Shape-Invariant Fuzzy Clustering of Proteomics Data

Michael R. Berthold, David E. Patterson, Marco Ortolani
Tripos Inc., Data Analysis Research Lab
601 Gateway Blvd., Suite 720
South San Francisco, CA 94080, USA
{berthold,pat,ortolani}@tripos.com

Heiko Hofer
Institute of Automation
Chemnitz University of Technology
Germany
heiko.hofer@e-technik.tu-chemnitz.de

Frank Höppner
University of Applied Sciences
Department of Computer Science
Salzdahlumer Str. 46/48, D-38302 Wolfenbüttel, Germany
frank.hoeppner@ieee.org

Ondine Callan
VistaGen Therapeutics, Inc.
Burlingame, CA, USA
ocallan@vistagen-inc.com

Abstract

In this paper we present a variant of fuzzy c-means that allows to find similar shapes in time series data in a scale-invariant fashion. We use data from protein mass spectrography to show how this approach finds areas of interest without a need for ad-hoc normalizations.

1. Motivation

When analyzing time series data, especially from biological domains such as proteomics mass spectrography, it is crucial to extract relevant pieces of information in order to quickly gain some insights into the vast amounts of data. In mass spectrography the recorded data often exhibits vast variances in quantitative information, which up to now required cumbersome heuristics for normalization and base-line subtraction. In Figure 1 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 2 shows an example of mass-over-charge diagrams derived from a real protein mass-spec 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 section shows an area where it is hard to identify all peaks using conventional peak-detection techniques, since some of them overlap and form head-shoulder constellations. Such shapes generally are hard to identify as separate peaks.

In this paper we present a method that finds areas in such spectra 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. Since quantitative information is only marginally reliable in many of these data sets, the matching needs to be invariant under certain transformations of the spectra, particularly scaling. The proposed method considers different sub-samples, obtained by sliding a temporal window over the set of time series and scores the resulting clustering of each passage in order to identify well separated clusters or clusters that offer good class

0-7803-7461-4/02/$17.00 © 2002 IEEE

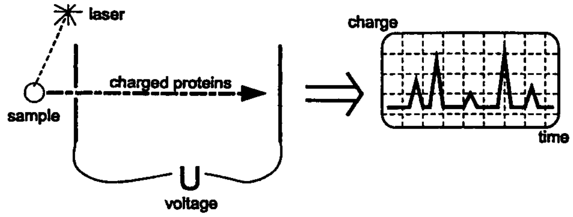

Figure 1. The principle behind Protein Mass Spectrography is based on accelerated charged proteins in vacuum. Based on voltage and distance, the observed charge over time-of-flight plot can be used to identify concentrations of proteins at a certain mass.

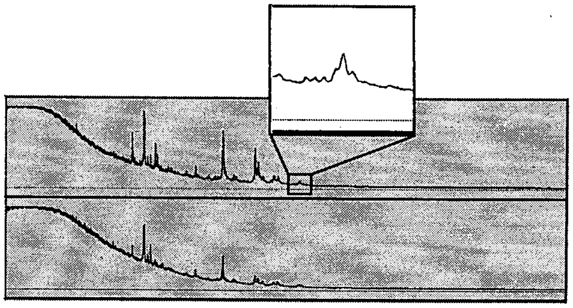

Figure 2. Two examples of real mass-over-charge plots from a protein mass-spectrography instrument.

discriminability; the final outcome are a small collection of relevant granules representing shape fragments of interest, which characterize the original set of mass specs.

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

$$X = \{x_1, ..., x_{|X|}\} \subset \mathbf{R}^n \ , n \in \mathbf{N}$$

the aim is to compute the prototypes $P = \{p_1, ..., p_{|P|}\}$ as a result of the following optimization problem:

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

using the constraints

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

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

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 $m \to 1$ the partition tends to be crisp ($u_{i,j} \to \{0, 1\}$); with $m \to \infty$, totally fuzzy ($u_{i,j} \to \frac{1}{|P|}$). 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

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

and for the point-like shape model the update equation

$$p_i = \frac{\sum_{j=1}^{|X|} u^m_{i,j} x_j}{\sum_{j=1}^{|X|} u^m_{i,j}} \tag{5}$$

Besides 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 Invariant Clustering

For our purposes, every data object represents (part of) a time series and the aim is to cluster them according to their similarity. Given a time series $(t_i)_{i \in \mathbb{N}}$ we define the associated data object x to consist of n consecutive observations: $x_j = (t_0, t_1, t_2, ..., t_{n-1})$. 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 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:

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

```
choose termination threshold $\varepsilon$
choose fuzzifier $m$ (popular choices $1.5 \le m \le 3$)
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 :
    $\forall i : p_i := \sum_{j=1}^{n} u^m_{i,j} s_{i,j} x_j$
    // normalize prototypes :
    $\forall i : p_i := \frac{p_i}{\|p_i\|}$
until change in prototypes $< \varepsilon$
```

Figure 3. The Scale-Invariant Clustering Algorithm.

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

$$\forall i : \|p_i\| = 1 \tag{7}$$

Skipping the derivation of the necessary conditions for the parameter updates, an alternating optimization clustering algorithm minimizing (6) under the constraint (7) is given in Figure 3.

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

Previous work to find features in protein mass spectrograms has mostly focused on detecting individual peaks and somehow assigning quantitative information to each peak. This requires some sort of normalization and a reliable peak detection algorithm.

However, biologists often do not want to rely on such summaries, since they want to investigate the overall shape of a region of a spectra to determine its category instead. The approach presented here allows the user to find clusters of similar shape as well, which mimics the human expert more closely than going through an intermediate process of translating the spectra into a set of peaks with associated heights.

Figure 4 shows two examples of running the presented algorihtm 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 specs on the left, together with a label indicating the categories rep$x$/39y-rep$x$/tc$y$. The number x following 'rep' indicates an individual experiment using 8 different samples (39, 40, 41, 42, 46, 47, rc, tc) and y='a'-'h' denotes duplicate experiments using the same sample.

The top row shows the cluster representatives, in this case for three clusters. The bars in each cell represent the degree of membership of each pattern to a specific cluster. It is interesting to see how the method finds clusters that group samples of class 39-42 and 46-tc together on the left side. A clustering in a different region, shown on the right, nicely separates the 6th repetition from the remaining five (rep6 vs. rep1-5), an indication that the 6th experiment ran into problems.

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 discriminative shape was bound to produce more clusters than necessary. Nevertheless, when a certain phenomenon (that is an area 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. The screen shot on the right of Figure 4 is a nice example of this effect.

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 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 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. We hope that we can overcome this limitation at least partially, using cluster validity assessment techniques ([7], [4]) could be a first step in this direction. In addition leaving the scaling factors completely unconstrained is usually 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] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

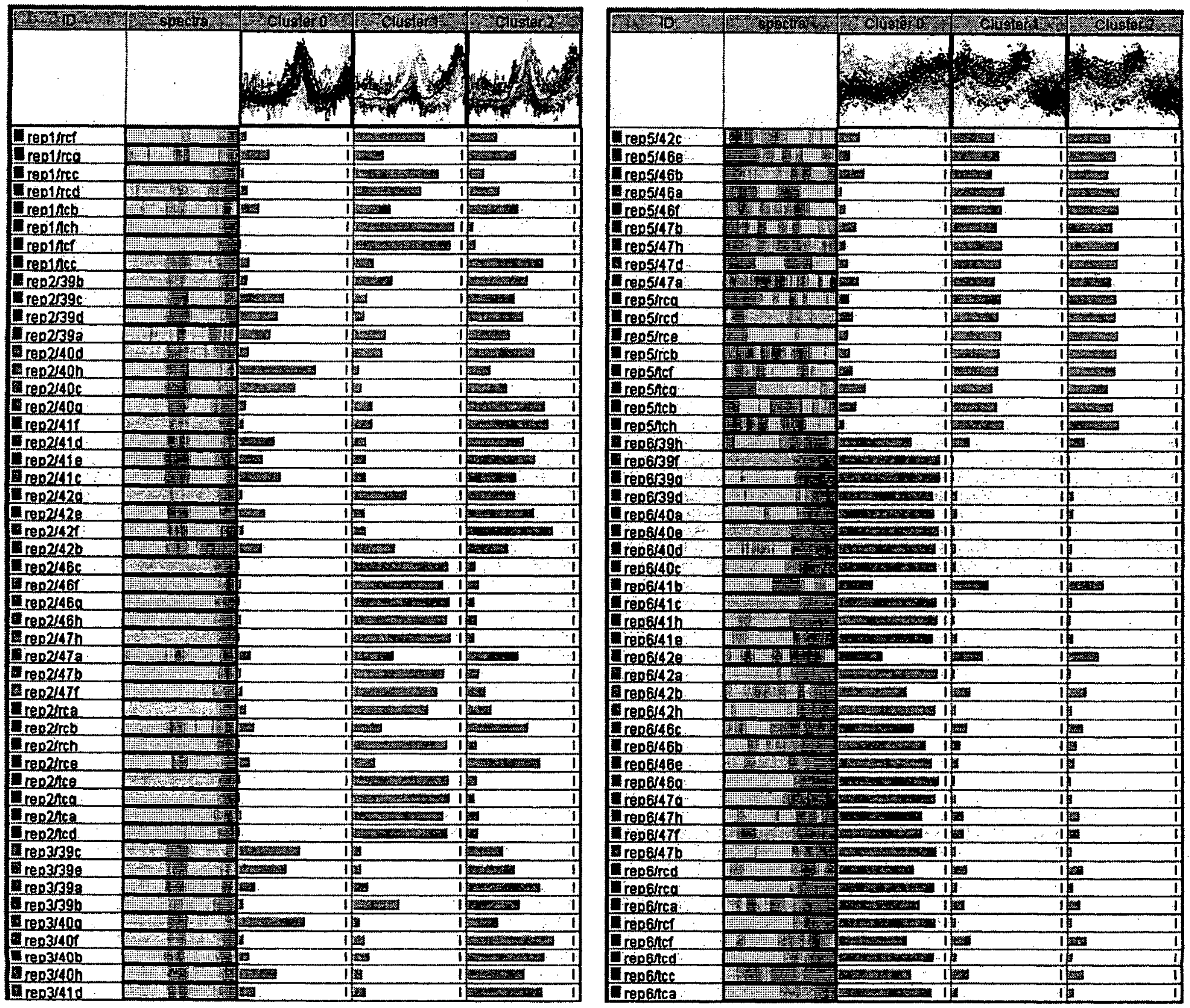

Figure 4. Two examples of clusters for a certain range of time-of-flight values.

[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, to appear in *Intelligent Information Systems Journal*, Kluwer Pulishers.

[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] 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.