*Ersch. in: International Journal of Intelligent Systems ; 19 (2004), 7. - S. 607-618*

# Fuzzy Information Granules in Time Series Data

Michael R. Berthold,[1,*] Marco Ortolani,[2,†] David Patterson,[3,‡]
Frank Höppner,[4,§] Ondine Callan,[5,¶] Heiko Hofer[1,‖]
[1]*University of Konstanz, 78457 Konstanz, Germany*
[2]*University of Palermo, Department of Electrical Engineering, Viale delle Scienze 90128 Palermo, Italy*
[3]*Tripos, Inc., 1699 S. Hanley Road, St. Louis, MO 64133*
[4]*University of Applied Sciences, Emden, Department of Electrical Engineering and Computer Science, Constantiaplatz 4, D-26723 Emden, Germany*
[5]*VistaGen Therapeutics, Inc., 1450 Rollins Road, Burlingame, CA 94010*

Often, it is desirable to represent a set of time series through typical shapes in order to detect common patterns. The algorithm presented here compares pieces of a different time series in order to find such similar shapes. The use of a fuzzy clustering technique based on fuzzy c-means allows us to detect shapes that belong to a certain group of typical shapes with a degree of membership. Modifications to the original algorithm also allow this matching to be invariant with respect to a scaling of the time series. The algorithm is demonstrated on a widely known set of data taken from the electrocardiogram (ECG) rhythm analysis experiments performed at the Massachusetts Institute of Technology (MIT) laboratories and on data from protein mass spectrography. 

## 1. INTRODUCTION

When processing large amounts of data, it is important to derive a compact representation that still retains all relevant information. For instance, a large part of the data might be grouped together into a few clusters, whereas the outliers should be isolated, thus making it easier to point them out for further analysis. Also, information gathered from experiments on real data often may be highly unreliable

*Author to whom all correspondence should be addressed: e-mail: Michael.Berthold@uni-Konstanz.de.
†e-mail: ortolani@pa.icar.cnr.it.
‡e-mail: pat@tripos.com.
§e-mail: frankhoeppner@ieee.org.
¶e-mail: ocallan@vistagen-inc.com.
‖e-mail: Heiko.Hofer@UniBw-Muenchen.de.

in quantitative terms and exhibit large amounts of noise. In the case of time series or other types of sequential data streams, it often is desirable to represent common groupings through typical shapes, representing a slice of the time window or a subsequence, respectively. In some cases, one may try to find relevant shapes such as peaks by using conventional peak-detection techniques. However, this may prove ineffective, because some of the peaks may overlap and form head-shoulder constellations. Such shapes generally are hard to identify as separate peaks and it might make more sense to look at them from a more global point of view, trying to recognize similarities instead of isolating peaks.

In this study, we present a method that finds areas in time series data that exhibit informative clusters of related shapes. The use of a fuzzy clustering technique based on fuzzy c-means allows us to assign overlapping degrees of membership and assign each pattern to prototypical shapes with a certain degree of membership. Because quantitative information is only marginally reliable in many of these data sets, the matching needs to be invariant under certain transformations of the time series, particularly scaling.

This article is organized as follows. Section 2 contains a short description of the fuzzy c-means clustering technique; in Section 3 we present our approach, introducing the use of a scale invariant objective function, and, after presenting some results in Section 4 and summarizing our conclusions in Section 5, we discuss some possible future developments in Section 6.

## 2. OBJECTIVE-FUNCTION–BASED FUZZY CLUSTERING

The general idea behind clustering is to partition a given data set into homogeneous subsets. One popular approach finds a partition of the original space and assigns each data element to one of the clusters by means of a similarity function, which often is based on the Euclidean distance as a metric. Each cluster then is represented by a prototype or cluster representative. The well-known fuzzy c-means algorithm[1] is an example of such a clustering algorithm, where, in addition, one allows each data element to belong to all clusters simultaneously but to different degrees. In formal terms, assuming we have a data set

$$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|X|}\} \subset \mathbf{R}^n, \quad n \in \mathbf{N}$$

the aim is to compute the prototypes $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_{|P|}\}$ as a result of the following optimization problem:

$$J_m(X; U, P) = \sum_{j=1}^{|X|} \sum_{i=1}^{|P|} u_{i,j}^m d_{i,j}^2 \tag{1}$$

using the constraints

$$\forall i \in \mathbf{N}_{\leq |P|} : \sum_{j=1}^{|X|} u_{i,j} > 0 \tag{2}$$

$$\forall j \in \mathbf{N}_{\leq |X|} : \sum_{i=1}^{|P|} u_{i,j} = 1 \tag{3}$$

i.e., we want to minimize the sum of weighted (squared) distances between data objects and cluster prototypes.

The membership degree of datum $\mathbf{x}_i$ to cluster $\mathbf{p}_i$ is denoted by $u_{i,j} \in [0, 1]$. The distance of datum $\mathbf{x}_j$ and cluster prototype $\mathbf{p}_i$ is denoted by $d_{i,j}$. The parameter $m > 1$ influences the "fuzziness" of the obtained partition.

With $m \to 1$ the partition tends to be crisp ($u_{i,j} \to \{0, 1\}$); with $m \to \infty$, totally fuzzy ($u_{i,j} \mapsto 1/|P|$), as will be evident considering the formula for updating $u_{i,j}$. Constraint in Equation 2 makes sure that none of the clusters is empty and thus we really have a partition into $|P|$ clusters. Constraint in Equation 3 assures that every datum has the same overall weight in the data set.

Fuzzy clustering under constraints in Equations 2 and 3 often is called "probabilistic clustering." Other fuzzy clustering techniques, using a relaxed constraint in Equation 3, are noise clustering[2] and possibilistic clustering.[3] The latter approaches are especially useful when dealing with very noisy data.

The most popular fuzzy clustering algorithm is the fuzzy c-means algorithm. It uses the Euclidean distance between data vector $\mathbf{x_j}$ and prototype $\mathbf{p_i}$ as distance measures. This model searches for spherical clusters of approximately the same size.

Most of the objective-function–based fuzzy clustering algorithms minimize Equation 1 by alternatingly optimizing the membership degrees and cluster shapes. From the membership model (e.g., "probabilistic") and the cluster shape model (e.g., "point-like") one can develop necessary conditions for a local minimizer of $J$ from $\partial J/\partial U = 0$ and $\partial J/\partial P = 0$. Of course, for each model, we obtain different update equations. Ideally, we have in both cases closed-form update equations, which makes the algorithms much faster and more robust when compared with variants that use additional numerical techniques like the Newton-Raphson method. In the case of the fuzzy c-means algorithm, we obtain for the probabilistic membership model the update equation

$$u_{i,j} = \frac{1}{\sum_{k=1}^{|P|} \left( \frac{d_{i,j}^2}{d_{k,j}^2} \right)^{1/(|P|-1)}} \tag{4}$$

and for the point-like shape model the update equation

$$\mathbf{p_i} = \frac{\sum_{j=1}^{|X|} u_{i,j}^m \mathbf{x_j}}{\sum_{j=1}^{|X|} u_{i,j}^m} \tag{5}$$

Besides the point-like clusters, hyperellipsoidal shapes,[4] linear shapes,[1] and many others are known in the literature. We refer to Ref. 5 for a thorough overview.

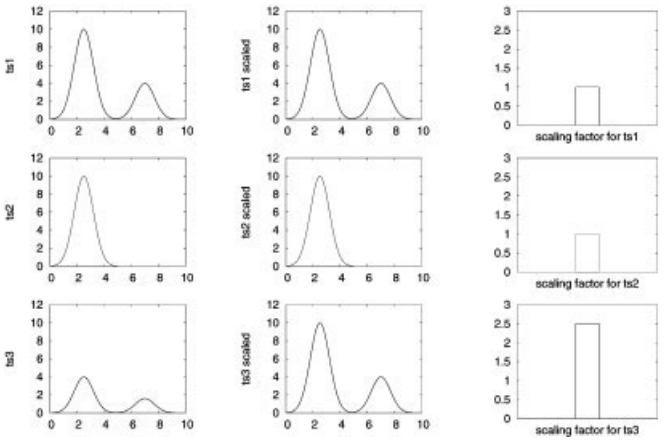


**Figure 1.** Three time series showing the advantage of a scale invariant approach. On the left, the three unscaled samples are displayed; in the middle, the same samples after scaling are shown; and on the right the scaling factors used are shown.

## 3. SCALE INVARIANT APPROACH

For our purposes, every data object represents (part of) a time series and the aim is to cluster them according to their similarity.

Figure 1 shows an example in which a scale invariant approach may be effective. Computing similarity using the usual Euclidean distance as a metric series *ts1* and *ts2* are closer to each other than to *ts3*. However, if one is interested in similar shapes, it would be preferable to have *ts1* and *ts3* in the same cluster. After scaling, it is obvious that series *ts1* and *ts3,* indeed, can be traced back to a common prototype, whereas *ts2* always will show up as "different."

Given a time series $(t_i)_{i \in \mathbf{N}}$ we define the associated data object $\mathbf{x}$ to consist of $n$ consecutive observations: $\mathbf{x}_j = (t_0, t_1, t_2, \ldots, t_{n-1})$. Analogously, every cluster is represented by a prototype, which is an $n$-dimensional vector that can be interpreted as (part of) a time series.

In addition, we are interested in a partition that takes into account that we are uncertain about the scale of each time series. Hence, we introduce variable scaling parameters and, unlike the original algorithm, measure the Euclidean distance of the scaled data object to the prototypes. Obviously, for different prototypes, different scaling factors minimize the Euclidean distance; therefore, we use $s_{i,j}$ to denote the scaling factor for data object $\mathbf{x}_j$ to match prototype $\mathbf{p}_i$.

This approach is more flexible than the standard fuzzy c-means because the requirement for each object to match exactly a prototype is softened. Moreover, it takes into account the whole (scaled) object and its overall shape rather than, e.g., trying to identify the present of peaks, which often is impractical depending on the nature of the data set.

Finally, it also is more effective than simply having a fixed scaling factor common for all the objects, as would be the case, for instance, when normalizing all the time series a priori and applying the standard fuzzy c-means algorithm.

choose termination threshold $\varepsilon$
choose fuzzifier $m$ (popular choices $1.5 \leq m \leq 3$)
initialize prototypes $p_i$
**repeat**
// update scaling factors :
$\forall i, j : s_{i,j} := \frac{\mathbf{x_j}^{\top} \mathbf{p_i}}{\|\mathbf{x_j}\|^2}$
// update memberships :
$\forall i, j : u_{i,j} := \left( \sum_{k=1}^{|P|} \left( \frac{\|s_{i,j}\mathbf{x_j} - \mathbf{p_i}\|^2}{\|s_{k,j}\mathbf{x_j} - \mathbf{p_k}\|^2} \right)^{\frac{1}{m-1}} \right)^{-1}$
// update prototypes :
$\forall i : \mathbf{p_i} := \sum_{j=1}^{n} u_{i,j}^m s_{i,j} \mathbf{x_j}$
// normalize prototypes :
$\forall i : \mathbf{p_i} := \frac{\mathbf{p_i}}{\|\mathbf{p_i}\|}$
**until** change in prototypes $< \varepsilon$

**Figure 2.** Scale invariant clustering algorithm.

This leads to a modified objective function:

$$J_m(X;\, U,\, P) = \sum_{j=1}^{|X|} \sum_{i=1}^{|P|} u_{i,j}^m \|s_{i,j}\mathbf{x}_j - \mathbf{p}_i\|^2 \tag{6}$$

now, we need to place an additional constraint on Equation 6 in order to avoid the trivial solution of $\{\mathbf{p_i} \equiv \mathbf{0},\, s_{i,j} = 0\}$. Every prototype $\mathbf{p_i}$ might be scaled by an arbitrary factor without changing anything in the value of the objective function if we consider the same factor for the scaling factors $s_{i,j}$. Therefore, we choose a fixed scale for the prototypes, requiring

$$\forall i : \|\mathbf{p}_i\| = 1 \tag{7}$$

Skipping the derivation of the necessary conditions for the parameter updates, an alternating optimization clustering algorithm minimizing Equation 6 under the constraint in Equation 7 is given in Figure 2.

Note that it is not necessary to store the scale and membership matrix completely if the prototypes $\mathbf{p}_i$ are updated incrementally.

## 4. EXPERIMENTAL RESULTS

The algorithm was tested using two different data sets; the former contained electrocardiogram (ECG) signals, and for the latter, the signal consisted of protein mass spectrograms. In the first case, the signals were not particularly noisy and required little preprocessing, whereas in the second case, many interesting shapes would fall unnoticed to a conventional peak detection algorithm. However, in both cases, similar conclusions may be drawn from the experiments as outlined in Section 4.3.

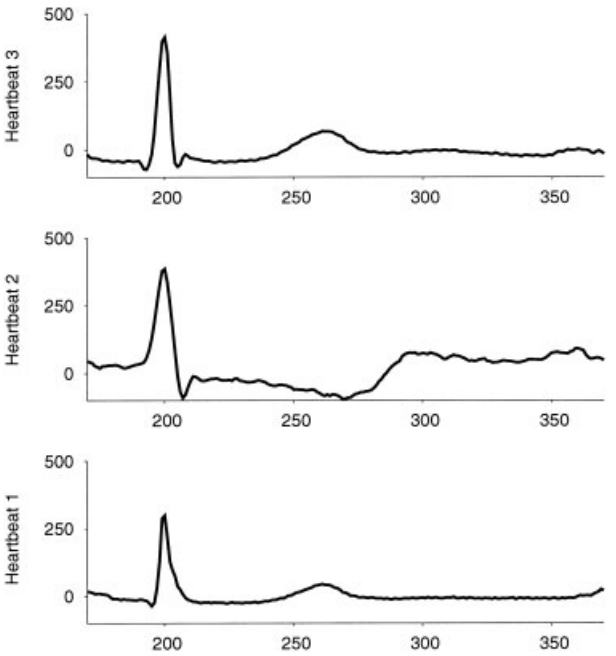


**Figure 3.** Heartbeats from three different series. Data is shifted to have zero mean but not scaled. Heartbeats 1 and 3, although with different scales, show roughly the same shape, whereas Heartbeat 2 is different. Note, however, that the (unscaled) Euclidean distance between Heartbeats 1 and 2 is smaller than the distance between Heartbeats 1 and 3.

### 4.1. Heartbeat Data

For our preliminary experiments, we used a data set consisting of ECG signals extracted from the Massachusetts Institute of Technology–Beth Israel Hospital (MIT-BIH) Arrhythmia Database,[6] which is a set of recordings analyzed and labeled by human experts. Both the signals and the cardiologists' annotations are freely available.

From a medical point of view, our data represent a range of common clinical phenomena such as normal beats (the majority), paced beats, premature ventricular contractions, right bundle branch block beat, and atrial premature beat. Only the ones that result in peculiar shapes lend themselves to be recognized by our algorithm (e.g., paced beats usually differ from normal beats only in magnitude, thus being intrinsically indistinguishable for us).

Figure 3 shows a typical scenario. The time series representing three recorded heartbeats were shifted to have zero mean but not normalized. Note how the two normal heartbeats (Numbers 1 and 3) have similar shapes but different scales. Heartbeat 2 clearly is different but still has a smaller Euclidean distance to Heartbeat 1 than Heartbeat 3. A purely distance-based clustering algorithm would incorrectly assign Heartbeats 1 and 2 to one cluster and Heartbeat 3 to a second cluster.

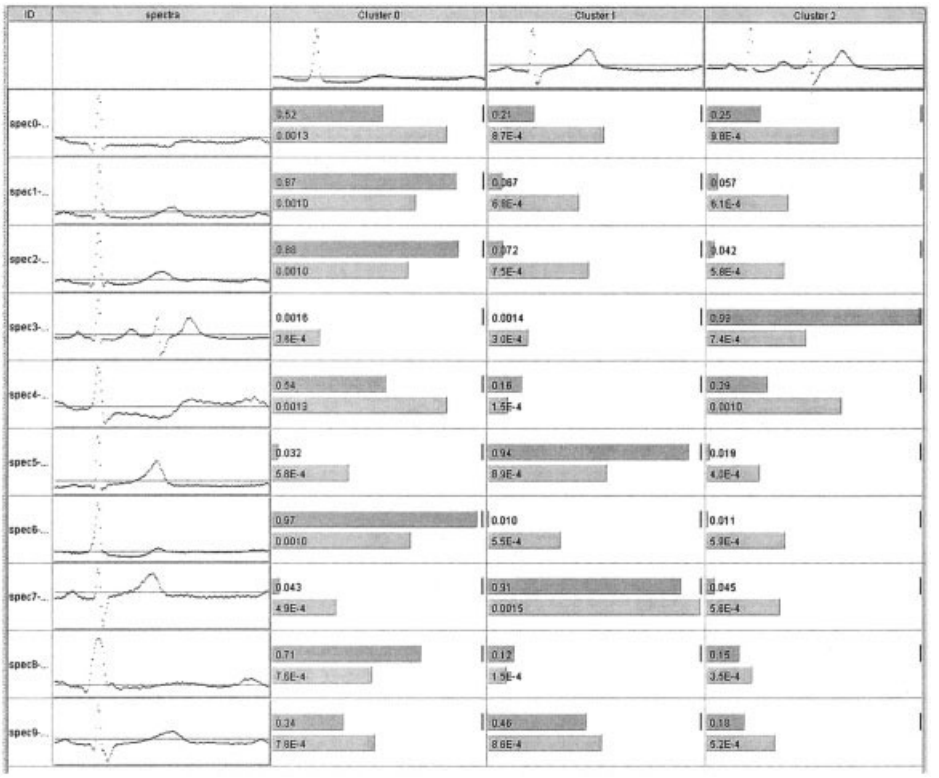


**Figure 4.** Ten time series are grouped using three clusters. The upper bar in each cell of the table shows the degree of membership of each time series to the relative cluster and the lower one is the resulting scaling factor. Note that the time series shown already are scaled accordingly.

Previous work to characterize different types of heartbeats mostly has focused on a set of parameters that were extracted automatically, such as width and height of certain, characteristic features. Using these parameters is not straightforward and methods to determine the influence of these measures have been proposed as well.[7]

However, most experts do not rely on such summaries, because they intuitively tend to analyze the overall shape of a time series to determine its category. On the other hand, our approach allows us to find clusters of similar shapes as well, which mimics the human expert more closely than going through an intermediate process of translating the time series into a set of parameters.

Even though the data is not particularly noisy, some preliminary cleaning was conducted, namely, the subtraction of the mean and an alignment of the signals in order to have all of the time series represent exactly one heartbeat.

Several experiments have been performed to test how the algorithm works in a real case. Figure 4 is an example of a typical output, where one of the time series presents an anomalous behavior (*spec3,* which shows a premature ventricular contraction). In the example shown, 10 samples are grouped using 3 clusters; the first row shows the 3-cluster representatives, and the time series (already scaled)

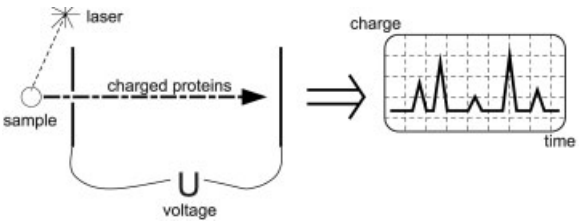


**Figure 5.** The principle behind protein mass spectrography is based on accelerated charged proteins in vacuum. Based on voltage and distance, the observed charge over time-of-flight plot can be used to identify concentrations of proteins at a certain mass.

are shown in the first column. The upper bar in each cell of the table shows the degree of membership of each time series to the relative cluster and the lower one is the resulting scaling factor. The relatively small values for the scaling factor stem from the fact that we require the prototypes to be normalized to one and that the spectra's norm is much larger than one.

The algorithm produces three rather well-separated groups; one of them contains *spec3* alone, and the remaining samples are clustered into two groups, with the more populated one containing the time series with a more common behavior.

### 4.2. Protein Mass Spectra

The results on heartbeat data were confirmed by further experiments conducted on protein mass spectrograms.

In Figure 5, the basic operation of protein mass spectrography is sketched. Charged proteins are accelerated in a vacuum and the charge over time-of-flight plot can be used to draw conclusions about the concentrations of proteins of specific mass. In reality, however, the resulting information is highly unreliable in quantitative terms and also exhibits large amounts of noise. Figure 6 shows an example of mass-over-charge diagrams derived from a real protein mass spectrography instrument. Note how although both plots were derived from the same sample, the quantitative information, i.e., the peak heights, vary. In addition, a heavy baseline offset and a substantial amount of noise is visible. The enlarged sections show areas where it is hard to identify all peaks using conventional peak-detection techniques, because some of them overlap and form head-shoulder constellations. Such shapes generally are hard to identify as separate peaks.

A typical approach to find features in those kinds of signals requires detecting individual peaks and somehow assigning quantitative information to each peak. This requires some sort of normalization and a reliable peak-detection algorithm, as opposed to the more intuitive approach of the algorithm presented here.

In this case, it is not desirable to present the entire time series to the algorithm as one spectra. Therefore, we chose a sliding window approach, where we used overlapping slices in time as input to subsequent runs of the algorithm as shown in Figure 6.

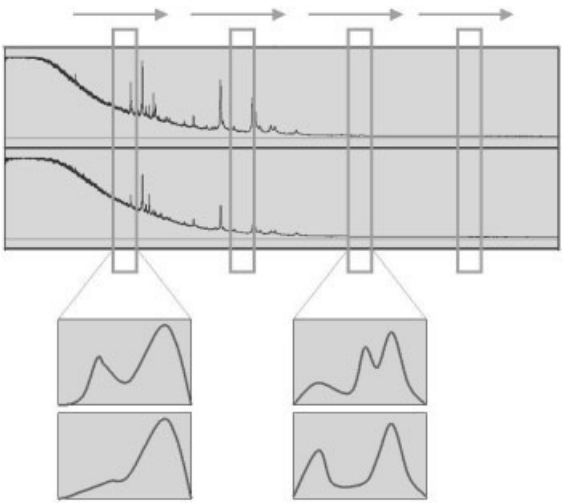

**Figure 6.** Two examples of real mass-over-charge plots from a protein mass spectrography instrument. Repeated runs of the algorithm on different parts of the time series help to point out characteristic shapes.

Assuming the time series are aligned, each run of the algorithm selects different parts of them and produces the respective cluster representatives.

Not all of the cluster representatives will be necessarily significant, but they are good candidates for a cluster validity assessment algorithm that can select the most promising ones.

The final result is a set of prototypes that represent characteristic shapes occurring in (parts of) the time series.

Figure 7 shows two examples of running the presented algorithm on a set of 192 mass spectrograms (the precise nature of the underlying sample is not of prime interest for this example). Two screen shots are shown, which display a series of mass spectrograms on the left, together with a label indicating the categories rep$x$/39$y$-rep$x$/tc$y$. The number $x$ following "rep" indicates an individual experiment using eight different samples (39, 40, 41, 42, 46, 47, rc, and tc) and $y$ = 'a' − 'h' denotes duplicate experiments using the same sample.

In this case, the cells display only a bar showing the degree of membership of each sample to each of the clusters, which are displayed on the top row together with the samples from which they were derived. It is interesting to see how the method finds clusters that group samples of classes 39–42 and 46–tc together on the left side. A clustering in a different region, shown on the right, nicely separates the sixth repetition from the remaining five (rep6 versus rep1–5), an indication that the sixth experiment ran into problems.

### 4.3. Discussion

Besides the discussed results, some observations arise from both sets of experiments. First, it is important to note that because the number of clusters is

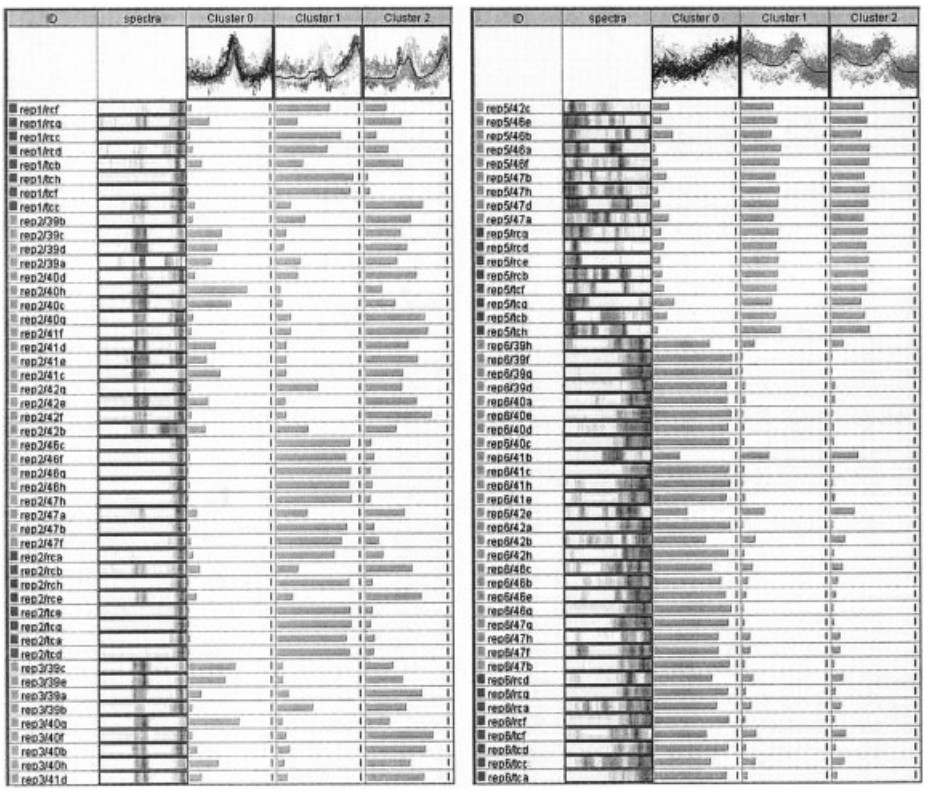


**Figure 7.** Two examples of clusters for a certain range of time-of-flight values.

chosen a priori, the analysis of a range where none of the samples showed any particular discriminative shape was bound to produce more clusters than necessary. Nevertheless, when a certain phenomenon (i.e., an area with a peculiar shape) was present, the algorithm usually was able to detect it as an outlier, assigning it to a cluster of its own.

When the number of clusters is chosen too large, a high fuzziness index results in the memberships being almost equally spread, which is not particularly meaningful. On the other hand, with fewer clusters, the fuzziness, together with the scaling factor, produces a better clustering. The usual situation is that some clusters are reserved for the outliers, if present, with the rest of the samples showing very low memberships on those clusters; at the same time, they will group together in the remaining clusters according to the respective similarities (but the difference in the memberships is not so evident).

We also compared the algorithm with a standard fuzzy c-means (i.e., without scaling factor). As expected, because the similarity measure is basically the same, the latter is bound to come up with worse results; with the same number of clusters and fuzziness, the results tend to be "sharper," because even small differences in the time series that appear similar but at different scales are enhanced. Because the

number of prototypes is not determined by the algorithm, it will try to assign each spectrum to one of the clusters, even if this may result in "bad" values for the memberships, i.e., memberships equally spread along the possible prototypes. The introduction of a validity assessment function would provide a quantitative measure of the goodness of the scaling invariant algorithm with respect to the original one.

## 5. CONCLUSIONS

The test on real data sets has shown that our algorithm is capable of generating meaningful clusters taking into account shape similarities, and it succeeded in separating common shapes from unusual ones. The procedure is similar to that of a human expert, which naturally rejects differences in scale but focuses on particular shapes. As expected, carefully choosing the fuzziness degree as well as the number of clusters is important and including the scaling factor into the objective function to be minimized has proved to be successful.

The fact that outliers usually are isolated can certainly be useful in some application and help to further refine the analysis. Even though these preliminary experiments were encouraging and basically confirmed theoretical results, they also gave us some hints on how to further improve the algorithm as outlined in the next section.

## 6. FUTURE WORK

It is clear that having a fixed number of clusters is not the best solution. This constraint is caused by the class of algorithms that the fuzzy c-means belong to. We hope that we can overcome this limitation at least partially. Using cluster validity assessment techniques[8,9] is a first step in this direction. In addition, leaving the scaling factors completely unconstrained usually is not desirable as well. In some instances, noise was artificially blown out of proportion to match a certain prototype in cases where this was clearly nonsensical. Defining valid ranges for the scaling factors would have helped to avoid these effects.

## References

1. Bezdek JC. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press; 1981.
2. Davé RN. Characterization and detection of noise in clustering. Pattern Recognit Lett 1991;12:657–664.
3. Krishnapuram R, Keller J. A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1993;1:98–110.
4. Gustafson DE, Kessel WC. Fuzzy clustering with a fuzzy covariance matrix. In: Proc of the IEEE Conf on Decision and Control. Fort Lauderdale, FL 1979. pp 761–766.
5. Höppner F, Klawonn F, Kruse R, Runkler T. Fuzzy cluster analysis. Chichester, UK: John Wiley & Sons; 1999.
6. The MIT-BIH Arrhythmia Database. http://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm.

7. Silipo R, Berthold MR. Input features impact on fuzzy decision processes. IEEE Trans Syst Man Cybern B 2000;30(6):821–834.
8. Pal NR, Bezdek JC. On cluster validity for the fuzzy c-means model. IEEE Trans Fuzzy Syst 1995;3(3):370–379.
9. Halkidi M, Batistakis Y, Vazirgiannis M. On clustering validation techniques. Intell Info Syst J. 2001;17(2):107–145.