A New Semi-Fuzzy Algorithm for Cluster Detection and Characterization

Hanane Bennachid, Rokia Fajr and Abdelaziz Bouroumi
Modeling and Instrumentation Laboratory, Ben M’sik Faculty of Sciences,
Hassan II Mohammedia Casablanca University (UH2MC), Casablanca, Morocco

Abstract: Researchers propose a new algorithm for detecting homogeneous clusters within sets of unlabeled objects represented by numerical data of the form \( X = \{x_1, x_2, ..., x_n\} \subseteq \mathbb{R}^d \). By quickly exploring the available data using an inter-objects similarity measure plus an ambiguity measure of individual objects, this algorithm provides the number of clusters present in \( X \), plus a set of optimized prototypes \( V = \{v_1, v_2, ..., v_c\} \subseteq \mathbb{R}^d \) where each prototype characterizes one of the \( c \) detected clusters. The performance of the algorithm is illustrated by typical examples of simulation results obtained for different real test data.

Key words: Cluster analysis, unsupervised learning, fuzzy clustering, pattern classification, Morocco

INTRODUCTION

Clustering multidimensional data is a very hard problem that is commonly encountered in a variety of real-world applications related to different fields of activity including industry, medicine, telecommunication, economics, etc. In terms of optimization, this problem consists in finding the best way to organize a set of \( n \) unlabelled vector objects \( X = \{x_1, x_2, ..., x_n\} \subseteq \mathbb{R}^d \) into \( c \) homogenous groups or clusters such that objects of each group should be as similar as possible and objects belonging to different groups as dissimilar as possible. The number of clusters \( c \) can be either fixed a priori as an input of the problem or algorithmically determined or learned from the available data.

In pattern classification and recognition which is one of the most known fields of application of clustering, this problem is also known under the name of unsupervised learning where the word learning refers to the determination of class characteristics and probably also the number of classes from the set of available data which is viewed as a learning database (Jain and Dubes, 1988).

In traditional clustering, classes are supposed to be well separated and each object is assigned to one and only one class. Fuzzy clustering is a generalization of traditional clustering that presents the advantage of dealing with situations where this supposition cannot be accepted because boundaries among classes are not clear and objects close to these boundaries seem to verify at different degrees, the conditions of belonging to more than one cluster (Zadeh, 1965). This generalization is based on the introduction of the concept of membership degree \( u_k \in [0, 1] \) which can also be interpreted as the degree to which the object \( k \) verifies the condition of belonging to the class \( i \) \((1 \leq k \leq n \text{ and } 1 \leq i \leq c)\). Hence, instead of assigning each object to a unique class, fuzzy clustering assigns a vector of \( c \) class-membership degrees to each object (Bouroumi et al., 2000).

More formally, a partition of \( X \) into \( c \) fuzzy clusters can be defined as a \((c \times n)\) matrix \( U = [u_{ik}] \), \( 1 \leq c, 1 \leq k \leq n \) which verifies the three following constraints:

\[
0 \leq u_{ik} \leq 1 \quad (1)
\]

\[
\sum_{i=1}^{c} u_{ik} = 1, \quad 1 \leq k \leq n \quad (2)
\]

\[
0 \leq \sum_{k=1}^{n} u_{ik} \leq n, \quad 1 \leq i \leq c \quad (3)
\]

The first constraint is evident because \( u_{ik} \) measures the degree to which the \( k \)th object satisfies the condition of belonging to the \( i \)th class. It is a fuzzy generalization of the classical constraint \( u_k \in \{0, 1\} \) where the condition for the \( k \)th object to belong to the \( i \)th class is either true (\( u_{ik} = 1 \)) or false (\( u_{ik} = 0 \)). The second constraint ensures that the membership of each object is distributed only over the \( c \) clusters and the constraint ensures that no cluster is empty (\( \sum_{i=1}^{n} u_{ik} = 0 \)) or equal to its entirety (\( \sum_{i=1}^{n} u_{ik} = n \)).

Given such a membership matrix \( U \) it is easy to define a representative or prototype for each class \( i \) using the expression:

Corresponding Author: Hanane Bennachid, Modeling and Instrumentation Laboratory, Ben M’sik Faculty of Sciences, Hassan II Mohammedia Casablanca University (UH2MC), Casablanca, Morocco

191
\[ v_i = \frac{\sum_{k=1}^{n} (u_k)^m x_k}{\sum_{k=1}^{n} (u_k)^m}; \quad 1 \leq i \leq c \]  \hfill (4)

where, \( m > 1 \) is a weighting exponent aimed at controlling the relative contribution of each object \( i \) according to its membership degree to the class \( i \), to the calculated prototype \( v_i \). This expression can be viewed as a fuzzy generalization of the classical barycentre or mean-vector:

\[ v_i = \frac{\sum_{k=1}^{n} x_k}{\text{card}(c_i)}; \quad 1 \leq i \leq c \]  \hfill (5)

where, \( \text{card}(c_i) \) denotes the cardinality of the class \( i \), i.e., the total number of objects belonging to this class. On the other hand, given a matrix of prototypes, \( V = \{v_1, v_2, ..., v_c\} \subset \mathbb{R}^d \) it is easy to calculate the membership degree of each object to each class according to the expression:

\[ u_k = \frac{d(x_k, v_i)^{-r}}{\sum_{j=1}^{c} (x_k, v_j)^{-r}}; \quad 1 \leq k \leq n; \quad 1 \leq i \leq c \]  \hfill (6)

where, \( d(x_k, v_i) \) denotes a distance measure between the \( k \)th object and the \( i \)th prototype and \( r > 0 \) is a weighting exponent and the sum in the denominator guaranties the constraints (2) and (3).

Based on these general ideas during the last decades, many algorithms have been proposed in the literature as heuristical approaches to the problem of determining the best prototypes of the \( c \) classes supposed present in a learning database \( X \). Some of these algorithms also known under the name of prototype generator classifiers (Karayiannis and Bezdek, 1997) require the number of clusters to be set as an input parameter (Rezaee et al., 1997) whilst others are more general and try to automatically determine also the best value of \( c \) and provide it as an output (Sun et al., 2004). In practice, however, sometimes, some practitioners chose to repeat the execution of a same algorithm that requires \( c \) as an input for different values of \( c \) and then use one or more validity criteria in order to assess and rank the quality of all the resulting solutions and finally select only the best solution according to the used validity criteria (Zolik and Zalik, 2011; Rezaee, 2010; Wang and Zhang, 2007; Kim et al., 2004; Bezdek, 1974).

All these heuristics are iterative procedures that try to exploit at best the information carried by each element of the learning database in order to converge to the best possible set of prototypes. They use for this a learning rule that fixes how this information should be used at each iteration. According to their learning rules, these heuristics can be divided into two main categories: hard and fuzzy.

In hard learning algorithms, at each iteration of the learning process, each object is assigned to a unique cluster causing the adaptation of only the prototype of this cluster (Ghosh et al., 2011; Qin and Suganthan, 2005). By contrast, in fuzzy learning algorithms, the information carried by each object of the learning dataset is used in order to update all the existing prototypes (Yang, 1993; Pedrycz, 1990).

In this study, researchers propose a new semi-fuzzy technique that tries to combine the advantages of these two categories avoiding in the same time, their drawbacks. This technique tries also to automatically determine the number of clusters. For this, at each iteration of the learning process the current object can be used in four different ways: to create a new cluster, to update the prototype of only one existing cluster, to update more than one existing prototype and to do nothing leaving the object for next re-examination. The choice among these four possibilities is based on a measure of inter-points similarities which can also be interpreted as a measure of membership degrees, plus a measure of ambiguity concerning the original class of an object when all we know about this origin is a set of membership degrees to different classes.

**DESCRIPTION OF THE PROPOSED METHOD**

The Clustering Method, researchers propose in this study is an explanatory data analysis tool that quickly explores the learning data by analyzing their similarities in order to detect and characterize the clusters they form. The inter-points similarity measure we use for this is given by:

\[ s(x_i, x_j) = 1 - \frac{d^2(x_i, x_j)}{p} \]  \hfill (7)

Where:

\[ d^2(x_i, x_j) = \frac{\sum_{l=1}^{p} (x_{ij} - x_{lj})^2}{r_i} \]  \hfill (8)

And:

\[ r_i = \max_{l \in S_i} \{x_{il}\} - \min_{l \in S_i} \{x_{il}\}; \quad 1 \leq j \leq p \]  \hfill (9)

The choice of this particular measure is justified by some important proprieties it verifies. The first propriety is that \( s(x_i, x_j) = 1 \) if and only if \( x_i = x_j \), meaning that the maximum of similarity (1 or 100%) is obtained for patterns
that are numerically described by the same object vector. The second property is described by \( s(x_i, x_o) = 0 \) if and only if \( |x_i - x_o| = r_i \forall 1 \leq j \leq p \) which means that the minimum of similarity (0%) is obtained for pairs of patterns whose vector objects present the maximum of difference for each of their \( p \) components. Another interesting property is that if one of the object vectors in \( s(x_i, x_o) \) is a cluster prototype which occurs for example when \( x_i = v_k \), this measure can also be viewed as a measure of the membership degree of the \( i \)th object to the \( k \)th cluster, i.e.,

\[ s(x_i, x_k) = u_k(i) = u_{ik}. \]

In practice however, users can replace this measure by any other one that would be more appropriate for each application. Given such a measure and an associated threshold \( \alpha \) that fixes the minimum of similarity that should exist on average between the elements of any class or cluster, the process of cluster detection operates as follows.

**The process of cluster detection:** The first cluster is created using the first element, \( x_i \), and the prototype of this cluster is initialized with this element putting \( c = 1 \) and \( v_i = x_i \). Then, using the chosen similarity measure, the \( n-1 \) remaining elements are sequentially and quickly analyzed in order to detect the clusters they are supposed to form. For each of these elements, say \( x_{i+n} \), researchers calculate the similarities to the prototypes of the \( c \) previously detected classes \((c \geq 1)\). The maximum of these similarities:

\[ \text{Max}_{1 \leq i \leq c} (s(x_i, v_i)) \]

is then compared to the value of the threshold \( \alpha \). If this maximum is \(< \alpha \), researchers consider that a new cluster is being detected and researchers initialize the prototype of this new cluster with \( x_i \) putting \( c = c + 1 \) and \( v_i = x_i \). The condition \( \text{Max}_{1 \leq i \leq c} (s(x_i, v_i)) \geq \alpha \) means in fact that the current element \( x_i \) does not possess the required minimum of similarity with the \( c \) already detected clusters in order to consider it as coming from one of these clusters.

On the other hand, when \( \text{Max}_{1 \leq i \leq c} (s(x_i, v_i)) \geq \alpha \) the current element cannot be viewed as coming from a not yet detected cluster but rather from an already detected one, probably the one whose prototype is the nearest to \( x_i \) or for which \( x_k \) presents the maximum degree of membership. However, in order to optimize the decision to make concerning the assignation of \( x_i \) and therefore, the way to exploit the information carried by \( x_i \) in order to enhance the learning process, a more rigorous method is utilized. This method is based on a measure of the ambiguity that arises when we try to assign \( x_i \) to one of the \( c \) clusters based only on its membership degrees to these clusters, i.e., \( u_{ik}, u_{ik} \rightarrow u_{ik} \). To quantify this ambiguity the following expression is used:

\[ A(k) = \frac{1}{\log(c)} \sum_{i=1}^{c} u_k \log(u_k) \quad \text{for} \quad 2 \leq k \leq n \quad (10) \]

According to this expression it is easy to see that the ambiguity is minimal and equals to 0 (or 0%) if there exists a class \( i \) such that \( u_{ik} = 1 \) and \( u_{ik} = 0 \) for all \( i \neq i \). On the other hand, the ambiguity is maximal and equals to 1 (or 100%) if \( u_{ik} = 1/c \) for each class \( i \). In the first case, it is clear that \( x_i \) can be assigned, confidently and without hesitation, to the class \( i \) for which \( u_{ik} = 1 \) and the prototype of this class adjusted in consequence so as to take into account this assignment in the next steps of the learning process. Let \( v_i(k-1) \) and \( v_i(k) \) denote, respectively the prototype of the \( i \)th class before and after assignment of, researchers have:

\[ v_i(k-1) = \frac{\sum_{i=1}^{k-1} u_{ik} \times x_i}{\sum_{i=1}^{k-1} u_{ik}} \]

\[ v_i(k) = \frac{\sum_{i=1}^{k-1} u_{ik} \times x_i + x_k}{\sum_{i=1}^{k-1} u_{ik} + 1} \quad (11) \]

with \( n_i(k-1) \) denoting the fuzzy cardinality of the class \( i \) after processing \((k-1)\) elements. Therefore, the learning rule which indicates how \( v_i(k) \) can be obtained from \( v_i(k-1) \) in this particular case can be derived as follows:

\[ v_i(k) = \frac{x_k + \sum_{i=1}^{k-1} u_{ik} \times x_i}{1 + n_i(k-1)} = \frac{x_k + n_i(k-1) \times v_i(k-1)}{1 + n_i(k-1)} \]

\[ = \frac{x_k - v_i(k-1)}{1 + n_i(k-1)} + v_i(k-1) \]

\[ = v_i(k-1) + \frac{1}{1 + n_i(k-1)} (x_k - v_i(k-1)) \quad (12) \]

By contrast, in the second case where the ambiguity is maximal, it is very hard to justify the assignation of \( x_i \) to any particular class. This is why we consider that the best decision to take in this particular case is to left \( x_i \) for later examination. This decision is in fact suitable not only for elements whose ambiguity is exactly equals to the maximal value \( A_{max} = 1 \) but also for any other element whose ambiguity can be considered as very high. However, the linguistic expression very high which can be verified at different degrees by different ambiguity values is fuzzy and needs to be clarified. For this, using the technique of alpha-cuts (Tajdjin et al., 2010) we consider that the condition very high is verified by all ambiguity values that are greater than the limit \( A_{max}/\sqrt{2} > 0.7 \). This limit is inspired from the calculation of the cut-off frequencies of low-pass and high-pass circuits in electronics (Horowitz and Hill, 1989) and its efficacy was empirically proven in the framework of this study.
By analogy, the hard decision of complete assignment to a unique class is not limited to elements with ambiguity of exactly 0 but rather extended to all elements with very low ambiguity where the expression very low means any value whose difference from 0 does not exceed $A_{low} = A_{max} / \sqrt{S}$ = 0.3.

Finally, for each element $x_k$ whose ambiguity $A(k)$ is in the interval (0.3–0.7) we consider that $x_k$ cannot be confidently assigned to a unique cluster because $A(k)$ is not very low but in the same time researchers cannot leave $x_k$ for later examination because its ambiguity is not very high. Consequently, the decision we make in this case is a fuzzy decision that consists in partially assigning $x_k$ to all the $c$ existing clusters. The result of this multiple assignment is an adjustment of all existing prototypes using the learning rule:

$$v_i(k) = v_i(k-1) + \frac{u_{ik}}{n_i(k)} [x_k - v_i(k-1)] \quad \text{for } 1 \leq i \leq c$$

(13)

which can be viewed as a fuzzy generalization of the hard rule expressed by Eq. 12. This rule means that the $c$ prototypes compete for the new object $x_k$ but instead of a unique winner that will takes all and be the only one which benefit from the operation of adjustment, researchers consider that all prototypes are winners but to different degrees that fix how each of them should be adjusted according to both its difference $x_k - v_i(k-1)$ with the current object and its similarity $u_{ik}$ to this object.

In summary, the learning rule that fixes both the number of prototypes to be adjusted after processing each element $x_k$ and the way these adjustments should be performed is based as shown in Fig. 1, on a classification of the ambiguity of $x_k$ into three fuzzy classes: Very Low (VL), Very High (VH) and NOT(VL or VH). For the first class, only one prototype is adjusted using the hard rule given by Eq. 12; for the second class, no prototype is adjusted and $x_k$ is left for later examination and for the last class, all existing prototypes are adjusted using the fuzzy rule given by Eq. 13. Hence the semi-fuzzy algorithm name.

Characterization of the detected clusters: Once the last element $x_n$ is analyzed, the process of cluster detection provides as results, the number of clusters $c$ which form the learning base $X$ plus a set of $c$ prototypes $v_1, v_2, \ldots, v_c$ that can be used in order to characterize the detected clusters. However, these prototypes are not necessarily the best because they do not optimize any objective criterion. This why we submit them to an optimization procedure which consists in a simplified version of the fuzzy $c$ means algorithm (Bezdek, 1981) whose role is to enhance these prototypes so that they minimize the criterion:

$$J(U, V, X) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^2 d^2(x_k, v_i)$$

(14)

where, $U$ and $V$ denote respectively the matrix of membership degrees and the matrix of prototypes. For this, starting from the produced prototypes as initial ones that should be enhanced, researchers calculate the elements of $U$ using the relation:

$$u_{ik} = \frac{d(x_k, v_i)^2}{\sum_{j=1}^{c} d(x_k, v_j)^2}; \quad 1 \leq k \leq n; \quad 1 \leq i \leq c$$

(15)

Then, using the calculated membership degrees researchers recalculate the prototypes according to the relation:

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^2 x_k}{\sum_{k=1}^{n} (u_{ik})^2}; \quad 1 \leq i \leq c$$

(16)

This process of recalculating $U$ and $V$ is then repeated until convergence, i.e., until the difference between the prototype matrices at two successive iterations $t$ and $t-1$ becomes insignificant or in other words, until:

$$\left| V_t - V_{t-1} \right| = \max_{1 \leq i \leq c} \left( \max_{1 \leq k \leq n} \left| v_{ik} - v_{ik-1} \right| \right) < \varepsilon$$

(17)

where, $\varepsilon$ is a small threshold that can be fixed by the user.
The problem of cluster validity: One of the most important characteristics of the proposed algorithm is that it does not require the number of clusters as an input. Therefore, it would be very useful in situations where the number of clusters is not a priori known. However, the number of clusters the algorithm detects within each learning data set may depend on the parameter \( \alpha \) whose value should be fixed by the user.

In the absence of any prior knowledge that can be used in order to fix the best possible value for \( \alpha \), the algorithm can be repeated for different values of this parameter. These values can be chosen within the interval \([\alpha_{\text{min}} - \alpha_{\text{max}}]\) where:

\[
\alpha_{\text{min}} = \min_{i \in k} (s(x_i, x_k))
\]

And:

\[
\alpha_{\text{max}} = \max_{i \in k} (s(x_i, x_k))
\]

The limit \( \alpha_{\text{max}} \) will result in because in this case the condition of detecting a new cluster will never be satisfied. On the other hand, for \( \alpha = \alpha_{\text{min}} \) the number of detected clusters will be equal to \( n \) because this condition will be always verified. In practice if researchers vary \( \alpha \) between these two limits with a reasonable step \( \Delta \alpha \), different classification results may be obtained and these results may differ in terms of the number of detected clusters and therefore also in terms of the quality of final classification.

Each of these results can be viewed as a candidate solution and the problem of cluster validity consists in finding a way to select the best one among all candidates. To solve this problem we propose a heuristic which consists in studying the variations of the number of detected clusters with the \( \alpha \) parameter. This heuristic is based on the intuition, verified by several experiments that if a good structure of \( c \) clusters is really present in the learning data base \( X \), this structure will be detected for several different values of \( \alpha \) and these values will form an interval whose length is greater than the one associated with any other value of \( c \).

Hence, by quickly exploring the learning data for different \( \alpha \) values that can be separated by a relatively large step \( \Delta \alpha \), researchers can easily determine first the optimal number of clusters and then apply the optimization procedure for only one partition. In contrast with other methods based on validity criteria (Xie and Beni, 1991) this method does not require the search of multiple sub-optimal solutions and then assess and rank these solutions in order to keep only one of them and reject all the others.

### EXPERIMENTAL RESULTS

To show the effectiveness of the proposed algorithm, researchers present in this section typical results of its application to four examples of test data sets publicly available at the UCI learning machine repository (Frank and Asuncion, 2010). The first example is the Breast Cancer Wisconsin data (BCW) which contains two classes of respectively 458 and 241 object vectors of 9 parameters.

The second example is the Pima Indians Diabetes data set (PID) which is provided by the National Institute of Diabetes and Digestive and Kidney Diseases. It is constituted of 768 twenty-two-dimensional object vectors, divided into two different classes of respectively 500 and 268 object vectors.

The third example is the well-known Iris data (Iris) set which contains 150 four-dimensional object vectors representing three different classes of flowers (Sesoto, Versicolour and Virginia) with 50 elements per class. A characteristic of this example is that the first class is well separated from the other two which are overlapping and difficult to separate. The last example, researchers call Iris23 is a subset of iris which contains only the two overlapping classes.

Table 1 shows the minimum, the maximum and the mean values of inter-point similarities within the four data sets.

For each of these examples, researchers first run the cluster detection procedure for the values of \( \alpha \) comprised between \( \alpha_{\text{min}} \) and \( \alpha_{\text{max}} \) and separated by a step \( \Delta \alpha = (\alpha_{\text{max}} - \alpha_{\text{min}})/10 \). The role of this first and quick exploration is to show how the number of detected clusters varies with the threshold.

Analysis of Table 2 shows that some values of \( c \) are skipped over when we vary \( \alpha \) with a relatively large step. This is the case, for example, for \( c \) values of 3-5 which are all skipped over when passes from 43.04-52.53% as well as for the values of 7-9 when passes from 52.53-62.02%.

**Table 1: Inter-point similarities for the four examples of test data**

<table>
<thead>
<tr>
<th>Limits</th>
<th>BCW</th>
<th>PID</th>
<th>Iris</th>
<th>Iris23</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_{\text{min}} ) (%)</td>
<td>5.06</td>
<td>39.71</td>
<td>17.44</td>
<td>17.69</td>
</tr>
<tr>
<td>( \alpha_{\text{max}} ) (%)</td>
<td>100.00</td>
<td>98.13</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>( \alpha_{\text{mean}} ) (%)</td>
<td>63.69</td>
<td>78.61</td>
<td>67.76</td>
<td>75.46</td>
</tr>
</tbody>
</table>

**Table 2: Variation of the number of detected clusters for the BCW data with\( \Delta \alpha = (\alpha_{\text{max}} - \alpha_{\text{min}})/10 \)**

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.44-50.64</td>
<td>1</td>
</tr>
<tr>
<td>24.05-43.04</td>
<td>2</td>
</tr>
<tr>
<td>52.53</td>
<td>6</td>
</tr>
<tr>
<td>62.02</td>
<td>10</td>
</tr>
</tbody>
</table>
To obtain these values researchers explored the interval (43-62) of $\alpha$ values with a small step $\Delta \alpha = 1$. Results of this exploration are reported on Table 3. Note that for all the experiments a maximum number of clusters was fixed and its value for this example is 10.

For the determination of the optimal number of clusters, researchers used the heuristic explained in the study according to which this number corresponds to the largest interval of $\alpha$ for which the number of detected clusters remains constant. Figure 2 shows that for the BCW data this number is equal to 2.

The same method was applied to the other examples and results of their analysis are reported of Fig. 3-5. As researchers can see on these figures, for each data set the largest interval of the threshold of similarity indicates the right value of the number of clusters (in the case of Iris data, both $c = 2$ or $c = 3$ is acceptable (Bezdek et al., 1999).

For each of these examples, once the optimal number of clusters is determined, the process of characterization of the detected clusters is launched. This process consists in optimizing the prototypes or representatives of these clusters using the method described in the previous section. These prototypes are then compared to the actual centers of the classes that form each data example. As researchers can see in Table 4, these comparisons show that the learned prototypes are very close to the actual centers which shows the ability of the proposed algorithm to both detect the clusters present in a totally unlabelled data set and characterize the detected clusters with good prototypes.

To show the practical usefulness of the learned prototypes, researchers finally used them as a basis for

---

Table 3: Variation of the number of detected clusters for the BCW data with $\Delta \alpha = 1$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[43-44]</td>
<td>2</td>
</tr>
<tr>
<td>[45-47]</td>
<td>3</td>
</tr>
<tr>
<td>[48-50]</td>
<td>4</td>
</tr>
<tr>
<td>[51-52]</td>
<td>5</td>
</tr>
<tr>
<td>53</td>
<td>6</td>
</tr>
<tr>
<td>[54-56]</td>
<td>7</td>
</tr>
<tr>
<td>57</td>
<td>8</td>
</tr>
<tr>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>59</td>
<td>10</td>
</tr>
</tbody>
</table>
applying the nearest prototype rule (Kuncheva and Bezdek, 1998) in order to reconstruct the original clusters. Results of these experiments are shown in Table 5 in terms of misclassification errors. These results confirm that the provided prototypes can be considered as good representatives of the corresponding clusters.

**CONCLUSION**

In this research, researchers proposed a new semi-fuzzy learning algorithm for detecting and characterizing clusters or homogenous groups within sets of totally unlabelled data. The algorithm combines two categories of learning rules: hard and fuzzy. Based on an inter-point similarity and a measure of ambiguity, the algorithm rapidly explores the data to analyze in order to detect the number of clusters they form. Then, an optimized prototype is calculated as a representative for each detected cluster.

The effectiveness of this algorithm was illustrated using simulation results obtained by its application to different examples of real test data publicly available. These results are very encouraging and motivate future investigations.

**REFERENCES**


