag{2}$$

After each iteration, we compute the maximum difference between the new *membership grades* and the last ones, if this difference is under a predefined threshold ϵ, it means the system converges and the desired optimal value is reached, this condition is noted as follow

$$max_{ki}\{|u_{ki}^{j+1} - u_{ki}^{j}|\} < \epsilon, \epsilon \in [0, 1] \tag{3}$$

When this condition is met, C contains all the optimal cluster centers and U the optimal *membership grades* to all elements' correct cluster.

This algorithm, although being unsupervised, still presents a more understandable way of representing memberships for the samples by allowing more flexibility in their definition. Ironically, allowing more uncertainty -i.e. degree of membership instead of binary values- provides a more accurate description of the data, closer to our grasp of its meaning. The *membership grades matrix* contains way more information than the simple *membership matrix* a classical clustering algorithm would provide, by expressing this "uncertainty", meaning *fuzzy rules* and *fuzzy classes* can be derived from it.

So, *fuzzifying* a clustering model is a good start to answering the *how* stated earlier : given the resulting membership matrix U, we can interpret those grades to understand the final "decision" in the associated cluster to each entry and even get ourselves a set of rules describing the process in attributing a cluster to each data entry (by comparing those grades to the input X and the clusters choice for each one). This might seem like a great result, but there's no guarantee those rule would have any meaning to us : the criteria on which the clusters are formed might be really far from our way of seeing it (as mentioned earlier this is a common observation on clustering algorithms results). If those are too far from our perspective, we may extract rules from it, but those will be stripped from all semantic meaning and therefore not re-usable, nor interpretable.

2.2 Semi-Supervised Fuzzy c-Means Clustering

In order to have more control over the way the data may be divided into cluster, another approach is to allow a human to bias the process by providing information about class labels for some element or pairwise constraint between items (i.e. if some elements should belong to the same cluster). This control over a part of the clustering process enhance the understandability and practicability of the result, while still being efficient on large sets.[3] [1] [4]
In order to achieve this, the precedent algorithm is used, although slightly modified. The *objective function* uses another parameter, a `supervised membership grades` matrix, described as follow :

$$\overline{U} = \{\overline{u}_{ki} | \overline{u}_{ki} \in [0, 1], \sum_{i=1}^{c} \overline{u}_{ki} \leq 1, i = 1...c, k = 1...n\}$$

The modified *objective function* is then described as follow

$$J_m(U, C) = \sum_{k=1}^{n}\sum_{i=o}^{c} |u_{ki} - \overline{u}_{ki}|^m ||x_k - c_i||^2, i = 1...c, k = 1...n \tag{4}$$

This equation is then derived to obtain the calculation formulas for the *membership grades* and the cluster's centers

$$u_{ki} = \overline{u}_{ki} + (1 - \sum_{j=1}^{c} \overline{u}_{kj}) \frac{(\frac{1}{d_{ki}})^{\frac{1}{m-1}}}{\sum_{j=1}^{c} (\frac{1}{d_{kj}})^{\frac{1}{m-1}}}, d_{ki} = ||x_k - v_i||^2, m > 1 \tag{5}$$

$$v_i = \frac{\sum_{k=1}^{n} |u_{ki} - \overline{u}_{ki}|^m x_k}{\sum_{k=1}^{n} |u_{ki} - \overline{u}_{ki}|^m} \tag{6}$$

Again, we compute after each iteration the maximum grade difference between this one and the last using equation (3). Once this value is under the predefined ϵ, the system is considered to be converging and the optimal results reached.

This approach requires more upstream work and prior knowledge about the data in order to be able to define pairwise constraints and/or some of the *membership grades*. The more and the most accurate the better, as those values will remain unchanged and used as a hard reference or constraint all along the process of learning. This may require some expertise and time to obtain, which could seem as constraining as *supervised classification*. Still, this approach differs from *supervised classification* in the fact that it doesn't require any constraint on known data : pairwise constraint are all self-sufficient, all can influence independently the clustering process and predefined *membership grades* may concern only a limited number of classes meaning they don't need to be equally distributed over the number of desired clusters to be efficient.

To summarize, all the advantages of *unsupervised clustering* are still provided by this approach (i.e. applicable to large sets of mostly unknown data while still being very efficient) and combined with those of *supervised classification* (i.e. getting a result close to what is expected and known) without either downsides (i.e. non-interpretable results for the first and very constraining data requirements for the latter).

2.3 Clustering Quality Control

In order to understand whether the results provided by the *FCM* are good, several indices have been used to measure the quality of the clustering. *Clustering Quality Indices* (CVI) may concern different aspects of the result : the separation between the clusters, their compactness, etc. In the scope of this project, aiming toward a semi-supervised solution, both supervised and unsupervised CVI have been used.

2.3.1 Unsupervised CVIs

In our words, *unsupervised CVI* are indices that use unlabelled data along with the results provided by the FCM algorithm : the resulting *membership-grade matrix*, the set of cluster's centers and the *fuzzy parameter*. Thus, no information about the known data is provided, the result is then completely oblivious to prior knowledge about the data, hence the *unsupervised* characterization used above.
From the very large set of existing indices, 3 have been chosen as they seemed to be pertinent in the context of ***Fuzzy*** *Clustering*, those indices are the following[5] for x a set of N samples, v the set of c cluster's centers, u the membership-grade matrix of dimension Nxc and the *fuzzy parameter* m :

Fuzzy Hyperbolic Volume , which measures the clusters *compactness*, :

$$FHV(m, u, x, v) = \frac{\sum_{i=1}^{N} u_{ij}^m (x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^{N} u_{ij}^m}$$

Partition Coefficient , which measure, for each cluster and given the *membership-grade matrix* how "decisive" those values are : the closer the grades are between cluster for each sample,

the smaller the value is, best score is 1 :

$$PC(u, N, C) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \cdot log_a(u_{ij})$$

Xie-Beni Index measures the separation between the cluster along with their compactness[6] :

$$XB(m, u, x, v) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^2 ||x_i - v_i||^m}{N min_{t\neq s}\{||v_t - v_s||^m\}}$$

2.3.2 Supervised CVI

To insert some supervised influence in the quality control of intermediary results, it seemed pertinent to use the known labels from the dataset in order to evaluate an estimate of the uniformity of each cluster i.e. whether a given cluster contains mostly the same class of samples, or not. On the contrary of the CVIs mentioned in the previous paragraph, this index is an arbitrary decision based on our estimation of what *being a good cluster* means. It has been decided to "accept" a cluster if 95% of the known samples it contains are of the same class.

In that respect, after each iteration of the FCM algorithm, each cluster is scanned and over the set of known sample it contains, a simple class-wise count is done and used then to obtain an estimation of the cluster's proportion inside it. If one of those is above 95%, the cluster is deemed acceptable.

For a set c of n clusters, a set C of m classes, and x the set of known samples :

$$sup_{cvi}(x, c, C) = \begin{vmatrix} \frac{\#\{x\in c_0 \wedge x\in C_0\}}{\#\{x\in c_0\}} & .... & \frac{\#\{x\in c_0 \wedge x\in C_m\}}{\#\{x\in c_n\}} \\ ... & ... & ... \\ \frac{\#\{x\in c_n \wedge x\in C_0\}}{\#\{x\in c_n\}} & .... & \frac{\#\{x\in c_n \wedge x\in C_m\}}{\#\{x\in c_n\}} \end{vmatrix}$$

With both those types of indices it is expected to take full advantage of the both the supervised and unsupervised parts of the algorithm by selecting the clusters based on their estimated quality (based on known samples) while still not depending too much on those value by considering the whole dataset clustering quality (using *unsupervised* CVIs) at each iteration - which is expected to reduce the impact of potentially unbalanced references among the classes.

3 Solution Conception

During the exploration of potential solutions, several major alterations have been made on the original decomposition. Indeed, the use of a *supervised membership-grade matrix*, despite providing numerous advantages, requires knowledge of the number of desired clusters beforehand : not only to estimate the correct number of cluster (required to define this matrix) but also to have specific pre-defined *fuzzy* membership grades for known entries. Thus, implementing such a model would require not only a deep study of each dataset in order to determine pertinent *membership-grades* for some known values, but also to begin with an exploration of the optimal number of cluster and only then defines the supervised matrix $\overline{U}$ corresponding to this parameter.
Hence, our knowledge of the dataset (i.e. known samples) being binary : every entry is known to belong to a certain class, and the optimal number of cluster being originally unknown to us, this projects constraints wouldn't take full advantage of this approach.

Thereupon, it has been decided to completely renounce to this model and focus on a more "iterative" supervision. Continued research about clustering methods lead the solution in a completely different approach : *cluster validity measures*. Using both tools from unsupervised models and the advantages of having a partially labeled data, several indices have been used and defined in order

to keep the algorithm between supervised and unsupervised approaches, expecting to derive the best from both. The resulting problem definition is as follow:

Unsupervised Model - Determine an appropriate *Fuzzy c-Means Clustering* model to use as a basis to our project. This model includes *unsupervised CVIs* to estimate the clustering quality based on results and unlabeled data only.

Supervise It - Introduce a new *Cluster Validity Index* based on known samples and use it to construct a new algorithm based on iterative cluster selection based on its results.

Experiment - Explore the different parameters of the algorithm : proportion of known samples in the dataset, fuzzy parameter, indices thresholds, etc.

Final Model - Use previous results to determine a final model with optimal parameters.

Rule Extraction - From the results of the optimal model, a set of rule (e.g. Decision Tree) is extracted in order to construct a generalized predictor for the given type of data. This can then be derived into a *Fuzzy Inference System*, providing a very much enhanced interpretability factor from the clustering results.

From this decomposition, we defined a general conception of our solution, which will be described in the next paragraphs. We consider a textual dataset (e.g. *csv* format) and our whole implementation will be in *Python3*, using the wonderful *Jupyter Notebooks* format to offer a familiar and good looking visual interface.

3.1 Basis Model

The base for our project is a homemade (although greatly inspired from here) implementation of a *Fuzzy c-Means Clustering* algorithm and some *CVIs* from which some will show to be useful in the future model conception. Given a text-format dataset X, a desired number of cluster c and a divergence threshold ϵ, it provides a result in the form of a *membership grades matrix* U, a cluster's center vector V and the whole labelized dataset X'.

3.2 Supervision Inclusion

From this model, the whole *FCM* algorithm is taken, along with three *Clustering Validity Indices*, considered pertinent in the scope of this project. Those 3 indices are the ones described in the previous section. The new *supervised* index sup_{cvi} is implemented as described in the previous section. From those elements, two algorithms are defined and explored in parallel.

3.2.1 Full-Iterative FCM

Algorithm 1: Full-Iterative FCM

Data: Dataset X with labelised subset X_c, Fuzzy parameter m, Labels names L
Result: Labels for input data X_L, Membeship-grade matrix U, Clusters centers V

begin
- c = length(L);
- *done* = False;
- `/* Membership-grade Matrix */`
- mb = [ ];
- `/* Cluster Centers Array */`
- ctrs = [ ];
- **while** $\neg done \wedge c < \sqrt{length(X)}$ **do**
  - `/* Run FCM on data */`
  - `/* X[n] contains label values */`
  - $mb, ctrs = FCM(X[x_0, ..., x_{n-1}], c, m)$
  - `/* Defuzzy membership grades */`
  - `/* to get each entry's cluster */`
  - $labels = getClusters(X[x_0, ..., x_{n-1}], mb)$
  - $idx = sup_{cvi}(X, labels, c, length(L))$
  - `/* If all cluster are deemed homogeneous based on` sup_{cvi} `*/`
  - **if** *all index in idx are ok* **then**
    - done = True;
  - **else**
    - `/* Else restart with more clusters */`
    - c = c + 1;
- `/* Extract class for each cluster, based on known labels */`
- $X_L = getClass(X, labels, length(ctrs), L)$
- U = mb;
- V = ctrs;

This algorithm aims to obtain the optimal number of cluster (and the best clustering) iteratively by starting with a number of cluster equal to the number of known classes in the dataset.
From this value, it runs the *FCM* algorithm on the those data (without the labels column, although many may be unknown) to get from it a resulting *membership-grade matrix* and a set of cluster's centers. With the matrix, each entry's rightful cluster is determined and the sup_{cvi} index defined previously is used to obtain, for each cluster, the proportion of each class in it.

If all clusters contain a single class above a predefined threshold (e.g. 0.95), the clustering is deemed acceptable and the algorithm stops its iterations. From the clusters memberships and the known samples, the algorithm assigns each cluster (and the samples in it) a certain class (based on mostly present class), and then returns the current *membership-grade matrix*, the current clusters' centers and the final labels for the input.

This approach only inserts a supervised influence in the termination of the algorithm. Indeed, the clusters themselves are assessed as a whole and all have to be good to be accepted, all are rejected if one isn't good enough. Hence, a cluster itself is formed without any supervision, only the number of cluster depends on supervised factors.

3.2.2 Selective FCM

Algorithm 2: Selective FCM

Data: Dataset X with labelised subset X_c, Fuzzy parameter m, Labels names L
Result: Labels for input data X_L, Membeship-grade matrix U, Clusters centers V

begin
- c = length(L);
- *done* = False;
- `/* Membership-grade Matrix */`
- mb = [ ];
- `/* Cluster Centers Array */`
- ctrs = [ ];
- `/* Remaining Entries */`
- X_{temp} = [X]
- `/* Results */`
- $X_F = []; V_F = []; M_F = []$
- **while** $\neg done \wedge c < \sqrt{length(X)}$ **do**
  - `/* Run FCM on data */`
  - *mb, ctrs = FCM(*$X_{temp}[x_0, ..., x_{n-1}], c, m$*)*
  - `/* Defuzzy membership grades */`
  - `/* to get each entry's cluster */`
  - *labels = getClusters(*$X_{temp}[x_0, ..., x_{n-1}], mb$*)*
  - $idx = sup_{cvi}(X_{temp}, labels, c, length(L))$
  - `/* If no cluster is deemed homogeneous based on` sup_{cvi} `*/`
  - **if** *no index in idx are ok* **then**
    - c = c + 1;
  - **else**
    - `/* Extract good clusters */`
    - C_{ok} = {$c \in$ ctrs if $0.95 \in$ idx[c]};
    - V += C_{ok}
    - X_F+ = {$x \in$ C | C $\in C_{ok}$}
    - M_F+ = {u_x | x $\in X_F$}
    - X_{temp}− = {$x \in$ C | C $\in C_{ok}$}
    - `/* Compute Unsupervised Indices to estimate whole clustering quality */`
    - $unsup_{cvis} = getUnsupIndices(X, mb, ctrs, m)$
    - `/* If clustering is good, add rest of elems and terminate loop */`
    - **if** $unsup_{cvis}$ *is ok* **then**
      - done = True;
      - addRestElem(X_{temp}, mb, ctrs);
    - **else**
      - c = c - length(C_{ok}) + 1
- `/* Extract class for each cluster, based on known labels */`
- $X_L = getClass(X_F, labels, length(V_F), L)$
- U = M_F
- V = V_F

Unlike *Algorithm 1*, this other version uses both *unsupervised* and supervised indices to evaluate the results obtained by the FCM. This approach starts by evaluating each cluster based on the sup_{cvi} index in order to determine in a supervised way which cluster to keep. Every cluster deemed acceptable by this function is kept as is and will be part of the result. This algorithm execution is as follow :
Start with empty array for result and copy the input data into a temporary variable, defined the number of cluster as the number of known classes and start the execution loop. Run the *FCM* algorithm with those parameters and extract the entries cluster membership from the resulting *membership-grade* matrix. Using sup_{cvi}, control each cluster's homogeneity, if no cluster matches the required threshold for any class, start over with an incremented number of cluster.

If one or more clusters are considered good enough, add their centers and the elements they contain, along with the entries' corresponding membership grades to the result arrays and remove those entries from the temporary dataset. Update the number of cluster to be passed to the FCM so that the remaining data is split in one more way than it was (e.g. for 2 clusters accepted out of 4, the remaining data will be split in 3 clusters). Then, use *unsupervised indices* to evaluate the global performances of the FCM based on cluster separation, compactnes, etc. If those indices present very good values (above a predefined threshold), accept the whole clustering as a satisfying solution and terminate the loop.

This approach tries to include more supervision in the final clusters definition (by keeping every good cluster) whilst trying to limit its dependency on the known dataset by using unsupervised indices as another measure of the clustering quality. Hence, keep making the best of both supervised and unsupervised models despite an increased supervised influence.

Using both those algorithms, the objective is to get a better grasp of the effect of supervision inclusion around an unsupervised clustering model and how it influences both result performances (e.g. accuracy) and interpretability (e.g. number of cluster).

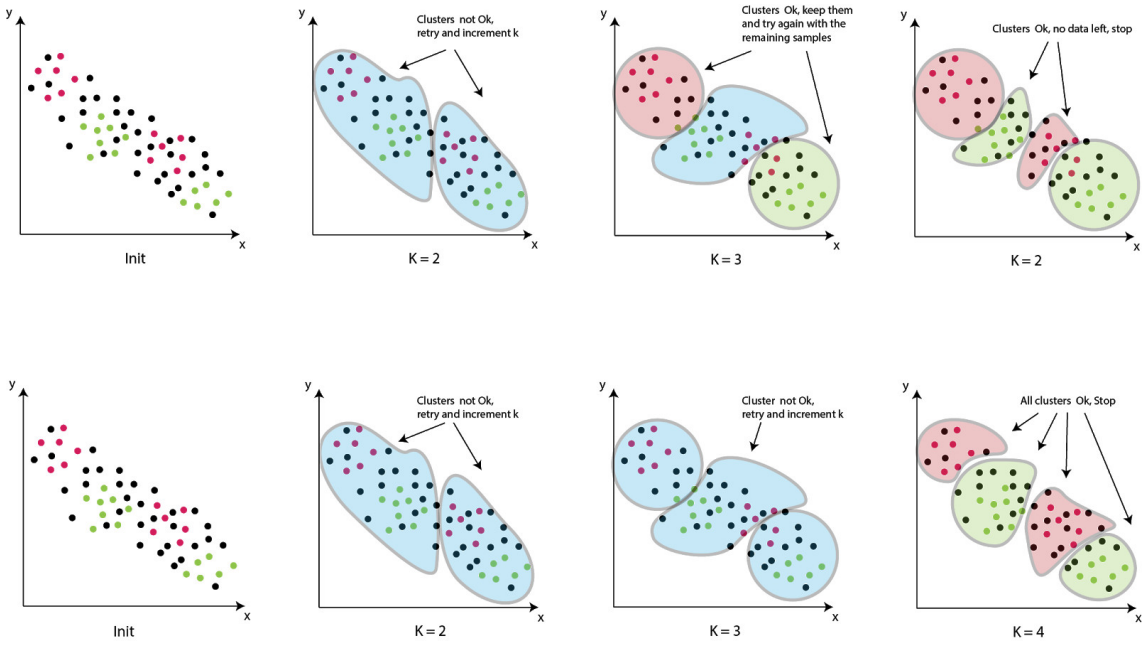

Figure 1: Schema illustrating the S-SCM (first row) and the FI-FCM (second row) algorithms execution

3.3 Experiment

In order to fully explore most aspects of our solution, we decided, for both algorithms to proceed to following way (each step in then justified by a short explanation):

1. Take a fully known labeled dataset
2. Randomly remove a given percentage of labels
3. Explore single run performance :
   - Run algorithm with given initial parameters (e.g. 10% of known labels, $fuzzy\_param = 2$, $membership\_threshold = 0.95$)
   - Measure Accuracy, F-1 score, Precision and plot Confusion Matrix
4. Explore the impact on accuracy of $fuzzy\_param$ and $membership\_threshold$ variations for the same share of known entries :
   - Measure Accuracy and plot 3D graph to show evolution
5. Explore the impact on accuracy and final number of cluster of variations in the share of known labels :
   - For all value i between 0 and 100, remove i% of labels in the dataset
   - Run FCM and measure predictions accuracy
   - Note number of cluster and plot comparative graph

Step 1 Starting from a fully known dataset allows all combination of *labeled/unlabeled* entrier in the dataset. Thus enabling an exploration on this dimension also.

Step 2 is intended to provide random information about the dataset to the algorithms, ensuring thus reliable results (i.e. independent from the combination of known entries for a given percentage)

Step 3 helps to get a first idea of what to expect, performance-wise, from our models.

Step 4 is expected to provide some useful informations about the impact of the "severity" of our sup_{cvi} index with respect to the "fuzzyness" of the FCM whilst providing insight about global performances compared to both factors.

Step 5 uses previously found optimal parameters to gain some insight about the interpretability of the result by measuring the number of resulting clusters, which is a decisive aspect of the solution in view of extracting interpretable results from the models and defining the limits of this approach.

3.4 Interpretability Improvement

Once the rest is up and going, it is expected to obtain an optimal clustering with both maximal accuracy and minimal number of cluster for given optimal parameters (known data share, *fuzzy_parameter*, *membership_threshold*). This result can then be used to create a *Boolean Inference System* as follow [7] :

1. Create a decision tree based on the result obtained from our best model and the input dataset
2. Translate transitions arrow and discriminating features into rules
3. Create a *Boolean Inference System* using those rules
4. Optimize those Fuzzy rules using concepts seen in class (syntactic and semantic criteria)

Step 1 should provide a tree with the minimal number of data features involved in the decisions and the optimal values to discriminate the entries as it uses the resulting data labeling. The number of node in the last layer is the final number of cluster.

Step 2-3 is about a *boolean* system as the *Decision Tree* provides binary decision transitions and not *fuzzy* values, this aspect will be further discussed in the **Discussion** section.

Step 4 requires human knowledge to reduce even more the complexity of the Decision Tree. The result should be a comprehensible and optimal *class predictor* for this type of data.

4 Result Analysis

In this section are presented the results from the aforementioned experiment. The first two paragraphs present both models results separately, the last paragraph show the result from the comparison between the two.
The dataset used for those experiments is "Leukemia_Mile_210", it was chosen because of the large number of samples (≈ 2000) it contains and the striking results it lead to, illustrating best our models properties, similarities and differences (cf. Figure 8).

4.1 Full Iterative Fuzzy C-Mean

Illustrative Confusion-Matrix for 10% of known entries, *membership_ threshold* of 0.95, and *fuzzy parameter* of 2

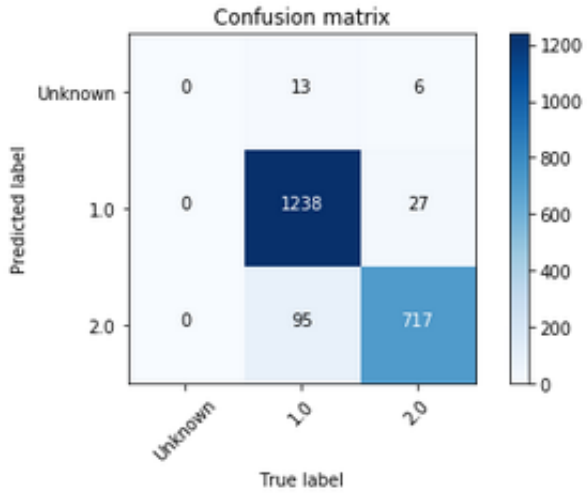

Figure 2: Confusion Matrix of FI_FCM, accuracy = 0.93

The *unknown* corresponds to elements belonging to a cluster the algorithm could not decide which class to assign (same number of known elements for both classes).
We see most elements are in the diagonal positions, with a total of 1955 elements.
We also notice that for a total of 19 entries, the algorithm was not able to determine a class.
Finally, we see that a total of 122 entries have been mislabeled by the algorithm.

3D Plot of Accuracy depending on *membership threshold* and *fuzzy parameter* variation with 10% of known entries

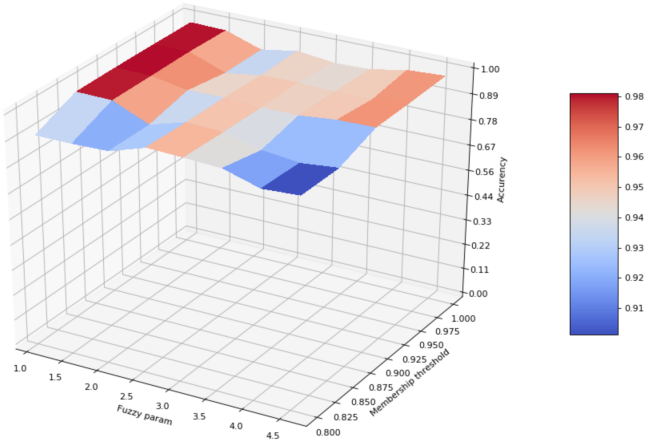

Figure 3: FI-FCM accuracy depending on the membership threshold and the fuzzy parameter

The figure 3 shows the accuracy rate obtained by varying the *fuzzy parameter* from 1 to 5 and the *membership threshold* from 0.8 to 1.0.

We observe that the measured accuracy never goes below 0.9, with a maximal value of 0.98. We also see that the plot doesn't show much variation among the different parameters value.

Finally, we notice that the best accuracy rate is reached with a *fuzzy parameter* of 1.5 and a *membership threshold* 0f 0.925.

2D Plot with accuracy and number of cluster depending on percentage of unknown entries with *fuzzy parameter* = 1.5 and *membership threshold* = 0.925

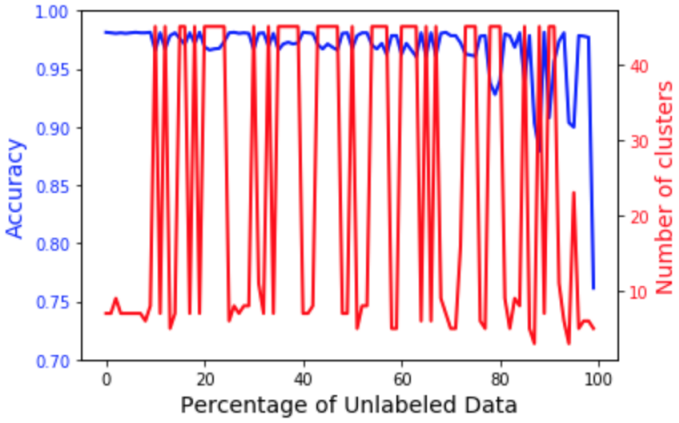

Figure 4: FI-FCM Accuracy depending on the percentage of unlabeled data

The figure 4 shows the accuracy rate and the number of cluster evolution with respect to the share of unknown labels in the dataset. We see that the accuracy stays steadily above 0.98 until 80% of unknown labels. From this point it varies much more, reaching its minimum of 0.75 for absolutely no known labels.
Concerning the number of cluster, we can see that it varies a lot between ≈ 8 and 45 with almost no values in between.

Finally, we notice that for most of the evolution, accuracy loss is often associated with a high number of cluster.

4.2 Selective Fuzzy C-Mean

Illustrative Confusion-Matrix for 10% of known entries, *membership_ threshold* of 0.95, and *fuzzy parameter* of 2

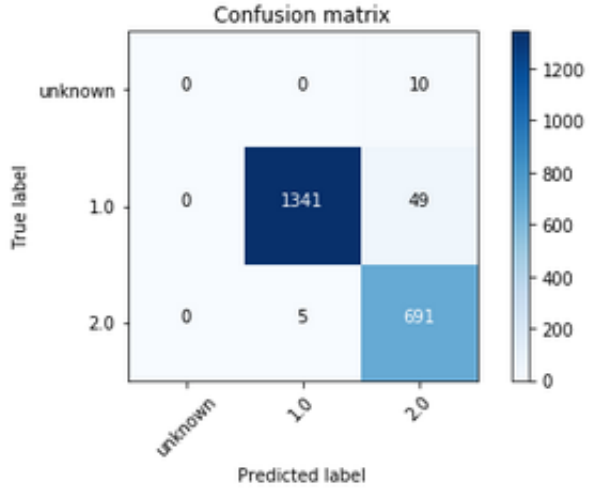

Figure 5: Confusion Matrix of S-FCM, accuracy = 0.97

We see most elements are in the diagonal positions, with a total of 2031 elements.

We also notice that for a total of 10 entries, the algorithm was not able to determine a class.

Finally, we see that a total of 54 entries have been mislabeled by the algorithm.

3D Plot of Accuracy depending on *membership threshold* and *fuzzy parameter* variation with 10% of known entries

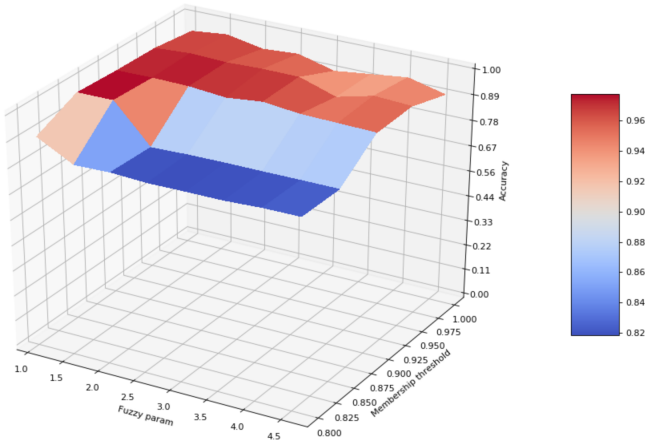

Figure 6: S-FCM Accuracy depending on the membership threshold and the fuzzy parameter

The figure 6 shows the accuracy rate obtained by varying the *fuzzy parameter* from 1 to 5 and the *membership threshold* from 0.8 to 1.0.

We observe that the measured accuracy never goes below 0.8, with a maximal value of 0.98. We also see that the plot doesn't show much variation among the different parameters value except a slight accuracy loss for low *membsership threshold* values ($<$ 0.9).

Finally, we notice that the best accuracy rate is reached with a *fuzzy parameter* of 1.5 and a *membership threshold* of 0.9.

2D Plot with accuracy and number of cluster depending on percentage of unknown entries with *fuzzy parameter* $= 1.5$ and *membership threshold* $= 0.9$

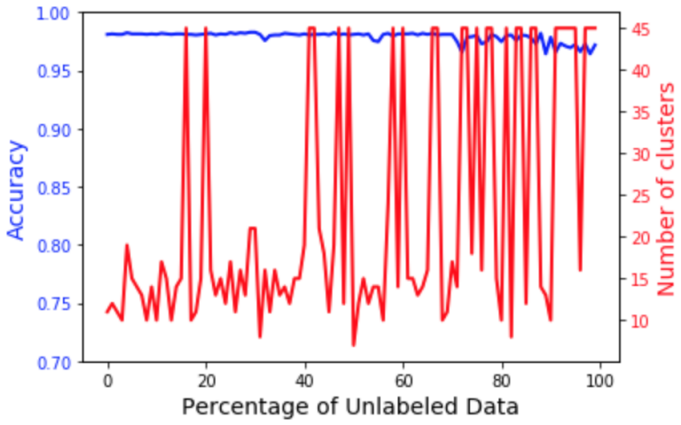

Figure 7: S-FCM Accuracy depending on the percentage of unlabeled data

The figure 7 shows the accuracy rate and the number of cluster evolution with respect to the share of unknown labels in the dataset. With this parameter we computed the second test, the result is quite impressive. While the number of clusters needed to classify is similar to the FI-FCM, the accuracy is very steady, no matter how many labels we have, it stays around 0.98. Let's not forget that even if we have 98% of unlabeled data, the dataset is big so we still have more than 20 labels of each class to guide our algorithm.

4.3 FI-FCM versus S-FCM

In order to compare the performances of our models, we decide to test them on four different datasets. We chose datasets with very different main properties, in order to highlight the pros and cons of each algorithm. The properties of each dataset are given in Figure 8

id	name	size	classes	labels balance	source
0	melanomaTest	98	2	[51, 46]	CIBCB & BBTC
1	seeds	199	3	[66, 68, 64]	UCI
2	CancerDiag2	208	2	[141, 66]	Professor Carlos Andrés Peña
3	Leukemia_Mile_210	2096	2	[1345, 750]	Professor Carlos Andrés Peña

Figure 8: Datasets properties

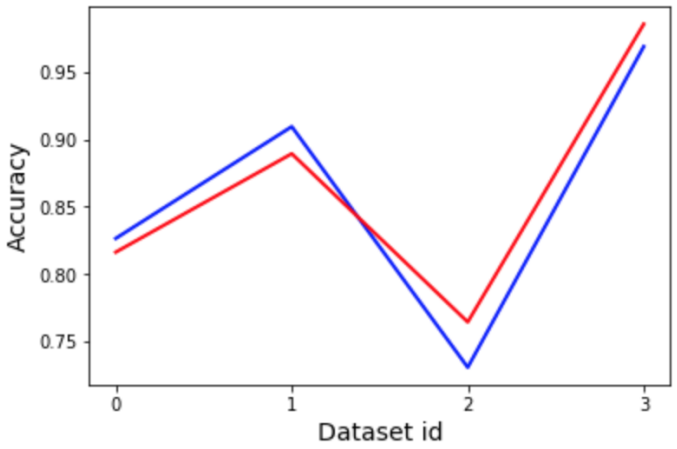

Figure 9: Accuracy comparison between FI-FCM & S-FCM on four datasets

The figure 9 shows the accuracy rate reached by both algorithms of the 4 different datasets with 90% of unknown labels, the red curve corresponds to *S-FCM* and the blue curve to *FI-FCM*

We can see that both algorithms accuracy performances vary much among the different input data, with a minimum of ≈ 0.73 and a maximum of ≈ 0.98.

Furthermore, we notice that despite reaching very close results, *FI-FCM* performs better on the first two, while *S-FCM* takes the lead on the last two dataset.

5 Discussion

5.1 Accuracy Performances Volatility

The Figure 9, shows us that the performances of both algorithms are quite volatile depending on the dataset, with a quick look to the figure 8, describing the dataset we can interpret those results as follow :

melanomaTest : small size but well balanced dataset among 2 classes with rather blurry distinction between those. We could interpret a lower accuracy rate (compared to other found values) as the direct consequence of the small number of samples : those being mostly unlabeled, the number of known entries per class is most likely around 4-5, which might explain some lack of precision in the labels prediction.

seeds : slightly bigger size, well balanced dataset among 3 classes with rather distinctive values between the classes. In this case, despite a still rather small number of samples, we observe a quite satisfying accuracy rate. This could be explained by the classes being more distinguishable from on another. Hence, although there might only be at best ≈ 6 known entries per class, the algorithm seems to find a good way to predict labels.

CancerDiag2 : still pretty small data set with unbalanced number of entries per class and rather blurry distinction between those. This dataset is the one obtaining the worst results and it seems to combine all the characteristics that could potentially lower those performances : inequal representation of the classes combined with a very low number of know entries for each (at best ≈ 14 and ≈ 6). This means the supervised index could be misguided by a lack of reference (potentially at most one known entry per cluster) along with the clustering itself that might suffer from the lack of distinctive values between the classes.

Leukemia_Mile_210 big size, unbalanced between the 2 classes but both seem rather distinct from one another. This set obtains by far the best accuracy performances of all 4 dataset. This could be explained by both a high number of entries, meaning each class could at best have ≈ 130 and ≈ 75 known samples, which might be more than enough for the sup_{cvi} index to correctly estimate clusters validity. Furthermore, good class distinction might explain how the *S-FCM* algorithm is able to keep an accuracy rate above 0.95 even with 0 known entries with the help of the 3 *unsupervised* indices that indicate an estimate of the clustering validity with no conception of validity whatsoever.

5.2 Clustering

We saw in Figures 4, 7 that both of our solutions produce sometimes large number of cluster to classify the datasets, and sometimes around 8-9 clusters. Noticing that it is most often either below 15 or equal to 45, 45 being the maximal number of cluster allowed for this dataset ($\sqrt{2096} = 45.78$). This maximal value can only be reached if no iteration produced a clustering deemed acceptable.

For *FI-SCM*, it means no clustering was ever only composed of homogeneous enough clusters. Considering this measure is based exclusively on pre-labeled entries, it is not surprising to note not only widely variable cluster numbers, but also mostly "extreme" values. Indeed, the random removal of label done on the original dataset might affect result in a "lucky" or "unlucky" distribution of the remaining known labels. Then again, this distribution is a double-edged sword, meaning here that if most cluster do not contain any known labels, then the number of cluster will be increased but the quality estimation of the other clusters might be more realistic. On the other hand if most clusters contain some labels, then each contain less and the probability that this label is not representative of the other members of the same cluster is increased.

Concerning the *S-FCM*, the issue explained above still holds, but might be compensated by the use of the other *CVIs*, oblivious to the labeling. Furthermore, the choice to accept any "good" cluster at each iteration increase the chance of a more indicative measure by sup_{cvi} : the first algorithm is more likely to reject correctly estimated clusters as it requires all of them to be good to do so. This algorithm keeps any good candidate, giving the remaining sample another "chance" to re-order in a more favorable way.

On the other hand, depending on 3 *unsupervised* this way is not ideal. Most *CVIs* don't have an absolute threshold to indicate a good enough clustering and are more often used in preliminary parameter exploration to find the best value among a large set of experiment on the optimal cluster number. It is then realistic to assume that their usage in the scope of this project clearly applies constraint on the result, considering an arbitrary value has been chosen as a threshold, necessary to the clustering acceptation. Meaning it is often not realistically reachable and "forces" an increase on the number of cluster.

5.3 Interpretability

The effort made to improve the interpretability of our algorithms results began by guiding our solution toward a minimal number of cluster. Hence, some compromise has to be found between the "severity" of the cluster's evaluation and the accuracy requirements.

From this result, this project tried to extract a *Decision Tree* from the obtained set of predicted labels by both *S-FCM* and *FI-FCM* (using the **sklearn** python library). This tree can then be *mined*[7] in order to extract rules from it.

In the scope of this project, the process saw an end with binary decision, resulting in a *boolean* predictor (illustrated in figure 10), another approach will be introduced in the next section.

This result, although being subject to boolean logic limitations can still be used as a predictor. Besides, being obtained from a *fuzzy* algorithmic model, this result still reflects an improved interpretable decision making system on a reduced number of feature.

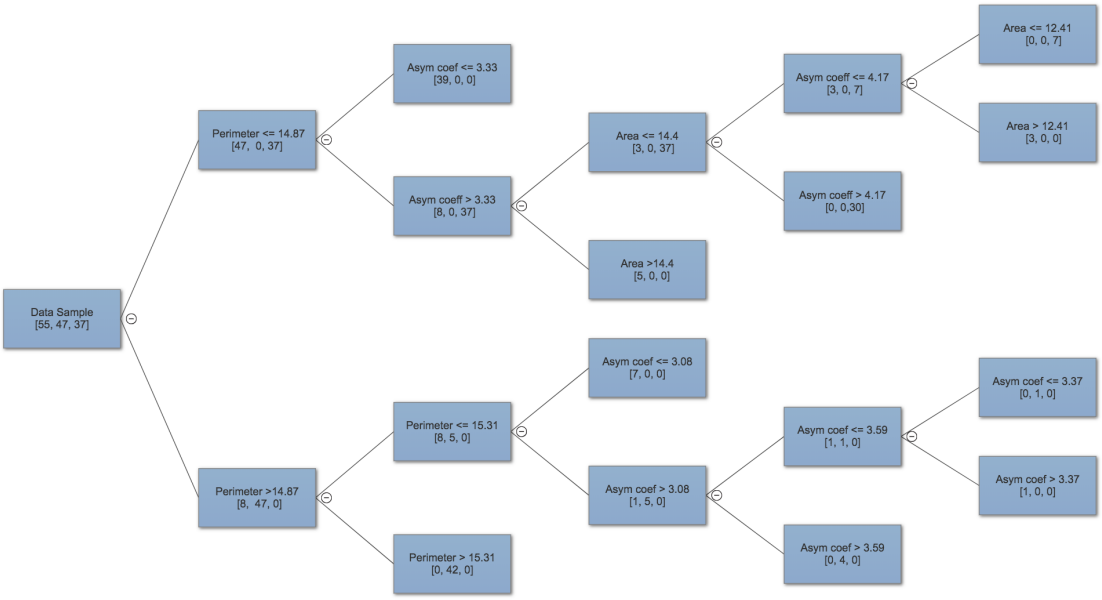

Figure 10: Decision tree obtained for the seeds dataset with S-FCM

5.4 Future work

From the statements of the previous section, it seems natural to contemplate a process of *fuzzy rule* extraction via supervised optimization of fuzzy clustering.

A more thorough and exhaustive exploration of the *CVIs* usage and their respective influence on the clustering iterative process might combine the already verified accuracy of this approach with a more stable clustering result.
Furthermore, it seems within reach to use this algorithm is such a way that a pertinent final *membership-grade* matrix could be constructed and then used along the *Decision Tree* to construct a coherent and precise *Fuzzy Inference System* based on the algorithms results.

6 Conclusion

This project mildly started to pave the way to a very interesting way of extracting *Fuzzy Inferences Systems* from any given dataset by a *supervised* optimization of an otherwise *unsupervised* but *fuzzy* clustering process. The observed tolerance to high variation in the reference sub-set accuracy-wise and globally very satisfying results clearly suggests this approach might be interesting to answer the many problems involving fuzzy decision making combined with mostly unknown, variable data.

BLABLABLA ENCORE LACHER UN MOT

References

[1] T. H. Thong and L. H. Son, "Semi-Supervised Fuzzy Clustering with Pairwise-constrained Competitive Agglomeration,"

[2] P. Milano, "A Tutorial on Clustering Algorithms,"

[3] T. H. Thong and L. H. Son, "An Overview of Semi-Supervised Fuzzy Clustering Algorithms,"

[4] Y. M. M. S. ENDO Yasunori, HAMASUNA Yukihiro, "On Semi-Supervised Fuzzy c-Means Clustering,"

[5] M. Halkidi and al., "Clustering validity checking methods: Part ii,"

[6] Q. ZHAO, "Cluster validity in clustering methods,"

[7] H. Khosravi and al., "FCM-Fuzzy Rule Base: A new Rule Extraction Mechanism,"