Unsupervised Minor Prototype Detection using an Adaptive Population Partitioning Algorithm

Jung-Hsien Chiang* and Zong-Xian Yin
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Abstract

This paper presents a new partitioning algorithm, adaptive c-population clustering (ACP) algorithm, which can be used to identify natural subgroups, as well as influential minor prototypes, in the unlabeled data set. The traditional fuzzy $c$-mean clustering algorithm is primarily a prototype-based clustering algorithm, whereas the ACP algorithm is able to adaptively separate isolated minor clusters from major clusters. Three specific objectives underlie the presentation of the adaptive c-population clustering algorithm in this paper. The first is to describe mathematical model of this approach, and the second is to show that proposed algorithm converges to a stable solution. The third and more complex goal is to demonstrate that the proposed algorithm is able to perform clustering adaptively. We illustrate this approach with two numerical examples in order to verify the clustering effectiveness.

Keywords: Minor Prototype, Cluster Analysis, Fuzzy Clustering, Outlier

* Author to whom correspondence should be sent
1. Introduction

Clustering methods have been used extensively in computer vision and pattern recognition [Dunn, 1973; Bezdek, 1981; Ismail, 1986; Kandel, 1986; Krishnapuram, 1993; Rose, 1993; Kamel, 1994; Man, 1994; Pedrycz, 1996]. Of these methods, fuzzy clustering has been shown to be effective in solving the problem that each data point has a membership grade indicating its belongingness degree to each cluster, rather than assigning it to only one of the clusters as in the crisp clustering case [Huntsberger, 1990; Pal, 1995; Pedrycz, 1996; Karayiannis, 1997; Imai, 1998]. The fuzzy c-means (FCM) algorithm has been one of the best known fuzzy clustering approaches, and its use for various applications is well described and analyzed in [Bezdek, 1992]. The research described here presents a new fuzzy clustering method from the point of view of variform cluster analysis. Specifically, our goal is to develop an adaptive c-population clustering algorithm that can separate isolated small clusters from multiple clusters. This problem is related to the minor prototypes and the adaptability of clustering, as discussed below.

Problem Description

In the cluster analysis process, clustering algorithms generate solutions that identify natural subgroups in the unlabeled data set. When the clustering algorithm is used to produce meaningful interpretation of data set, it must identify “good and useful” subgroups. In some engineering applications it is advantageous to perform clustering adaptively, rather than by using the classical prototype-based clustering techniques. This is especially true in real applications when the characteristics of the prototype are unknown. In this approach, we consider those data points that have a small and dense region but may have relatively large distances to all cluster centers as an individual prototype, and this is referred to as a minor prototype, referred to as an influential prototype as well in this paper. Fig 1 depicts a classic problem, in which all of the three handwritten images are valid representations of the digit “7”. Traditional classification schemes may consider all of the patterns which belong to the same class as one category. However, this may not be a suitable approach for handwritten character patterns since many character classes consist
of sub-classes due to different styles of writing. Those specific prototypes with minor populations of patterns are referred to as *allographs* [Parizeau, 1995; Chiang, 1997], and we consider them as minor prototypes. As shown on the left of Fig 1, in a minority of this kind of pattern, called *Type I* or strikeout pattern, there is an oblique line crossing the vertical stroke. Having this oblique line is unique for the digit “7”, making *Type I* as a peculiar pattern. Another type, the middle one of Fig 1, is called *Type II* or printed style. This exemplar has a neat and tidy shape. The last one is shown in right side of Fig 1, is called *Type III* pattern. The differences between *Type II* and *Type III* patterns are that *Type III* has a square body, and *Type II* has a clear stroke on the upper-left. *Type III* comprises the largest share of the dataset, with about 50%; *Type II* has about 45 % of the 600 samples of the handwritten digit “7” dataset; and *Type I* is a very minor group, very different from the others, and comprising of about 5 %. Although *Type I* is a very small group in the dataset, it is a significant allograph in handwritten digits, so clearly, we do not treat it as “outlier”. In robustness aspect, outliers represent contaminated patterns (noise) that should be ignored while clustering [Krishnapuram, 1993; Dave, 1997; Keller, 2000]. It is worth noting here that “outliers” are not necessarily noise patterns. They are patterns that are significantly different from the majority. One may not want to discard them if they are legitimate patterns that represent a minority class. For example, minority class detection and analysis is a significant data mining task that can be used to detect unusual usage of credit cards or identify unusual responses to various medical treatments.

![Fig 1. Example of typical prototypes and minor prototype for the handwritten digit “7”.](image)

In the FCM approach, a group of data points is divided into a given number of subgroups having almost even numbers of points. It was unable to obtain such results because it is not possible to separate the minor prototype from dataset in the application. An effective clustering procedure should consider the minor prototypes as separated cluster. Therefore, there is a need for an adaptive clustering procedure that effectively considers minor prototypes as separated classes,
and represents them as individual clusters.

The main contribution of this work is the development and demonstration of a novel clustering method in pattern analysis application. The most important advantage of the ACP algorithm over traditional clustering algorithm is that it is capable of explicitly separating minor prototypes from mixed data pools for further cluster memberships analysis or classification purpose, whereas classic clustering is not able to do so. Specifically, the goal of this work is to seek a proper representation for minority classes. This is especially important for the application examples described above.

The organization of this paper is as follows. We first give an outline of the fuzzy clustering family in Section 2, and then describe the adaptive c-population clustering algorithm. In Section 3, we give the mathematical justification of the adaptive c-population clustering algorithm, which is based on the convergence theorem of stochastic approximation. We also provide a transform analysis for the adaptive clustering algorithm. Experiments are then discussed.
2. Adaptive C-Population Clustering Algorithms

Partitional fuzzy clustering algorithms

Let $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^r$ be a finite set of data points and let $c$ be an integer, $1 < c < n$. A real $c \times n$ matrix $U$ can be used to represent the result of a cluster analysis of $X$ by interpreting $u_{ij}$ as the degree to which $x_j$ belongs to cluster $i$. We use $\beta = \{\beta_1, \beta_2, \ldots, \beta_c\}$ to denote the $c$-tuple of cluster prototype.

The fuzzy $c$-means clustering (FCM) algorithm attempts to find good cluster structure descriptors $(u, v)$ as minimal solutions of a particular objective function [Bezdek, 1981; Hathaway, 1995],

$$ F(U; V; X) = \sum_{y=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^2 $$

where $u_{ij} \in U$, and

$$ U = \{u_{ij} \in [0,1] \quad \forall i, j, \quad \sum_{j=1}^{n} u_{ij} = 1 \quad \forall j, \quad \sum_{i=1}^{n} u_{ij} > 0 \quad \forall i \} $$

$V = \{v_1, v_2, \ldots, v_c\}$, where $v_i$ specifies the $i$th cluster prototype;

$m > 1$ is the degree of fuzzification;

$c \geq 2$ is the number of clusters;

$d_{ij}$ is the distance between pattern $X_j$ and $i$-th cluster prototype.

The FCM scheme is primarily an equal population partitioning algorithm. This type of algorithms simply divides the sample points into $c$ clusters equally according to summations of the inverted within-cluster distances. The generic clusters whose populations are sufficiently large to be detected by the algorithm are referred to as “typical prototypes”. However, a good cluster should include the group of points that are highly concentrated, and should not be equally partitioned. The most significant drawback of FCM is that it does not capture the nature groups’ centers when minor prototypes exist. A minor prototype is defined as an isolated and small group
of a set of compact sample points. Considering the Type I allograph in Fig. 1, this group represents a special and important prototype that has significantly different characteristics from the others. Since there are only a few samples in this group, it is not readily distinguish from the typical prototypes, and is often overlooked by the FCM algorithm.

*Adaptive C-Population Clustering algorithm*

The adaptive c-population clustering (ACP) algorithm introduced in this study generalizes the constrained membership values as:

\[ \sum_{i=1}^{c} u_{ij} = w_j \]

(2)

where \( w_j \) are remedial factors for prototype deviation. This remedial factor is an indicator of the power of the sample point \( x_j \) to form the legitimate clusters. The estimation of \( w_j \) will be discussed in detail later. The ACP approach is introduced primarily to improve a weakness of the FCM-type clustering, whose memberships summation of a sample point is equal to 1 which implying that each sample point is assigned an equal importance when forming clusters. It is clear from (2) that the Lagrangian objective function may be formulated as

\[ F_j = \sum_{i=1}^{c} u_{ij}^m d_{ij}^2 - \lambda \left( \sum_{i=1}^{c} u_{ij} - w_j \right) \quad \text{for all } j \]

(3)

where \( \lambda \) is the Lagrangian multiplier.

The first-order necessary conditions for optimality are found by setting the gradient of \( F_j \) with respect to \( u_{ij} \) equal to 0. Here:

\[ \frac{\partial F_j}{\partial u_{ij}} = m(u_{ij})^{m-1} d_{ij}^2 - \lambda \]

Setting \( \partial F_j / \partial u_{ij} = 0 \) yields
This is an ACP membership function computation formula. The estimation of the memberships of a given point considers not only the c-partition in the global pattern distribution (the denominator part) but also the individual importance in the local region (the numerator portion). The remedial factor is utilized to enhance the data points in the concentration region with greater membership values, and eventually causes the deviations of the cluster centroids towards those concentration regions. Hence, the major advantage of the proposed ACP approach is that it can identify minor clusters, and is not simply restricted to normal and typical clusters. As mentioned above, the adaptive c-population clustering algorithm achieves its adaptability by adjusting the remedial factor $w_j$ appropriately when performing membership computations. The estimation of $w_j$ is related to the scale of the within-cluster density. A quantitative measure of the density ratio is given by

$$w_j = \frac{N_{r_j}}{N_{w}}$$

(5)

where $N_{r_j}$ is the number of data points which are in the neighborhood within a certain reference distance of $x_j$, and $N_{w}$ is the minimum number of neighbors of a point that can be considered as a member of a legitimate cluster. The reference distance is usually chosen as approximately 1.8–3.4 times the minimum inter-distance in the data set. The remedial factor $w_j$ determines the “degree of importance” of the data points in clustering, and the ACP algorithm considers point $x_j$ as a member of the prototype if $w_j$ is sufficiently large. During the clustering operation, high value of $w_j$ indicates that the point has a high possibility of contribution to any one of the given clusters, which ensuring that the ACP can detect the minor prototypes if they existed. The value of $w_j$ is fixed for all iterations, and as a special case, the ACP algorithm is degenerated as the FCM.
algorithm when $N_r$ and $N_m$ are identical.

Table 1 The clustering procedures of the ACP algorithm

```
BEGIN
    Given a set of $n$ patterns $X = \{x_1, x_2, \ldots, x_n\}$.
    Fix the numbers of clusters $c$
    Initialize $k = 0$
    Initialize cluster center $v_i$, for $i = 1, 2, \ldots, c$
    Compute remedial factors $w_j$, for $j = 1, 2, \ldots, n$
    REPEAT
        Update membership of the prototypes $u_j$ using (4)
        Update all cluster centers $v_{i,k}$
        $v_{i,k} = \frac{\sum_{j=1}^{n} u_j^n x_j}{\sum_{j=1}^{n} u_j^n}$
        Increment $k$
    UNTIL $\|v_{i,k} - v_i\| < \varepsilon$
END
```

We may therefore state that the interpretation of the $u_j$ is the typicality of data point $x_i$ belonging to cluster, $v_i$. It may be noted that the $w_j$ in Eq.(4) of the ACP algorithm, the $\eta$ in membership computation of the possibilistic clustering [Krishnapuram,1993] and the $f_j$ in the conditional FCM [Pedrycz, 1996] are quite different. The interpretations of these parameters are quite different. $\eta$ in the possibilistic clustering determines the distance at which the membership value of a sample point in a cluster becomes 0.5. It was introduced primarily to make the clustering algorithm less sensitive to noise and outliers, whereas the $w_j$ is used to adaptively identify isolated minor clusters from several major clusters. The conditional variance $f_j$ in the conditional FCM is a prior specified indicator describes a level of involvement of $x_j$ in the constructed clusters, whereas the $w_j$ is a factor proportional to the within-cluster density. Furthermore, Keller [2000] considered all outliers had a relatively large distance to all other groups and were equally shared among the groups, hence, he also assigned the outliers with a large weight to reduce their influence. On the contrary, $w_j$ in our ACP is utilized to detect the minor prototypes instead of getting rid of
the noise prototype.

The ACP algorithm for finding stationary points of $F$ are summarized in Table 1. A detailed discussion of parameter $w_j$ will be given later in this section. A convergence analysis of this algorithm is presented in next section.

We now use a simple example to illustrate the idea of deviation clustering. This example involves an artificial dataset with two well-separated clusters, each of which consists of 9 points. The feature vectors are numbered in the order in which they would be encountered from the top down, and from left to right in a scan of the plot shown in Fig. 2. Assuming that the numbers of clusters is 2, then the data set should be classified into two clusters: $C_1: \{x_1, x_2, \ldots, x_9\}$, and $C_2: \{x_{10}, x_{11}, \ldots, x_{18}\}$. Using the syntactic data set, both the FCM and ACP algorithms obtain reasonable partitions, as shown in Fig 2.

Now we evaluate the clustering process with an additional minor prototype, containing two samples $x_{19} = (7.5, 2.1)$ and $x_{20} = (7.5, 1.8)$. However, we would like to caution that the isolated cluster $\{x_{19}, x_{20}\}$ represents a minor prototype rather than noise. Here we set the number of clusters as 3. Figs. 3 (a) and (b) show the final crisp partitions obtained from FCM and ACP algorithms with identical initializations, respectively. The final partitions are much different between FCM and ACP. It can be seen that the FCM algorithm actually finds a c-partition of a given data set,
regardless of how many “clusters” are present in the data set. Obviously, this would be unable to separate an isolated minor cluster from two major clusters, which is unacceptable. Contrary to FCM, the ACP approach sets both cluster centers at the major clusters, and identifies the two close patterns that have large deviations from the major clusters as another cluster. It can be seen that ACP has the ability to estimate proper cluster centers when minor prototypes exist.

Table 2 Membership values and cluster centers obtained from the FCM and the ACP algorithms.

<table>
<thead>
<tr>
<th></th>
<th>FCM</th>
<th></th>
<th>ACP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cluster 1</td>
<td>Cluster 2</td>
<td>Cluster 3</td>
<td>Cluster 1</td>
</tr>
<tr>
<td>1</td>
<td>0.8957</td>
<td>0.0710</td>
<td>0.0334</td>
<td>0.4494</td>
</tr>
<tr>
<td>2</td>
<td>0.8957</td>
<td>0.0711</td>
<td>0.0332</td>
<td>0.4494</td>
</tr>
<tr>
<td>3</td>
<td>0.9779</td>
<td>0.0156</td>
<td>0.0666</td>
<td>0.977</td>
</tr>
<tr>
<td>4</td>
<td>0.9779</td>
<td>0.0156</td>
<td>0.0666</td>
<td>0.977</td>
</tr>
<tr>
<td>5</td>
<td>0.9998</td>
<td>0.0002</td>
<td>0.0001</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>0.9597</td>
<td>0.0296</td>
<td>0.0107</td>
<td>0.9661</td>
</tr>
<tr>
<td>7</td>
<td>0.9597</td>
<td>0.0296</td>
<td>0.0107</td>
<td>0.9661</td>
</tr>
<tr>
<td>8</td>
<td>0.7600</td>
<td>0.1805</td>
<td>0.0595</td>
<td>0.3919</td>
</tr>
<tr>
<td>9</td>
<td>0.7600</td>
<td>0.1809</td>
<td>0.0591</td>
<td>0.3920</td>
</tr>
<tr>
<td>10</td>
<td>0.1215</td>
<td>0.7587</td>
<td>0.1198</td>
<td>0.0705</td>
</tr>
<tr>
<td>11</td>
<td>0.1205</td>
<td>0.7628</td>
<td>0.1166</td>
<td>0.0705</td>
</tr>
<tr>
<td>12</td>
<td>0.0144</td>
<td>0.9547</td>
<td>0.0308</td>
<td>0.0203</td>
</tr>
<tr>
<td>13</td>
<td>0.0139</td>
<td>0.9570</td>
<td>0.0292</td>
<td>0.0202</td>
</tr>
<tr>
<td>14</td>
<td>0.0025</td>
<td>0.9883</td>
<td>0.0092</td>
<td>0.0000</td>
</tr>
<tr>
<td>15</td>
<td>0.0219</td>
<td>0.8311</td>
<td>0.1470</td>
<td>0.0167</td>
</tr>
<tr>
<td>16</td>
<td>0.0217</td>
<td>0.8365</td>
<td>0.1418</td>
<td>0.0167</td>
</tr>
<tr>
<td>17</td>
<td>0.0409</td>
<td>0.4411</td>
<td>0.5180</td>
<td>0.0223</td>
</tr>
<tr>
<td>18</td>
<td>0.0426</td>
<td>0.4637</td>
<td>0.4937</td>
<td>0.0228</td>
</tr>
<tr>
<td>19</td>
<td>0.0043</td>
<td>0.0254</td>
<td>0.9703</td>
<td>0.0007</td>
</tr>
<tr>
<td>20</td>
<td>0.0043</td>
<td>0.0254</td>
<td>0.9703</td>
<td>0.0007</td>
</tr>
</tbody>
</table>

Initial centers | (0.00, 0.00) | (1.00, 0.00) | (2.00, 0.00) | (0.00, 0.00) | (1.00, 0.00) | (2.00, 0.00) |
Final centers  | (0.98, 2.00) | (4.79, 2.00) | (7.08, 1.95) | (0.98, 2.00) | (4.98, 2.00) | (7.42, 2.00) |

Furthermore, the membership values and the cluster centers obtained are considerably different, as can be seen in Table 2. Points \( \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \} \) in the table are far away from the dense regions or cluster centers, and the FCM algorithm gives fairly high memberships in
cluster 1 for those scattered points. A similar result is also obtained for cluster 2. This significantly affects the estimation of the cluster centers. In contrast to this, the ACP algorithm gives low memberships for those far points in each cluster. On the other hand, the ACP algorithm provides more graded membership values, and the farther point has a lower membership than the close one, as desired.

Here we further discuss the relationship between cluster centers and remedial factor $w_j$ in the ACP algorithm. In Eqs. (4) and (5), we formulated $w_j$ as a ratio form of $N_{r_j}$ and $N_u$. In the left charts of Fig 4, the horizontal axis $N_{r_j}$ is the number of data points which are in the neighborhood within a certain reference distance of $x_j$, which can be represented as a fuzzy set. The $N_u$ is the minimum numbers of data points within the same reference distance, and it defines for any point the minimum surrounding numbers of data points for which the point can be considered as a member of a legitimate cluster. The trapezoid membership function is the upper bound of $u_j$ for different $N_{r_j}$, representing the optimistic degrees for a point to be a member of a cluster. The right side of the chart illustrates the cluster centers after partition for corresponding $N_u$ values. It can be seen that the points with $N_{r_j}$ values smaller than $N_u$ are considered to have a low degree of importance to a cluster. In other words, their memberships are within \[
\frac{N_{r_j}}{N_u} < 1
\] limit. The clustering results are shown in Figs. 4 (a), (b), and (c) for when the parameter $N_u$ is set as 1,3, and 5, respectively. For example, when $N_u = 1$, the ACP cannot find appropriate cluster centers, which is similar to the unacceptable result of FCM. When we set $N_u = 3$, cluster centers will deviate from this manner and move toward reasonable positions. The clustering results of Figs. 4 (b) and (c) are similar, both unaffected by the exceptional two points. Clearly, the ACP algorithm identifies minor prototypes by examining the concentration characteristics of objects in a group.
Transform Analysis of the ACP algorithm

The updating rule (6) of the ACP scheme can be explicitly represented as a recursive and time-varying difference equation. Using discrete-time formalism, the updating formula is cast in a form whereby, given the \( i \text{th} \) cluster center \( v_{i,k} \) at current discrete time \( k \), we may compute the updated value \( v_{i,k+1} \) at time \( k+1 \) as follows

\[
 v_{i,k+1} = v_{i,k} + \frac{\sum_{j=1}^{n} u_{j,k+1}^{-1} (x_j - v_{i,k})}{\sum_{j=1}^{n} u_{j,k+1}^{-1}}
\]

(7)

Let \( x \in X \subset \mathbb{R}^r \). The salient features of the ACP model are contained in Fig. 5, and the updating rule can be propagated towards using the general discrete-time formalism [Haykin, 1999]:

![Upper bound of \( U_{ij} \) for different \( N_m \) values](image)

(a) \( N_m = 1 \)

(b) \( N_m = 3 \)

(c) \( N_m = 5 \)

Fig. 4 Relations between the cluster centers and corresponding \( N_m \) values for different \( N_m \) values. Final cluster centers are shown as “\( \Diamond \)".
where $\alpha_{y,i}$ is the corresponding time-varying learning parameter which is a non-linear function of $u_y$, varying dynamically during learning for stable results.

We now explore the ACP algorithm based on the Z-transform analysis method [Oppenheim, 1989]. According to Eq (8), we can readily see that

$$v_{i,k+1} = v_{i,k} + \alpha_{y,i} \times (x - v_{i,k})$$

$$= (1 - \alpha_{y,i}) v_{i,k} + \alpha_{y,i} x$$

Applying the discrete-time Z-transform on Eq. (9), we obtain the polynomials in the $z$ domain as

$$zV(z) = (1 - \varphi)V(z) + \varphi X(z)$$

where $\varphi$ is the z-transform of $\alpha_{y,i}$, and $z$ is the complex frequency variable in the Z-plane, i.e., on the unit circle $z = e^{iw}$, and $w$ is the discrete frequency. We assume zero-initial states in this case.

The relationship between the input pattern, $X$, and the estimated cluster centers, $V$, is given by the transfer function

$$H(z) = \frac{V(z)}{X(z)} = \frac{\varphi}{z - (1 - \varphi)}$$

Fig. 5 AFC clustering networks
and the magnitude-squared function is

\[
|H(e^{j\omega})|^2 = H(e^{jw})H(e^{jw})
\]

\[
= \left(\frac{\varphi}{\cos w + j\sin w - (1 - \varphi)}\right)\left(\frac{\varphi}{\cos w - j\sin w - (1 - \varphi)}\right)
\]

\[
= \frac{\varphi^2}{1 + (1 - \varphi)^2 - 2(1 - \varphi)\cos w}
\]

(12)

where \( w \in [-\pi, \pi] \). The pole-zero plot and the region of convergence (ROC) for the example are shown in Fig. 6(a). In this case the ROC is \(|z| > |\varphi|\), and the condition for stability is \(|1 - \varphi| < 1\), which is consistent with the convergence properties. For \(|1 - \varphi| > 1\), the ROC does not include the unit circle and, thus, the Z-transform would not exist. The schematic frequency-response function is shown in Fig. 6(b). It can be seen that the frequency-response functions of Eq.(12), i.e. the relationship between the input pattern and cluster center for the ACP clustering algorithm, are similar to the properties of the \textit{discrete-time filter}. Specifically, the transfer function behaves as a “low-pass filter” for \( \varphi < 1 \) whereas the transfer function behaves as an “all-pass filter” for \( \varphi = 1 \).

It may be worth noting that the ACP algorithm behaves as a “low-pass filter” and a “high-pass filter” when \( \varphi < 1 \) and \( \varphi > 1 \), respectively. Since the learning parameter \( \varphi \) is a suitable value, i.e., it is a bounded sequence and always smaller than 1, it follows that an ACP algorithm may be
viewed as a *low-pass filter*. Therefore, the clustering procedure can be considered as an iterative procedure to minimize the error (i.e. distance) between the “input signal” and its “estimated fuzzy centroids” defined over the partitioning objective function. Furthermore, it can be seen that the learning parameter $\phi$ controls the degree to modify the value between them, which can be interpreted as allowing the cluster centers to adaptively drift among patterns through the clustering procedure.
3. Experimental Results

In this section, we give two numerical examples to illustrate the characteristics of the proposed ACP algorithm. First we present an unequal cluster population data to illustrate our approach. We also demonstrate that the ACP can be used to separate the minor prototype pattern in this example. The second example describes the allograph exemplars clustering based on the ACP algorithm for the dataset of handwritten digits. In both examples we utilize the Xie-Beni ($XB$) validity index [Xie, 1991] and the normalized Xie-Beni ($nXB$) index to validate the clustering results. The $XB$ index denotes as

$$XB = \frac{\sigma_z/N}{d_{\min}}$$

(13)

where $\sigma_z = \sum_{i=1}^{c} \sum_{j=1}^{n} u^2_{ij} \|v_j - v_i\|^2$ and $d_{\min} = \min_{i,j=1,...,c} \|v_j - c_j\|^2$. The Normalized $XB$ index is defined as

$$nXB = \frac{\sigma'_z/N}{d_{\min}}$$

(14)

where $\sigma'_z = \sum_{j=1}^{c} \sum_{j=1}^{n} u^2_{ij} \|v_j - v_i\|^2 \sum_{k=1}^{c} u^2_{kj}$. We utilize the $XB$ measure to validate the effectiveness of clustering results with different numbers of clusters and the $nXB$ measure to compare the relative clustering effectiveness between the FCM and the ACP algorithms. Lower $XB$ and $nXB$ values indicate that the obtained clustering results are compact and well-separated.

Example 1

In order to compare the adaptive properties of the FCM and ACP, we use an unequal population data as an experimental data set. This data set has been used to illustrate the partially supervised clustering algorithm in [Bensaid, 1996]. We note that those 3 recognizable patterns far away from the large cluster represent minor prototype. Our experiments are based on clustering of both FCM and ACP with all of the data points (43 patterns), and Fig. 7(a) depicts FCM’s partitioning of the data set using $c = 2$ and $m = 2$. It can be seen that the FCM arbitrarily divides the
large cluster into two components (although this may depend on initialization), which is not a meaningful partition from a natural grouping point of view. This result does not mean that the FCM is not useful, only that FCM clustering considers all of the patterns equally important. In the ACP approach, with $N_m = 3$, it could identify the minor prototype patterns and produce two reasonable clusters. The clustering results obtained from the ACP are shown in Fig 7(b).

![Fig 7](image)

Fig 7 Plots of results on a data set of unequal cluster populations: With one minor prototype, (a) the crisp partition resulting from the FCM algorithm; (b) the crisp partition from the ACP algorithm. With two minor prototypes, (c) the results from the FCM; (d) the results from the ACP. Points belonging to different clusters are shown using different symbols, and cluster centers are shown as “◊”.

We then extend the original data set as an example of one major cluster and two minor clusters of three points each. It may be noted that the pattern distributions in both cases are similar, but in this example we examine if the clustering algorithm is able to detect both minor clusters. By
using the FCM, since it favors major clusters of large size, the clustering results do not capture the notion of “major prototype with concentration region”. Therefore, the resulting FCM partitions, as shown in Fig. 7(c), split the “major” group into several clusters, which is not desirable. It can be seen in Fig 7(d) that the ACP approach with identical initializations is quite successful in improving the robustness of FCM clustering. Using the ACP algorithm, the three prototypes (one major prototype and two minor prototypes) are automatically attracted to respective concentration regions in pattern space as the iterations proceed.

Table 3 shows the $XB$ and $nXB$ validity indices of above two examples. The lower validity index indicates that the obtained clustering results are more compact and well-separated. Clearly, the ACP algorithm obtains proper partitions for minor prototypes according to the adaptive estimation of cluster centers. This is desirable, especially if the clustering algorithm is to be used to estimate membership distribution of multiple prototypes within a single class. In the next example, we illustrate this behavior in more detail.

<table>
<thead>
<tr>
<th></th>
<th>2-cluster example</th>
<th>3-cluster example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering algorithm</td>
<td>FCM</td>
<td>ACP</td>
</tr>
<tr>
<td>$XB$ index measure</td>
<td>0.2817</td>
<td>0.0139</td>
</tr>
<tr>
<td>$nXB$ index measure</td>
<td>0.4476</td>
<td>0.1459</td>
</tr>
</tbody>
</table>

**Example 2:**

In this example, we describe experiments in which the ACP algorithm was used to detect allographs for handwritten digit patterns, and patterns analysis was then considered. This example involves the proposed ACP algorithm for handwritten digit clustering by identifying multiple clusters from each class of patterns. Those clusters may be denoted as reference patterns. Reference patterns play an important role in achieving high performance in the character recognition system based on shape (or pattern) matching [Park, 1995; Jain, 1997].

For handwritten character patterns, many character classes consist of sub-classes due to different styles of writing. These sub-classes are also referred to as allographs, as described in
Section 1. Many, but not all, allographs represent the differences between cursive and printed styles, and it is hard to recognize these allographs if we ignore the variations of different styles [Chiang, 1998]. The data set used for the clustering experiment consists of un-normalized binary handwritten digits extracted from the BR digit set of the standard CEDAR CDROM-1 [Hull, 1994].

To construct the prototypes we used three classes of patterns, class “5”, class “7”, and class “8”, where 600 samples per class. In this experiment, we apply the ACP algorithms to three handwritten digit datasets to determine their allograph clusters, then comparing the results obtained from both ACP and FCM algorithms.

We used the transition feature [Gader, 1997] as the input vector for each sample during the ACP clustering. The transition feature extraction computes the location and number of transitions from white pixel to black pixel along horizontal and vertical lines. The transitions are 3 stroke-based sequences for each direction, resulting in a 60-dimensional feature vector. This feature set attempts to detect strokes, closed loops, shape tendency, etc.

**Handwritten digit “7”**

We take the handwritten digit “7” as our first example, samples of which are shown in Fig. 8.

![Fig 8 Samples of handwritten digit “7”](Image)

Humans can identify several cursive styles, as shown in Fig.1, and we will test whether the FCM or the ACP can find out these variations automatically and efficiently. We first group the data set of “7” using the FCM algorithm. The clustering results and partial exemplars for four clusters are shown in Fig.9. There are ten exemplars for each row in Fig. 9, and each row represents a cluster. The left 5 exemplars are the top 5 digits close to the cluster centers. They are
most representative patterns for corresponding cluster. The other five digits are selected from top 50 digits from each cluster, except for the exemplars with a star. These exemplars with a star are members that are not within the top 50 digits, and we will discuss later. Digits of \textit{Type I} spread in between clusters 1, 2, and 3. From the result shows in Fig. 9(a), we observe that the FCM algorithm is not a suitable way to detect the peculiar group, such as the \textit{Type I} set. The clusters derived from ACP are shown in Fig 9(b), with parameters $N = 3$ and $m = 1.4$. As the results show, the peculiar cursives belong to \textit{Type I} are grouped into the minor Cluster A, although the left 5 exemplars appear unlikely. Clusters B and C belong to \textit{Type II} group, and Cluster D is a typical \textit{Type III} group. \textit{Type I} patterns are separated successfully from digit “7” dataset by the ACP algorithm. The $XB$ and $nXB$ measures for the clustering results obtained from the FCM and the ACP are shown in Table 4, and it seen that these validity indices of the ACP are smaller than the indices of the FCM. We further set nine different numbers of clusters to examine clustering validation. The $XB$ measures for different numbers of clusters of the FCM algorithm are shown

![Fig 9 Clustering results of handwritten digit “7” by using (a) the FCM algorithm, and (b) the AFC algorithm.](image-url)
in Fig. 10(a), and the $XB$ measures for the ACP are shown in Fig. 10(b). Fig. 10(c) shows the $nXB$ measures for the clustering results from the two clustering algorithms. Clearly, the clustering results obtained from the ACP are obviously better than that from the FCM.

<table>
<thead>
<tr>
<th>Table 4 Validity measures for the clustering results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering algorithm</td>
<td>FCM</td>
</tr>
<tr>
<td>$XB$ index measure</td>
<td>0.8144</td>
</tr>
<tr>
<td>$nXB$ index measure</td>
<td>1.8528</td>
</tr>
</tbody>
</table>

It notes that we use different $m$ values for the two clustering algorithms, but we found that the clustering results provided by FCM were not as good as those provided by ACP. Without loss of any generality, we only demonstrated its clustering results with $m = 2$. Two exemplars located on the right in the first row of Fig. 9(b) belong to Cluster $\mathcal{A}$, but they are divided into two different clusters, Clusters 1 and 2, by the FCM in Fig 9(a). It can be seen that the cluster centers are quite different between FCM and ACP. Since it is unreasonable to assign these exemplars to different clusters, the clustering results of the ACP algorithm therefore perform better than the FCM.

**Handwritten digit “5”**

In the second experiment, we take the handwritten digit “5” as example, examples of which are shown in Fig. 11.
This digit class “5” has more strokes than other numerals, and its strokes include a right angle “┌” and an arc “└”. We observe that there are three kinds of different patterns from dataset. The exemplars shown on the left of Fig. 12, Type I, are similar to a printed style, neat and easily identified. This kind of style is a major class, which makes up almost half of the dataset. The next three images, located in the middle of Fig. 12, Type II, represent the cursive writing style of this digit. Approximately 40% of the patterns belong to this type. Some patterns, located on the right side of Fig 12 are special, denoted as Type III. This type is much different from printed style of numeral “5” and they can easily be confused with the letter “S”, as shown in Fig 13. This Type III is nearly 10% of the total, so it is a minority but important prototype.

Now we compare the cluster results obtained from both FCM and ACP. The six clusters obtained from FCM are shown in Fig 14(a). The first five clusters almost all belong to Type I and the last cluster belongs to Type II. Patterns of Type III are in between clusters 1 and 3. It can be seen that the FCM algorithm could not identify the peculiar allograph.
Here we utilize ACP for the same dataset. Parameters are $N_s = 4$ and $m = 1.5$. Six clusters derived from ACP are shown in Fig 14(b). The characteristics of clusters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ all belong to Type I group, and sixth cluster, $\mathcal{F}$, belongs to Type II. Cluster $\mathcal{E}$ clearly represents Type III group. The $XB$ measures for two clustering results are shown in Table 5. In this experiment, ACP successfully separates this peculiar allograph. Fig 15 shows $XB$ and $nXB$ measures for clustering results of the two cluster algorithms with different numbers of clusters, and we can obtain better clustering results from the ACP algorithm.

![Clusters ACP](image)

![Clusters FCM](image)

---

**Table 5  Validity measures for the clustering results**

<table>
<thead>
<tr>
<th>Clustering algorithm</th>
<th>Digit “5”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FCM</td>
</tr>
<tr>
<td>$XB$ index measure</td>
<td>4.4539</td>
</tr>
<tr>
<td>$nXB$ index measure</td>
<td>11.8538</td>
</tr>
</tbody>
</table>
Handwritten digit “8”

In the final experiment, we discuss allographs of the handwritten digit “8”, samples of which are shown in Fig. 16.

We can observe several major writing styles of the handwritten digit “8”. One characteristic is the slant of writing. Two different slants shown in Fig 17(a), one to the right, and the other to the left. Another style is postures of profile. Three different postures are shown in Fig 17(b). The left one is fatness, the middle one is thinness, and right one is an example with large loop on top.

In the following, we design a slant measure \( m_{rl} \) to calculate the degree of slant for each digit. In Fig 18, we assume there are two circumscribed lines at both sides of the digit, denoting the right and left hand circumscribed lines as \( SLP_r \) and \( SLP_l \), respectively. We select 5 related
elements, the 16th, the 19th, the 22nd, the 25th, and the 28th from 60-dimensions of the feature vector. Each element represents first locations from white pixel to black pixel along corresponding right-to-left transitions. We divide these elements into two groups, \{\text{Vec}_{16,5}, \text{Vec}_{19,4}, \text{Vec}_{22,3}\} and \{\text{Vec}_{22,3}, \text{Vec}_{25,2}, \text{Vec}_{28,1}\}, and select the maximum value from each group.

\[
(\text{Vec}_{\text{h}, \text{h}})_{\text{vec}, \text{i}} = \max(\text{Vec}, i) \quad \text{for } i = 16, 19, 22 \\
(\text{Vec}_{\text{h}, \text{d}})_{\text{vec}, \text{j}} = \max(\text{Vec}, j) \quad \text{for } j = 22, 25, 28
\] (25)

The slant measure \(m_{\text{m}}\) and its angle \(\theta_{\text{a}}\) can be calculated as

\[
m_{\text{m}} = \frac{(H_{\text{h}} - H_{\text{d}})}{V_{\text{h}} - V_{\text{d}}} \times \frac{\text{Height}}{\text{Width}}, \quad \text{and} \quad \theta_{\text{a}} = \tan^{-1}(m_{\text{m}})
\] (26)

Similarly, we compute \(SLP_L\) and \(m_{\text{m}}\) in the same manner.

Clustering results obtained from the ACP are shown in Fig. 19. We set number of clusters as 5 and 6 in Fig 19(a) and Fig 19(b), respectively. In Fig 19(a), most of the members in Cluster \(\mathcal{A}\) have positive slant measures \(m_{\text{m}}\), and are similar to the first exemplar in Fig 17(a). Members of Cluster \(\mathcal{B}\) are thin patterns. Cluster \(\mathcal{D}\) belongs to third exemplar shown in Fig 17(b). Clusters \(\mathcal{C}\) and \(\mathcal{E}\) are both fat patterns, although the degree of slant of Cluster \(\mathcal{E}\) is generally larger than that in Cluster \(\mathcal{C}\).

Then we increase number of clusters to 6, with the results as shown in Fig 19(b). It can be seen that the clustering results obtained show very small changes in clusters \(\mathcal{A}, \mathcal{B}, \mathcal{D},\) and \(\mathcal{E}\) but not
Cluster C in Fig. 19(a) is divided into Cluster \textbf{C-I} and Cluster \textbf{C-II} now. The differences in \textbf{C-I} and \textbf{C-II} are the degree of slant. The slant measures $m_a$, $m_b$, and $\theta_s$, $\theta_s$ for each cluster are listed in the Table 6. We evaluate the variations between clusters according to those measures. It can be seen that the Clusters \textbf{C-I} and \textbf{C-II} are significant different on $\theta_s$ value. Others are small change on the degrees of slants when the numbers of clusters increase. Furthermore, the $\theta_s - \theta_s$ values reach high values for Cluster \textbf{D} on both results, as shown in the last row of Table 6.

Fig 19 Clustering results of handwritten digit “8” for (a) $c = 5$, (b) $c = 6$. 

![Cluster A](image1)

![Cluster B](image2)

![Cluster C](image3)

![Cluster C-I](image4)

![Cluster D](image5)

![Cluster E](image6)

(a)

![Cluster A](image7)

![Cluster B](image8)

![Cluster C-I](image9)

![Cluster D](image10)

![Cluster E](image11)

(b)
Table 6 Averaged $m_c$ and $\theta$ values for each cluster

<table>
<thead>
<tr>
<th>Slant</th>
<th>$m_{\theta_l}$</th>
<th>$m_{\theta_R}$</th>
<th>$\theta_L$</th>
<th>$\theta_R$</th>
<th>$\theta_L - \theta_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ACP (c = 5)$</td>
<td>$A$</td>
<td>$B$</td>
<td>$C$</td>
<td>$D$</td>
<td>$E$</td>
</tr>
<tr>
<td>$C$</td>
<td>90.07</td>
<td>98.19</td>
<td>94.16</td>
<td>96.46</td>
<td>95.03</td>
</tr>
<tr>
<td>$D$</td>
<td>85.39</td>
<td>94.23</td>
<td>89.99</td>
<td>89.92</td>
<td>92.62</td>
</tr>
<tr>
<td>$E$</td>
<td>4.68</td>
<td>3.96</td>
<td>4.17</td>
<td>6.54</td>
<td>2.41</td>
</tr>
</tbody>
</table>
4. Conclusions

In this paper, we present an adaptive c-population approach to clustering. This algorithm has been analyzed for convergence, and behavior of the algorithm has been related to stochastic approximation. Furthermore, when the assumptions were satisfied, the algorithm converges to a stationary point. Several examples were given to verify the applicability of the analysis. The overall conclusion we reach from our experiments is that the ACP algorithm retains the adaptive characteristic of the fuzzy clustering approach, which is shown to be advantageous over the classical fuzzy c-means clustering. The work described in this paper also suggests several possible applications for further investigation, such as abnormal transactions detections for large amount of credit card applications in data mining and malignant illnesses symptoms detections in medical diagnosis.

5. Acknowledgment

This research work was supported in part by the National Science Council Research Grant NSC90-2213-E-006-089.
References


Haykin, S., Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999


Park, J.-S. and Lee, S.-W., “Adaptive Nonlinear Shape Matching for Unconstrained Handwritten
Character Recognition”, Pattern Recognition, 28(8), 1223-1235, 1995.


Rose, K., E. Gurewitz, and G. C. Foz, “Constrained Clustering as an Optimization Method,” *IEEE