

Fuzzy Information Granules in Time Series Data

Marco Ortolani^{1,3}, Heiko Hofer¹, David Patterson¹, Frank Höppner², and Michael R. Berthold¹

¹Tripos Inc., Data Analysis Research Lab
601 Gateway Blvd., Suite 720, South San Francisco, CA 94080, USA

²University of Applied Sciences, Emden
Department of Electrical Engineering and Computer Science
Constantiaplatz 4, D-26723 Emden, Germany

³University of Palermo
Department of Electrical Engineering
Viale delle Scienze 90128 Palermo, Italy

¹{ortolani,hhofer,pat,berthold}@tripos.com

²frank.hoepner@ieec.org

Abstract - It is often desirable to summarize a set of time series through typical shapes in order to analyze them. The algorithm presented here compares pieces of different time series in order to find similar shapes. The use of a fuzzy clustering technique based on fuzzy c-means allows us to consider such subsets as belonging to typical shapes with a degree of membership. Additionally, this matching is invariant with respect to a scaling of the time series. The algorithm is demonstrated on a widely known set of data taken from the ECG rhythm analysis experiments performed at MIT laboratories.

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 few clusters whereas the outliers should be isolated, thus making it easier to point them out for further analysis. In this paper we show a modification to fuzzy c-means to find scale-invariant clusters of similar shapes in time series.

The paper 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 independent 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 dataset into homogeneous subsets. One popular approach

consists in finding a partition of the original space and assigning each data element to one of the clusters by means of a similarity function, which is often based on the Euclidean distance as a metric. Each cluster is then represented by a prototype, or cluster representative. The well-known fuzzy c-means algorithm [1] is an example for 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 $X = \{x_1, \dots, x_{|X|}\} \subset \mathbb{R}^n$, $n \in \mathbb{N}$, the aim is to compute the prototypes $P = \{p_1, \dots, p_{|P|}\}$ as a result of the following optimization problem:

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

using the constraints

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

$$\forall j \in \mathbb{N}_{\leq |X|} : \sum_{i=1}^{|P|} u_{i,j} = 1 \quad (3)$$

that is, we want to minimize the sum of weighted (squared) distances between data objects and cluster prototypes. The membership degree of datum x_j to cluster p_i is denoted by $u_{i,j} \in [0, 1]$. The distance of datum x_j and cluster prototype p_i is denoted by $d_{i,j}$. The parameter $m > 1$ influences the “fuzziness” of the obtained partition.

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

Fuzzy clustering under constraints (2) and (3) is often called “probabilistic clustering”. Other fuzzy clustering techniques, using a relaxed constraint (3), are noise clustering [2] and possibilistic clustering [6]. 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 x_j and prototype p_i as distance measure. This model searches for spherical clusters of approximately the same size.

Most of the objective function based fuzzy clustering algorithms minimize (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 $\frac{\partial J}{\partial U} = 0$ and $\frac{\partial J}{\partial P} = 0$. 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 case of the fuzzy c-means algorithm, we obtain for the probabilistic membership model the update equation

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

and for the point-like shape model the update equation

$$p_i = \frac{\sum_{j=1}^{|X|} u_{i,j}^m x_j}{\sum_{j=1}^{|X|} u_{i,j}^m} \quad (5)$$

Besides the point-like clusters, hyper-ellipsoidal shapes [3], linear shapes [1] and many others are known in the literature. We refer to [5] for a thorough overview.

3 SCALE INDEPENDENT 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 a typical scenario with three recorded heartbeats. The time series were shifted to have zero mean but not normalized. Note how the two normal heartbeats (number 1 and 3) have similar shape but different scale. Heartbeat 2 is clearly different, but still has a smaller Euclidean distance to heartbeat 1 than heartbeat 2. A purely distance based clustering algorithm would incorrectly assign heartbeats 1 and 2 to one cluster and heartbeat 3 to a second cluster.

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

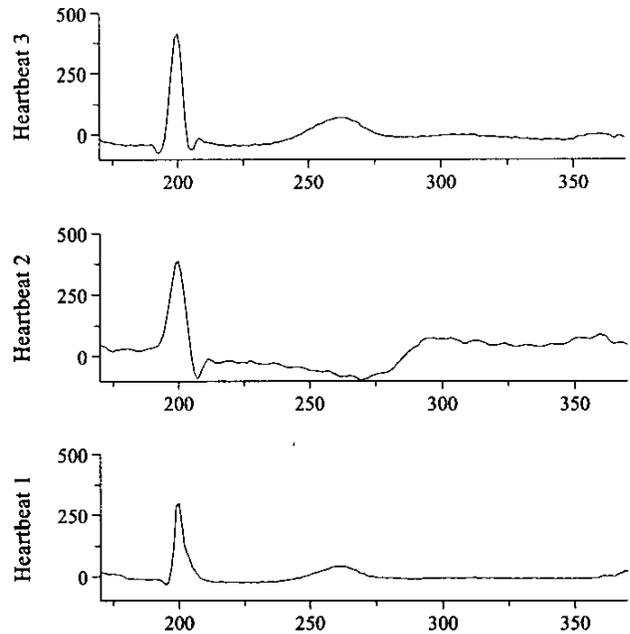


Figure 1: Heartbeats from 3 different series. Data is shifted to have zero mean, but not scaled. Heartbeat 1 and 3, although with different scales, show roughly the same shape, whereas heartbeat 2 is different. Note, however, that the (unscaled) Euclidean distance between heartbeat 1 and 2 is smaller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

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 measure the Euclidean distance of the scaled data object to the prototypes rather than the distance between the unscaled objects; this gives the algorithm more flexibility as opposed to having a fixed scaling factor (as would be the case, for example, when normalizing all the time-series a-priori and applying the standard fuzzy c-means algorithm). Obviously, for different prototypes different scaling factors minimize the Euclidean distance, we therefore use $s_{i,j}$ to denote the scaling factor for data object x_j to match prototype p_i . This leads to a modified objective function:

$$J_m(X; U, P) = \sum_{j=1}^{|X|} \sum_{i=1}^{|P|} u_{i,j}^m \|s_{i,j} x_j - p_i\|^2 \quad (6)$$

To avoid the trivial solution of $\{p_i \equiv 0, s_{i,j} = 0\}$, we have to place a constraint on (6). Every prototype 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

$$\forall i : \|p_i\| = 1 \quad (7)$$

Skipping the derivation of the necessary conditions for the parameter updates, an alternating optimization clus-

```

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

```

Figure 2: Scale-Invariant Clustering Algorithm.

tering algorithm minimizing (6) under the constraint (7) is given in Figure 2.

Note that it is not necessary to store the scale and membership matrix completely if the prototypes p_i are updated incrementally.

4 EXPERIMENTAL RESULTS

Our dataset consists of ECG signals extracted from the MIT-BIH Arrhythmia Database [7], 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 is representative of 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 (for instance, paced beats usually differ from normal beats only in magnitude, thus being intrinsically undistinguishable for us).

Previous work to characterize different types of heart beats has mostly 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 [9]. Most medical experts do not rely on such summaries, however, since they analyze the overall shape of a heartbeat to determine its category. Our approach on the other hand allows us to find clusters of similar shape 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 subtrac-

tion of the mean and an alignment of the signals in order to have all the time series represent exactly one heart beat.

Several experiments have been carried out to test how the algorithm works in a real case. Figure 3 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 case shown, 10 samples are grouped using 3 clusters; the first row shows the 3 cluster representatives, while the time series (already scaled) 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 algorithm produces three rather well separated groups; one of them contains *spec3* alone, while the remaining samples are clustered into two groups, with the more populated one containing the time series with a more common behavior.

One of the major drawbacks of this kind of algorithms is that it requires a fixed number of clusters. The typical choice for this was 3 or 4, which correspond to the most evident shapes (which also correspond to particular physical conditions of the patients). It is important to note that, since the number of clusters is chosen a priori, the analysis of a range where none of the samples showed any particular clinical condition was bound to produce more clusters than necessary. Nevertheless, when a clinical phenomenon (that is a time series with a peculiar shape) was present, the algorithm was usually 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, since 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 time series that appear similar but at different scales are enhanced. Since 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, that is 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

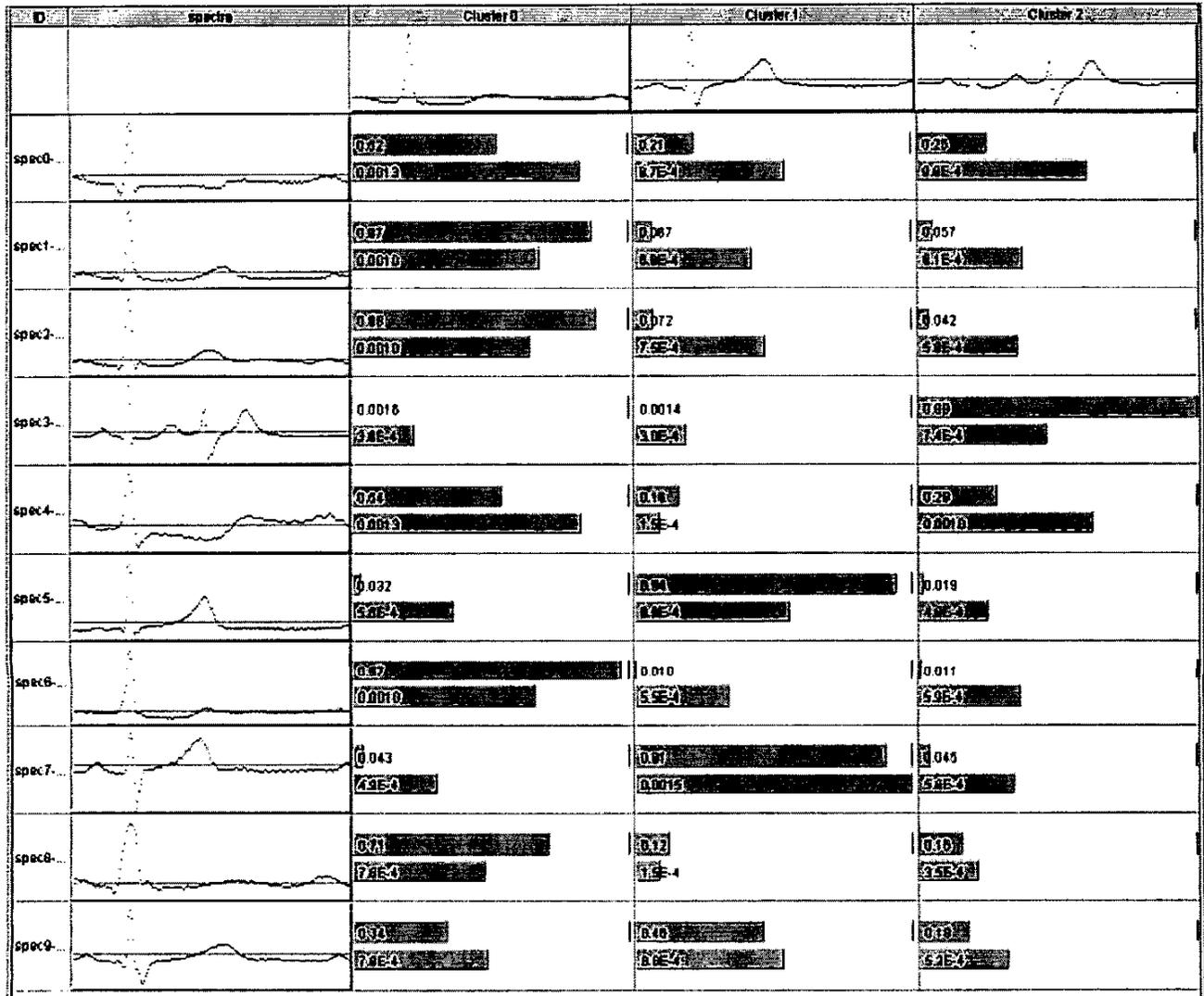


Figure 3: 10 time series are grouped using 3 clusters. The upper bar in each cell of the table shows the degree of membership of each time series to the relative cluster and the lower one is the resulting scaling factor. Note that the time series shown are already scaled accordingly.

respect to the original one.

5 CONCLUSIONS

The test on a real dataset 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 rather 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 proven to be successful.

The fact that outliers are usually isolated can certainly be useful in some application and 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 due to the class of algorithms which the fuzzy c-means belongs to; nevertheless it can be partially overcome and the use of a cluster validity assessment technique ([8], [4]) could be a first step in this direction, in that it would suggest us a better choice for the number of clusters required to characterize our set of spectra.

Another feature that is highly evident in our chosen dataset is the periodicity of the data. Even though this cannot be assumed to be the rule, it suggests that an analysis in the frequency domain could be useful, for example by taking into account the Fourier transform of each sample. The idea is to apply the same algorithm on both domains and collect the results; roughly speaking, since a scaling in the frequency domain corresponds to a complex shift in time, we can argue that it is possible to combine both effects by running the same algorithm in parallel on both domains.

References

- [1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [2] R. N. Davé, Characterization and detection of noise in clustering, *Pattern Recognition Letters*, 12:657–664, Nov. 1991.
- [3] D. E. Gustafson and W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, *Proc. of the IEEE Conference on Decision and Control*, pages 761–766, Jan. 1979.
- [4] M. Halkidi, Y. Batistakis and M. Vazirgiannis, On Clustering Validation Techniques, invited paper to appear in *Intelligent Information Systems Journal*, Kluwer Publishers.
- [5] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*, John Wiley & Sons, Chichester, England, 1999.
- [6] R. Krishnapuram and J. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Systems* 1, pages 98–110, 1993.
- [7] The MIT-BIH Arrhythmia Database available at <http://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>
- [8] N. R. Pal and J. C. Bezdek, On cluster validity for the fuzzy c-means model, *IEEE Trans. on Fuzzy Systems*, 3(3), pages 370–379, Aug. 1995.
- [9] R. Silipo and M. R. Berthold, Discriminative Power of Input Features in a Fuzzy Model, *Advances in Intelligent Data Analysis, Lecture Notes in Computer Science (LNCS 1642)*, pages 87–98, D. Hand, J. Kok, M. Berthold (eds), Springer Verlag, 1999.